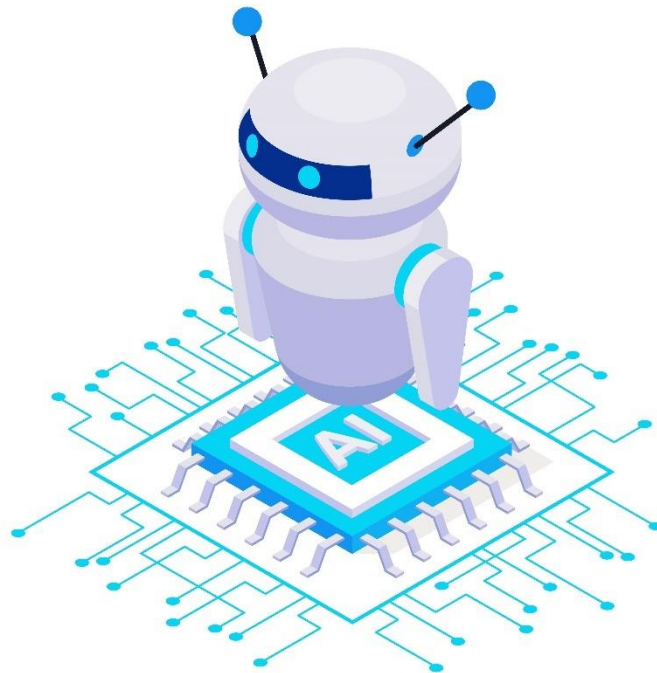


М. В. Васильківський, Г. Г. Бортник, В. М. Кичак

**Програмні технології в інфокомунікаційних
системах. Навчальний посібник для
студентів спеціальності 172
«Телекомунікації та радіотехніка»**



М. В. Васильківський, Г. Г. Бортник, В. М. Кичак

**Програмні технології в інфокомунікаційних
системах. Навчальний посібник для
студентів спеціальності 172
«Телекомунікації та радіотехніка»**

**Електронний навчальний посібник
комбінованого (локального та мережного) використання**

Вінниця
ВНТУ
2023

УДК 621.0
B90

Рекомендовано до видання Вченою радою Вінницького національного технічного університету Міністерства освіти і науки України (Протокол № 5 від 29.12.2022 р.)

Рецензенти:

В. Г. Крижановський, доктор технічних наук, професор
С. М. Захарченко, кандидат технічних наук, професор
О. В. Осадчук, доктор технічних наук, професор

Васильківський, М. В.

B90 Програмні технології в інфокомунікаційних системах. Навчальний посібник для студентів спеціальності 172 «Телекомунікації та радіотехніка» : електронний навчальний посібник комбінованого (локального та мережного) використання [Електронний ресурс] / Васильківський М. В., Бортник Г. Г., Кичак В. М. – Вінниця : ВНТУ, 2023. – 141 с.

Навчальний посібник є частиною навчально-методичного комплексу та призначений для підготовки та проведення лекційних занять з дисципліни «Теоретичні основи технології програмування телекомунікаційних систем та мереж». У навчальному посібнику міститься необхідний теоретичний матеріал, алгоритми та методика оптимізації інфокомунікаційних систем та мереж.

УДК 621.0

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	4
ВСТУП.....	6
1 ТЕОРЕТИЧНІ ОСНОВИ ПРОГРАМНО-КЕРОВАНИХ ІНФОКОМУНІКАЦІЙНИХ МЕРЕЖ.....	8
1.1 Огляд керування мережею та послугами.....	8
1.2 Огляд штучного інтелекту та машинного навчання.....	20
2 ПРОГРАМНІ ТЕХНОЛОГІЇ В РАДІОМЕРЕЖАХ.....	28
2.1 Керування віртуалізованими мережами та службами за допомогою машинного навчання.....	28
2.2 Самокеровані мережі 5G	56
3 МОДЕЛІ ТА АЛГОРИТМИ ПРОГРАМНОГО КЕРУВАННЯ.....	80
3.1 Штучний інтелект в мережах 5G.....	80
3.2 Машинне навчання для розподілу ресурсів у мобільних широкосмугових мережах	97
3.3 Навчання з підкріпленням для розподілу функціональних сервісів в туманних обчисленнях.....	113
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	137

ПЕРЕЛІК СКОРОЧЕНЬ

5G – Стандарт п'ятого покоління для широкосмугових стільникових мереж
6G – Стандарт шостого покоління для широкосмугових стільникових мереж
AD – Адміністративний домен
AE – Автоматичний кодувальник
ШІ – штучний інтелект
ШНМ – Штучні нейронні мережі
API – Інтерфейс прикладного програмування
AP – Точка доступу
ARQ – Автоматичний запит на повтор
AC – Автономна система
ASIC – Спеціальна інтегральна схема
BNG – Шлюз широкосмугової мережі
C/S – клієнт-сервер
CNN – Згорткові нейронні мережі
CQI – Індикатор якості каналу
DC – Центр обробки даних
DNN – Глибинна нейронна мережа
GUI – Графічний інтерфейс користувача
ILP – Цілочислове лінійне програмування
IoT – Інтернет речей
KPI – Ключовий показник ефективності
LoRaWAN – Мережа великого радіусу дії
LP-WAN – Глобальна мережа малого споживання
MEC – Граничні обчислення з кількома доступами
MIB – Інформаційна база керування
MILP – Змішано-цілочислове лінійне програмування
MINLP – Змішані задачі цілочисельного нелінійного програмування
ML – Машинне навчання
MNO – Оператор мобільної мережі
MPLS – Багатопротокольна комутація міток
MSE – середня квадратична помилка
NFV – Віртуалізація мережних функцій
NFVI – Інфраструктура віртуалізації мережних функцій
NFVO – Оркестратор віртуалізації мережних функцій
NIC – Контролер мережного інтерфейсу
NN – Нейронна мережа
NOC – Центр керування мережею
OTN – Оптична транспортна мережа
QoS – Якість обслуговування
RAN – Мережа радіодоступу
RAP – радіоточка доступу

RDD – Стійкий розподілений набір даних
RIP – Протокол інформації про маршрутизацію
RL – Навчання з підкріпленням
RNN – Рекурентна нейронна мережа
RRM - Керування радіоресурсами
SDN – програмно визначена мережа
SNMP – Простий протокол керування мережею
SNR – Відношення сигнал/шум
vBS – Віртуальна базова станція
VM – Віртуальна машина
VMO – Віртуальний мобільний оператор
VNF – Функція віртуальної мережі
WAN – Глобальна мережа
WLAN – Бездротова локальна мережа
WN – Бездротові вузли

ВСТУП

Досягнення штучного інтелекту та алгоритмів машинного навчання надають безмежні можливості в багатьох різних наукових та інженерних дисциплінах, зокрема мережі комп'ютерного зв'язку. Тому необхідні дослідження, щоб зрозуміти й покращити потенціал і придатність штучного інтелекту та машинного навчання загалом для комунікаційних і мережних технологій і досліджень, а також зокрема для операцій і керування системами й мережами. Такі підходи та методи, як штучний інтелект, інтелектуальний аналіз даних, статистичний аналіз і машинне навчання, є багатообіцяючими механізмами використання величезного потоку оперативних даних для покращення керування та безпеки ІТ-систем і мереж. Це не тільки забезпечить глибше розуміння та прийняття кращих рішень на основі значної кількості зібраних і доступних операційних даних, але також створить можливості для вдосконалення алгоритмів і методів аналізу даних щодо таких аспектів, як точність, масштабованість та узагальнення.

Навчальний посібник буде зосереджений на нещодавніх нових підходах і технічних рішеннях, які можуть використовувати штучний інтелект, машинне навчання та аналітику великих даних для комунікаційних мереж і рішень для керування послугами. У цьому контексті навчальний посібник призначений для викладачів інформаційних і комунікаційних технологій, інженерів і професіоналів з точки зору уявлення картини поточного розвитку та обговорення можливостей і викликів цієї галузі на майбутнє. Навчальний посібник можна використовувати для бакалаврату першого рівня.

Посібник складається з трьох частин, які містять поглиблений огляд поточного розвитку, можливостей, викликів і вдосконалень, створених штучним інтелектом і технікою машинного навчання для керування мережами та послугами.

Перший розділ, вступ, дають загальний огляд досліджень керування мережами та послугами, а також методів штучного інтелекту та машинного навчання.

Підрозділ 1.1 «Огляд керування мережею та послугами» окреслює сферу керування мережею та послугами, яка передбачає встановлення, конфігурацію, адміністрування та керування мережами та пов'язаними службами, щоб гарантувати, що мережні ресурси ефективно надаються клієнтам і споживаються так само ефективно.

Підрозділ 1.2 «Огляд штучного інтелекту та машинного навчання» містить огляд алгоритмів AI/ML, які найчастіше використовуються в сфері керування мережами та послугами, а також обговорює стратегічні сфери в керуванні мережами та послугами, які свідчать про зростаючий інтерес спільноти до розробки розподілу інформаційних ресурсів на основі AI/ML.

Другий розділ посібника «Програмні технології в радіомережах» присвячений таким важливим моделям і інфраструктурам керування, як віртуалізовані мережі, мережі 5G і хмарні обчислення.

Підрозділ 2.1 «Керування віртуалізованими мережами та послугами за допомогою машинного навчання» розкриває найсучасніші дослідження, які використовують штучний інтелект і машинне навчання для вирішення складних проблем у розгортанні та керуванні віртуалізованими мережами та службами. Він також окреслює відкриті, видатні дослідницькі завдання та можливості реалізації автоматизованого керування віртуалізованими мережами та послугами.

Підрозділ 2.2 «Самокеровані мережі 5G» обговорює основні виклики, що виникають на шляху до успішної розробки систем 5G, зосереджуючись, зокрема, на мережах радіодоступу, оптичних мережах, керуванні даних, нарізці мережі та оркестровці послуг, а також висвітлюється автономне керування даними, керування мережею та об'єднання адміністративних доменів, які є критично важливими для розвитку систем 5G.

Третій розділ посібника, «Функції керування та додатки», зосереджений на життєво важливих функціях керування та додатках, зокрема керування продуктивністю, безпекою та додатки Blockchain.

1 ТЕОРЕТИЧНІ ОСНОВИ ПРОГРАМНО-КЕРОВАНИХ ІНФОКОМУНІКАЦІЙНИХ МЕРЕЖ

1.1 Огляд керування мережею та послугами

Сьогодні мережа, тобто Інтернет, стала основним інструментом для ефективної підтримки високоцінних рішень, які стосуються нашого повсякденного життя. Створений переважно для передачі даних, сьогодні ми використовуємо Інтернет для перегляду відео високої чіткості, проведення відеоконференцій, отримання інформації, участі в соціальних мережах, іграх, купівлі товарів і веденні бізнесу. Усі ці послуги з доданою вартістю потребують підтримки найвищого рівня обслуговування мережі, де збої в обслуговуванні неприпустимі, а якість послуг має бути гарантована. В результаті численних впливів цифровізації Інтернет стає дедалі складнішим і важчим в керуванні, з мережами мобільного широкосмугового доступу, здатними з'єднувати мільярди користувачів зі швидкістю сотні мегабітів на секунду, магістральними мережами, що простягаються на тисячі кілометрів з кількома терабітами на секунду, виділеними каналами та величезними центрами обробки даних, що містять сотні тисяч серверів, віртуальних машин і програм.

Потреба в керуванні мережами та послугами виникла разом із першими мережними концепціями з основами, які були визначені в еталонній моделі взаємозв'язку відкритих систем (ISO/OSI) Міжнародної організації зі стандартизації [1, 2]. Телефонні мережі почали переходити на цифрові послуги в 1970-х роках, що створило потребу в автоматичному керуванні цими послугами [3].

Технологія комп'ютерного передавання даних радикально змінила мережну парадигму з використанням протоколу керування передачею/протоколом Інтернету (TCP/IP), що привело до створення Інтернету, яким ми його знаємо сьогодні. Спочатку керування комп'ютерною мережею було здебільшого ручною діяльністю, під час якої мережний адміністратор знав конфігурацію кожного пристрою та міг швидко втрутитися у разі виникнення проблем. У наш час, коли мережі містять мільярди пристроїв, мільйони вузлів, тисячі додатків, керування мережею та послугами стало настільки автоматизованим, наскільки це можливо. Удосконалення централізованих і розподілених підходів дозволили Центру керування мережею (NOC) візуалізувати та контролювати мережу максимально автоматизованим способом.

Сьогодні, маючи можливість збирати та обробляти великі обсяги даних, керування мережею та послугами стикається з новим стимулом до повної автоматизації, завдяки підходам машинного навчання та штучного інтелекту, які починають розгортатися в операційній діяльності проектування та експлуатації телекомунікаційних систем та мереж.

Керування мережею та послугами реалізує, в основному, цикл керування, у якому дані про стан мережі збираються для подальшої централізованої або розподіленої обробки для виявлення змін з метою визначення дій, які потрібно впроваджувати, реагувати та контролювати зміни. На рисунку 1.1 показано загальний огляд всього процесу. Ліворуч збираються дані про стан мережі для постійного моніторингу її стану. Технології великих даних у поєднанні з машинним навчанням і рішеннями штучного інтелекту дозволяють збирати, аналізувати та отримувати плани вирішення проблем, які потім поширюються на мережні пристрої для впровадження бажаних змін. Нижче відображено огляд технологій для моніторингу та виконання кроків. При цьому, чітко зосереджуємося на протоколах для зборання та моніторингу стану мережі й розповсюдження управлінських рішень. Натомість залишаємо для окремих розділів опис алгоритмів і підходів, які, за визначенням, дуже залежать від варіанта використання та конкретних технологій. Метою цього розділу є дати короткий огляд останніх тенденцій у технологіях керування мережами та послугами, а також навести огляд рішень високого рівня в домінуючих сценаріях для відображення ширшої картини проблеми. При цьому залишаємо конкретні рішення для окремих розділів разом із прикладами та більш глибокими обговореннями. Також зосереджуємося, в основному, на Інтернеті, оскільки він сьогодні є домінуючою мережею.

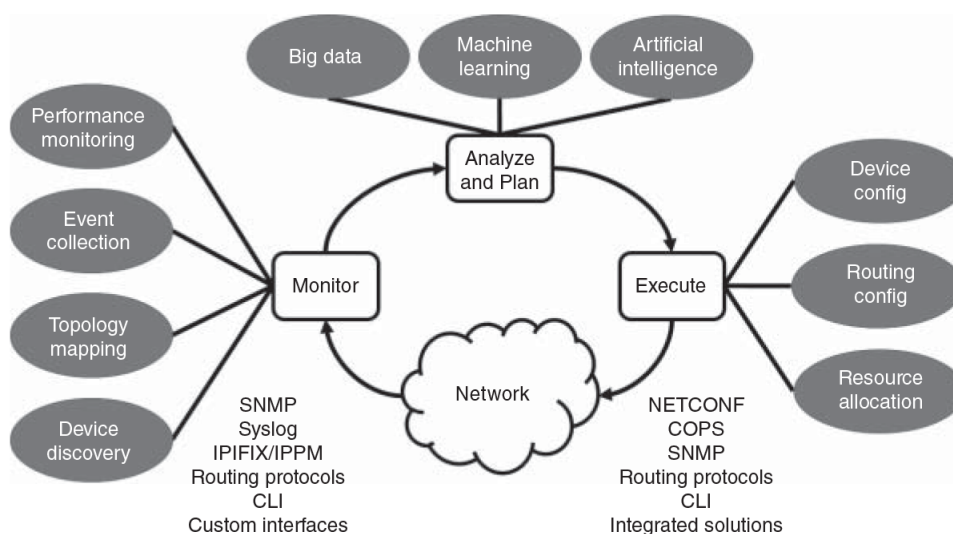


Рисунок 1.1 – Керування інфокомунікаційною мережею та інформаційними послугами

Будь-який процес прийняття рішення має керуватися можливістю отримати дані про стан системи. У типовій мережі пристрої від різних постачальників, з різними функціями, різними можливостями, різними адміністративними доменами створюють неоднорідні сценарії, де збирання даних потребує стандартизованих інструментів. Часто ця неоднорідність

породжує індивідуальні рішення, які надає кожен постачальник, пропонуючи розширені та власні рішення для взаємодії з різними спеціальними пристроями. Тому наводимо огляд основних стандартних протоколів, які дозволяють збирати дані з мережних пристроїв, залишаючи спеціальні рішення поза цим описом.

Оригінальне керування мережею TCP/IP базується на сімействі простого протоколу керування мережею (SNMP). SNMP стандартизує збирання та організацію інформації про пристрої в IP-мережі. Він заснований на моделі менеджера/агента з простим форматом запиту/відповіді. В результаті менеджер мережі надсилає запит, а керовані агенти надсилають відповіді у відповідь. SNMP надає дані керування у формі змінних, організованих у інформаційній базі керування (MIB), яка описує стан і конфігурацію системи. Потім ці змінні можна дистанційно запитувати та маніпулювати ними, дозволяючи як збирати інформацію, так і змінювати конфігурацію, за умови, що менеджер має повноваження контролю щодо таких змінних. SNMPv1 є оригінальною версією протоколу [4]. Останні версії, SNMPv2c і SNMPv3, мають покращення продуктивності, гнучкості та, особливо, безпеки [5, 6].

За допомогою цього простого підходу авторизований агент може дистанційно перевіряти та змінювати конфігурацію пристроїв у своєму адміністративному домені, поширюючи зміни, одночасно отримуючи оновлену картину стану мережі. Таким чином, SNMP пропонує засоби як для збирання інформації з мережних пристроїв, так і для керування ними, але не надає засобів для визначення найкращої конфігурації для впровадження.

Подібно SNMP, сімейство протоколів Syslog [7] пропонує механізми для збирання інформації про журнал подій. Спочатку використовувався в системах Unix і розроблявся з 1980 року, протокол є багаторівневою архітектурою, що дозволяє використовувати будь-які транспортні протоколи. Протокол Syslog дозволяє машині надсилати повідомлення системного журналу через мережі до збирачів повідомлень про події. Він реалізує підхід push, коли пристрої надсилають інформацію збирачам. Протокол просто призначений для транспортування та розповсюдження цих повідомлень про події, уможливорюючи централізоване збирання журналів із серверів, маршрутизаторів і пристроїв загалом.

На відміну від SNMP, Syslog не дозволяє поширювати будь-яку конфігурацію, яка має бути досягнута за допомогою інших каналів зв'язку.

Повідомлення містять код пристрою та рівень серйозності. Першим визначається тип програми, яка реєструє повідомлення (наприклад, ядро, користувач, пошта, сервер тощо).

Останнім визначається терміновість повідомлення (наприклад, екстрене, попередження, критичне, помилка, налагодження тощо). Це дозволяє легко фільтрувати та легко читати повідомлення. Під час роботи в мережі syslog використовує парадигму клієнт-сервер, де сервер збирача

читає повідомлення від клієнтів. Останні версії, створені для використання протоколу дейтаграм користувача (UDP), підтримують TCP і протокол TLS (Transmission Level Security) для надійного та безпечного зв'язку.

Syslog характеризується відсутністю стандартного формату повідомлень, тому кожна програма підтримує власний набір повідомлень. Зазвичай навіть різні версії програмного забезпечення однієї програми використовують різні формати, що ускладнює оброблення повідомлень автоматичними рішеннями.

Як syslog, так і SNMP дозволяють збирати інформацію про стан пристроїв.

Протокол експорту інформації про потік Інтернет-протоколу (IPFIX) натомість визначає засоби збирання стандартним способом інформації про трафік, що протікає в мережі. Деталізація, на якій він працює, – це потік, тобто група пакетів, що мають однакове джерело та призначення [8]. Він визначає компоненти, які беруть участь у вимірюванні та звітності інформації про IP-потіки. Процес вимірювання генерує записи потоку; процес експорту передає інформацію за допомогою протоколу IPFIX; і процес збирання отримує їх як записи даних IPFIX.

Протокол IPFIX є лише механізмом push, і IPFIX не може поширювати конфігурації експортерам. Аналогічно Syslog, він пропонує засоби для збирання інформації про трафік, що протікає в мережі, але не надає засобів для його обробки. Бувши заснованим на лічильниках трафіку, він відкриває можливість реалізації рішень для профілювання трафіку, проектування трафіку, моніторингу QoS і виявлення вторгнень, які аналізують вимірювання трафіку на основі потоку та створюють цінний зворотний зв'язок для менеджерів мережі. IPFIX є еволюцією NetFlow, спеціального попередника, представленого Cisco у 1996 році для збирання та моніторингу інформації про потік IP-мережі.

IPFIX не тільки підтримує протокол передачі керування потоком (SCTP) на транспортному рівні, але також дозволяє використовувати TCP або UDP для розвантаження програми вимірювання.

Протоколи NetFlow і IPFIX є прикладами методів «на основі метаданих», які можуть надати цінну операційну інформацію щодо продуктивності мережі, безпеки та інших програм. Наприклад, у IP-мережах записи метаданих документують потоки. У кожному записі потоку «ідентифікація» – це IP-адреси та номери портів, а «тривалість» – це кількість байтів і пакетів. Пряме збирання даних і аналіз базових пакетів даних також можна використовувати для усунення несправностей у роботі мережі та безпеки, наприклад, експорт необроблених пакетів. Зазвичай це передбачає певний рівень технічної складності та витрат, який у більшості ситуацій не забезпечує більш ефективного розуміння порівняно з ефективною системою для збирання та аналізу метаданих, що містять записи мережного потоку.

Основним критичним моментом IPFIX є його недостатня масштабованість для збирання даних та надмірне навантаження на мережу. Це змушує часто активувати параметри вибірки пакетів, що обмежує видимість.

Показники продуктивності протоколу Інтернету (IPPM) є прикладом успішної стандартизації [9]. Він визначає показники для точного вимірювання та звітності про якість, продуктивність і надійність мережі. До них належать з'єднання, затримка та втрати в одному напрямку, затримка та втрати в обидві сторони, варіації затримок, шаблони втрат, переупорядкування пакетів, потужність масового транспортування та вимірювання пропускнуої спроможності каналу. Він пропонує стандартну та спільну основу для визначення та вимірювання продуктивності, щоб навіть вимірювання, які виконуються різними постачальниками та реалізаціями, стосувалися однієї контрольованої метрики. В результаті це відкриває можливість загального моніторингу продуктивності.

Серед стандартних протоколів специфікація метрик протоколу одностороннього активного вимірювання та протоколу двостороннього активного вимірювання (OWAMP [10] і TWAMP [11], відповідно) дозволяє вимірювання затримки, втрати та зміни порядку. OWAMP можна використовувати в обох напрямках для вимірювання односторонніх показників в обох напрямках між двома елементами мережі. Однак він не підтримує вимірювання в обидві сторони. TWAMP розширює можливості OWAMP, додаючи двостороннє вимірювання. У вимірюванні беруть участь два хости. У випадку OWAMP відправник і одержувач активно співпрацюють, щоб виміряти бажаний індекс продуктивності. Наприклад, щоб обчислити затримку в один бік, беруть правильну позначку часу пакета вимірювань у час надсилання та отримання відповідно. У TWAMP приймач може діяти як простий відбивач, який просто надсилає назад (або третій стороні) тестовий пакет, надісланий відправником, без додаткових обчислювальних дій.

Відкритий вихідний код і пропрієтарні реалізації легко доступні для стеків протоколів IPv4 і IPv6. Вони також зазвичай інтегровані в платформи моніторингу [12], а саме: Perfsonar [13] або RIPE Atlas [14].

Протоколи маршрутизації є одними з найуспішніших розгорнутих рішень для керування мережею. Протокол маршрутизації визначає, як маршрутизатори спілкуються один з одним для обміну інформацією, яка дозволяє їм отримати поточну топологію мережі та обчислити шляхи для досягнення можливих місць призначення. Протоколи маршрутизації дають Інтернету можливість динамічно пристосовуватися до таких мінливих умов, як зміни топології, збої каналів і вузлів, а також ситуації перевантаження. Існує два основних класи протоколів маршрутизації, які використовуються в IP-мережах. Протоколи внутрішнього шлюзу, засновані на протоколах маршрутизації на основі вектора відстані, наприклад, на протоколі інформації про маршрутизацію (RIP) [15], вдосконаленому протоколі

маршрутизації внутрішнього шлюзу (EIGRP) [16], або на основі таких протоколів маршрутизації за станом зв'язку, як Open Shortest Path First (OSPF) [17], Intermediate System to Intermediate System IS-IS [18]; використовуються в мережах, які належать одному домену адміністратора, тобто в межах однієї автономної системи (AS). Протоколи внутрішніх шлюзів базуються на мінімізації вартості шляху, визначеної як сума вартості зв'язку. Таким чином, вони спрямовані на мінімізацію витрат на маршрутизацію трафіку, тобто максимізацію продуктивності. Натомість протоколи зовнішніх шлюзів спрямовані на обмін інформацією про маршрути між автономними системами та пошук найбільш зручного шляху (з точки зору автономних систем) для досягнення пункту призначення. Тут фактично єдиним вибором є Border Gateway Protocol (BGP) [19]. Це протокол векторної маршрутизації, і він приймає рішення щодо маршрутизації на основі мережних політик і правил, а не на основі функцій вартості. BGP дозволяє операторам мережі визначати політику маршрутизації, яка відображає адміністративні витрати та політичні рішення в рамках угод між автономними системами.

Враховуючи важливість оптимізації зовнішньої політики маршрутизації та часткове уявлення, яке кожен оператор мережі може отримати щодо топології рівня глобальної автономної системи (AS), існує кілька механізмів для отримання видимості поточної маршрутизації Інтернету.

Усі згадані протоколи маршрутизації реалізують механізми замкнутого циклу – від моніторингу до дій. Інша категорія протоколів маршрутизації надає можливості щодо проектування трафіку та керування мережею. Серед них багатопрокольна комутація міток (MPLS) [27], яка є технікою маршрутизації, заснованою на принципі заміни міток. Кожен вузол уздовж шляху зчитує мітку вхідних пакетів і використовує її для швидкої маршрутизації пакетів до наступного переходу. Перед операцією пересилання мітка пакета замінюється новою міткою, яка вказує на наступну операцію пересилання, що має бути виконана на наступному вузлі. Через конкатенацію міток пакети передаються за попередньо обчисленим шляхом (так званий тунель MPLS), який розподіляється на всі вузли вздовж шляху перед фактичною передачею. Це, з одного боку, дозволяє уникнути складних пошуків у таблиці маршрутизації, а з іншого – це дозволяє визначити чіткі та добре контрольовані шляхи, якими передаються потоки трафіку. За допомогою обчислення явних тунелів можна реалізувати складні політики інженерного трафіку [28], налаштувати наскрізні віртуальні приватні мережі (VPN) [29] і розробити спеціальні механізми захисту, які швидко відновлюють підключення у разі збою [30].

Як було сказано, хоча існують стандартизовані засоби збирання інформації про стан пристроїв і трафіку, кожен постачальник зазвичай пропонує власні механізми для розповсюдження конфігурацій. Гетерогенність пристроїв, постачальників і версій справді ускладнює

визначення загальної та гнучкої структури, здатної відповідати різним вимогам. Це перешкоджало прийняттю стандартних протоколів, які здебільшого обмежені академічним дизайном із незначним впровадженням.

Протокол NETCONF є прикладом стандартних механізмів, які дозволяють встановлювати, маніпулювати та видаляти конфігурацію мережних пристроїв [31]. Він використовує кодування даних на основі XML для конфігураційних даних, а також повідомлень протоколу. Ключовим аспектом NETCONF є те, що він дозволяє функціонально точно віддзеркалювати рідний інтерфейс командного рядка пристрою та забезпечує стандартний спосіб автентифікації, цілісності даних і конфіденційності. При цьому все залежить від транспортного протоколу, що лежить в основі цієї можливості. Наприклад, з'єднання можуть бути зашифровані за допомогою TLS або SSH, залежно від підтримки пристрою. Разом з NETCONF мова моделювання даних, що визначає семантику операційних і конфігураційних даних, сповіщень і операцій, була визначена через впровадження мови моделювання YANG [32]. Ні NETCONF, ні YANG ніколи не вдасться стати справжнім стандартом, враховуючи складність знайти загальну та гнучку основу, яка відповідає всім вимогам.

Internet Engineering Task Force (IETF) визначила загальну структуру політики для керування, спільного використання та повторного використання політик у незалежній від постачальника, сумісний та масштабований спосіб [33]. Основна інформаційна модель політики (PCIM) – це об'єктно-орієнтована інформаційна модель для подання інформації про політику. Вона визначає два основні архітектурні елементи: точку впровадження політики (PEP) і точку прийняття рішень (PDP). Політики дозволяють оператору вказувати, як мережа має бути налаштована та відстежена за допомогою мови опису. Це дозволяє автоматизувати завдання керування відповідно до вимог, викладених у модулі політики. Структура політики IETF була прийнята галуззю як стандартний підхід до керування політикою, а також була прийнята стандартизацією Проекту партнерства третього покоління (3GPP).

Служба загальної відкритої політики (COPS) – це протокол, який забезпечує модель клієнт/сервер для підтримки контролю політики. Специфікація COPS не залежить від типу політики, що надається (QoS, безпека тощо), але зосереджується на механізмах і угодах, які використовуються для розподілу інформації між PDP і PEP. COPS ніколи не набув широкого застосування, оскільки оператори виявили, що використання бінарних повідомлень ускладнює розробку автоматизованих сценаріїв для простих завдань керування конфігурацією.

Як було сказано раніше, кожен постачальник реалізував власне рішення для збирання, зміни, розповсюдження конфігурацій і оновлень системи. Великі постачальники, такі як Cisco Systems, Juniper Networks, Huawei тощо, надають різні пакети, які варіюються від рішень для простих локальних мереж (LAN) до рішень для масштабування Інтернет-

провайдерів. Так звані системи керування мережею [34] спрощують керування адміністрованою мережею, пропонуючи централізовані рішення, які дозволяють виконувати виявлення пристроїв, моніторинг і керування ними, аналіз продуктивності мережі, інтелектуальні повідомлення та налаштовані оповіщення. Для взаємодії з пристроями вони будуються на основі таких стандартних протоколів, як SNMP або syslog, але часто також використовують спеціальні рішення на основі інтерфейсів командного рядка (CLI), до яких можна отримати доступ через SSH або telnet (застаріле з міркувань безпеки). Наприклад, Cisco Configuration Professional – це інструмент керування пристроями на основі графічного інтерфейсу користувача (GUI) для маршрутизаторів доступу Cisco. Цей інструмент спрощує налаштування маршрутизації, брандмауера, системи запобігання вторгненням (IPS), VPN, уніфікованих комунікацій, глобальної мережі (WAN) і локальної мережі за допомогою простих у використанні майстрів на основі графічного інтерфейсу користувача.

Як було сказано раніше, постачальники та сторонні компанії пропонують портфолію рішень для керування, які варіюються від простого керування мережею для невеликих впроваджень до рішень для масштабування Інтернет-провайдерів, від локальної мережі до мереж центрів обробки даних.

Основна мета цих платформ – запропонувати уніфіковане уявлення про стан мережі та послуг. Ці платформи можуть збирати дані з пристроїв, що належать до домену адміністрування, через SNMP, Syslog, IPFIX і пропрієтарні вирішення. Часто вони впроваджують механізм автоматичного виявлення, щоб знаходити та додавати пристрої до своєї бази колекції, щоб мінімізувати втручання адміністратора. За допомогою графічного інтерфейсу користувача вони забезпечують можливість переглядати статус мережі, показуючи часові ряди завантаження каналу та процесора, розділені за програмами або джерелом-призначенням трафіку. Таким чином, адміністратору пропонується уніфіковане уявлення про стан мережі з можливістю детальніше ознайомитися безпосередньо з графічним інтерфейсом користувача. Вони також можуть виявляти проблеми зі справністю мережного вузла та підключення за допомогою простих алгоритмів на основі порогових значень. У таких випадках можна видати сповіщення, щоб попередити адміністраторів. На рисунку 1.2 наведено приклад архітектури Zabbix.

З точки зору архітектури всі ці платформи схожі. Вони мають проксі-модулі, також звані модулями агентів, для взаємодії з різними протоколами та пристроями для збирання даних, які потім зберігаються в модулі бази даних на основі рішень з відкритим кодом, таких як MySQL, PostgreSQL, або комерційних рішень, таких як Oracle SQL.

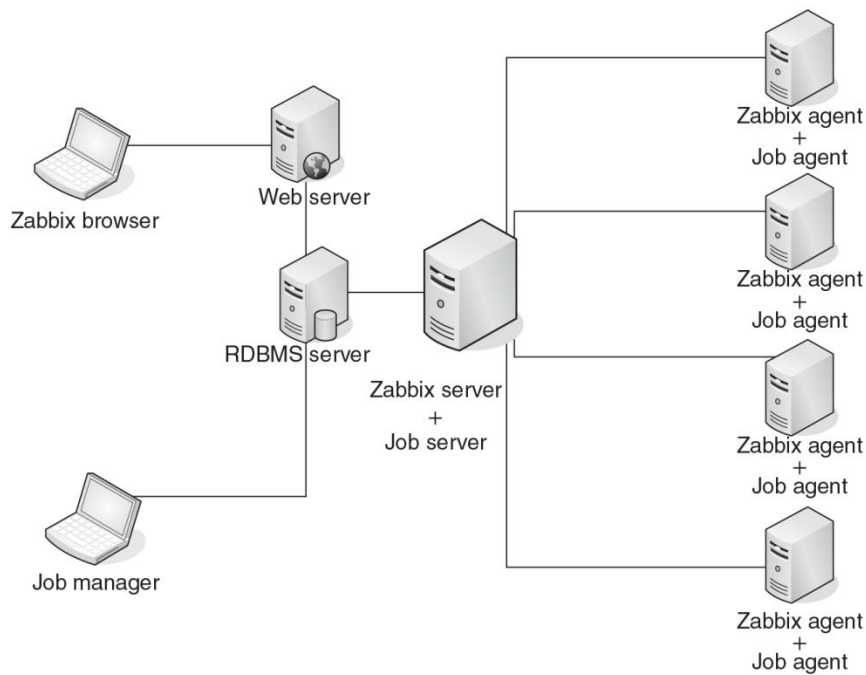


Рисунок 1.2 – Архітектура моніторингу Zabbix

Зазвичай веб-інтерфейс або інформаційна панель дозволяють адміністратору взаємодіяти та переміщатися між даними. Інформаційна панель також може запропонувати можливості конфігурації, зазвичай відкриваючи підключення для керування пристроями. Нарешті, медіа-шлюз дозволяє системі розповсюджувати сигнали тривоги через електронну пошту, службу коротких повідомлень, системи чату, системи продажу квитків тощо.

Деякі платформи є відкритими. Вони дозволяють інтегрувати дані, зібрані з різних платформ, в єдиний централізований центр, але рідко пропонують можливість змінити основну конфігурацію через труднощі взаємодії з різними пристроями. Серед них Zabbix (<https://www.zabbix.com>), Nagios (<https://www.nagios.org>) або Cacti (<https://www.cacti.net>) є найстарішими з більш сучасних рішень, як-от LibreNMS (<https://www.librenms.org>) або Observium (<https://www.observium.org>), що з'являються як нові та більш реактивні рішення.

Запатентовані рішення зазвичай пропонують більше опцій і гнучкості, а також мають можливість змінювати налаштування мережі. Кожен постачальник має портфолію рішень, які відповідають різним сценаріям і розмірам впровадження, від невеликих локальних мереж до національних Інтернет-провайдерів. Доступні також рішення від незалежних постачальників, які зазвичай підтримують багато платформ.

На початку цього розділу ми описали найбільш стандартний підхід до контролю та керування мережею. Надалі ми коротко наводимо найновіші підходи, які все ще досліджуються дослідницькими та технічними спільнотами і розвиток яких швидко відбувається.

Технологія програмно визначеної мережі (SDN) – це підхід до керування мережею, який відокремлює площину керування від площини даних. Дійсно, у оригінальному дизайні Інтернету площина керування, де виконуються протоколи керування та дії керування, тісно вбудована в площину даних, де пакети маршрутизуються та пересилаються. SDN розділяє дві площини, так що комутатори стають чистими пристроями пересилання, тоді як усі операції контролю та керування віднесені до централізованого контролера. Контролер визначає правила пересилання, які потім надсилаються на комутатори, що використовують їх для пересилання пакетів по правильному та бажаному шляху. Це забезпечує динамічну, програмно ефективну конфігурацію мережі для покращення продуктивності мережі та моніторингу. Мартін Касадо в 2007 році подав ідею покладатися на централізований контролер для покращення керування мережею [35]. З тих пір технологія SDN стала основною [36], спочатку підтримуючи мережу кампусу, потім розширивши її підтримку для мереж центрів обробки даних, а останнім часом і в глобальних мережах через SD-WAN [37], приносячи в область WAN переваги відокремлення мережного обладнання від його механізму керування.

Архітектура SDN визначає три площини – додавання площини додатків поверх площини керування. На рисунку 1.3 зображено загальну архітектуру. Програми SDN – це програми, які безпосередньо та програмно передають свої вимоги та бажану поведінку через північний інтерфейс до мережного контролера SDN. Програми отримують абстрактне глобальне уявлення про мережу та пропонують такі розв'язання й дії, як чіткі маршрути, правила фільтрації тощо. Контролер SDN знаходиться між ними. Це логічно централізована сутність, яка транслює вимоги від додатків до фактичних дій, які мають бути реалізовані елементами площини керування, і надає додаткам оновлене загальне уявлення про стан мережі. Логічно централізований, він може бути реалізований розподіленим способом, щоб гарантувати як масштабованість, так і надійність. Він підтримує як концепцію об'єднаних контролерів, кожен з яких відповідає за керування частиною мережі; і ієрархічних контролерів, де контролери вищої ієрархії підсумовують інформацію, отриману нижчими рівнями, і роблять її доступною для програм. У нижній частині площина даних – або Datarath – це логічна мережа пристроїв, які пропонують можливості пересилання та обробки даних. Механізми пересилання даних відповідають за швидке перемикання пакетів. Вони спілкуються з контролером SDN через південний інтерфейс, який визначає стандартні інтерфейси прикладного програмування (API) для обміну інформацією. Функції обробки трафіку реалізують розв'язання на основі корисного навантаження пакетів. Наприклад, вирішення про перемикання можна прийняти з урахуванням адрес як відправника, так і одержувача, увімкнувши маршрутизацію для кожного потоку.

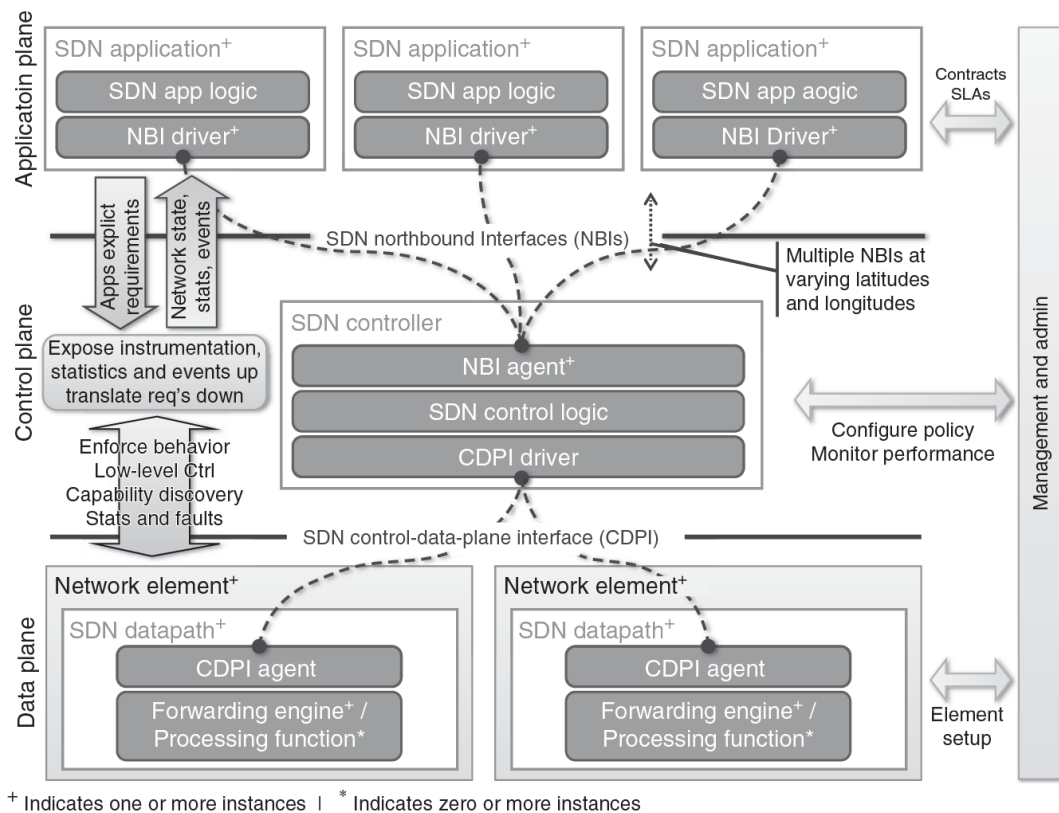


Рисунок 1.3 – Архітектура SDN

Так само розв'язання щодо фільтрації може ґрунтуватися на номерах портів TCP.

SDN часто асоціюється з протоколом OpenFlow [38], який забезпечує віддалений зв'язок з елементами мережної платформи та контролером. Однак для багатьох компаній це більше не є ексклюзивним вирішенням, і тепер доступні такі власні технології, як відкрите мережне середовище та платформа мережної віртуалізації Nicira. Усі вони пропонують стандартний API для зв'язку через південний інтерфейс.

Віртуалізація мережних функцій (NFV) – це мережна архітектура, вона спирається на вершину концепцій віртуалізації [39], яка пропонує можливість віртуалізації мережних вузлів і функцій у будівельні блоки, що їх можна з'єднувати та об'єднувати для створення складніших комунікаційних послуг. Віртуалізована мережна функція (VNF) складається з однієї або кількох віртуальних машин і контейнерів, які запускають спеціальне програмне забезпечення для реалізації мережних операцій у програмному забезпеченні. Брандмауери, контролери списків доступу, балансувальники навантаження, системи виявлення несанкціонованих входів, термінатори VPN тощо можуть бути реалізовані в програмному забезпеченні – без купівлі та встановлення дорогих апаратних рішень.

NFV складається з трьох основних компонентів, як показано на рисунку 1.4: у верхній частині VNF, які будуть реалізовані за допомогою програмного рішення; інфраструктура віртуалізації мережних функцій

(NFVI) знаходиться посередині та пропонує апаратні компоненти, на яких впроваджуються VNF. Вони містять фізичні сервери та мережні пристрої, які створюють інфраструктуру NFV; нарешті, структура NFV Management and Orchestration (MANO) дозволяє керувати платформою, пропонуючи сховища даних і стандартні інтерфейси для обміну інформацією. При побудові складної функції базові блоки можна з'єднати в ланцюг, щоб створити конвеєр обробки.

Це називається «ланцюжком послуг» і дозволяє повторно використовувати високоспеціалізовані та ефективні блоки для створення складних функціональних можливостей.

Враховуючи операції керування, очевидно, що NFV потребує від мережі створення екземплярів, моніторингу, ремонту та виставлення рахунків за послуги, які вона пропонує. NFV справді націлений на сценарій великих операторів, будь то менеджер центру обробки даних або постачальники послуг Інтернету. Ці функції розподіляються на рівень оркестровки, який має керувати VNF незалежно від фактичної апаратної та програмної технології, що знаходиться нижче.

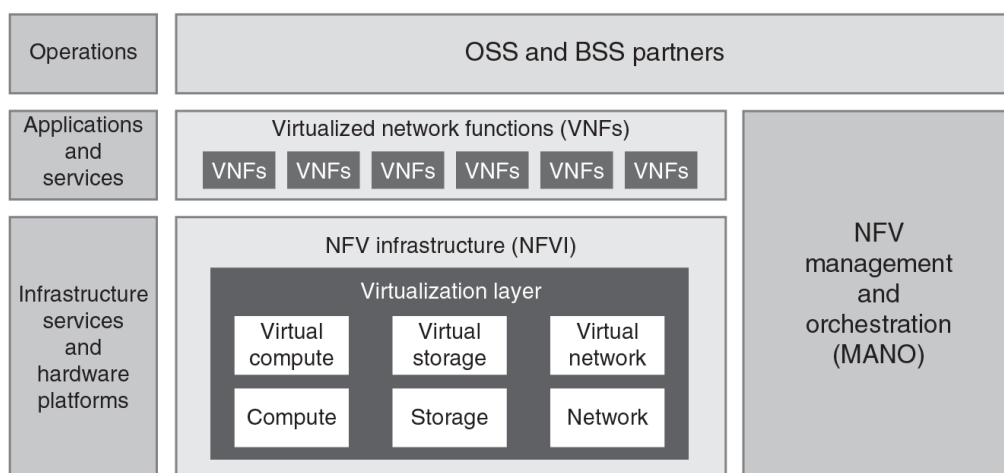


Рисунок 1.4 – Архітектура віртуалізації мережних функцій

NFV – це засіб для зниження витрат і прискорення розробки та впровадження послуг. Замість того, щоб вимагати встановлення дорогого апаратного забезпечення зі спеціальними функціями, постачальники послуг покладаються на недорогі мережні пристрої, системи зберігання та сервери для запуску віртуальних машин, які реалізують бажану мережну функцію.

Коли клієнт запитує мережні функції, постачальник послуг може просто створити нову віртуальну машину для реалізації цієї функції. Це також має перевагу, оскільки зменшує залежність від виділених апаратних пристроїв і покращує надійність за допомогою можливостей міграції, які переміщують служби у разі збоїв або операцій з обслуговування.

Зрозуміло, що NFV потребує стандарту для забезпечення сумісності вирішень. З 2012 року понад 130 провідних мережних операторів світу

нещодавно об'єдналися, щоб створити Групу галузевих специфікацій (ISG) для NFV Європейського інституту стандартів телекомунікацій (ETSI) (<https://www.etsi.org/technologies/nfv>). NFV також є основоположним на арені 5G, де всі передові функції, які пропонує мережа, як-от нарізка мережі, периферійні обчислення або функції децентралізованого керування радіо, реалізовані на основі NFV.

1.2 Огляд штучного інтелекту та машинного навчання

З удосконаленням комп'ютерних і мережних технологій зростає можливість отримувати доступ, зберігати та обробляти величезні обсяги даних із фізичних віддалених/близьких місць. Наприклад, люди зі смартфонами, весь час підключені до різних систем соціальних медіа, можуть обмінюватись текстом, голосом, фотографіями та відео в будь-який час і в будь-якому місці. Зазвичай, це передбачає передавання від гігабайтів до терабайтів даних щодня на платформах соціальних мереж. Ці збережені дані стають корисними, коли їх аналізують і перетворюють на інформацію, наприклад для прогнозування, кореляції тощо. З цією метою штучний інтелект (AI) і машинне навчання (ML) стали методами, які все частіше використовуються протягом багатьох років [1–3].

ШІ здебільшого базується на логіці [4]. Штучний інтелект має на меті змусити комп'ютери робити те ж саме, що й людський розум. Існує кілька основних типів штучного інтелекту: від класичного або символічного штучного інтелекту до машинного навчання, кожен з яких містить багато варіацій. Класичний/символічний ШІ моделює планування та міркування, а також може моделювати навчання. Він базується на принципі машини Тьюрінга в поєднанні з пропозиційною логікою та теорією нейронних синапсів. Складні пропозиції створюються, а дедуктивні аргументи проводяться за допомогою логічних операторів для опису систем міркувань. Експертні системи, бази знань і аргументація на базі випадків є прикладами класичного ШІ. Експертні системи імітують процес прийняття розв'язань людиною-експертом. Програма запитувала експерта в певній галузі, як реагувати в тій чи іншій ситуації, і як тільки це було вивчено для достатнього діапазону ситуацій, неексперти могли отримати поради від цієї програми.

Ці програми будуть використовуватися для створення баз знань, які потім можуть бути використані для систем підтримки прийняття рішень для різних прикладних сфер. Експерти, засновані на кейсах, вирішують нові проблеми, відновлюючи збережені «кейси», що описують подібні попередні епізоди вирішення проблем, і адаптуючи свої рішення відповідно до нових потреб. Протягом багатьох років ми бачили застосування експертних систем і систем аргументації на основі випадків у керуванні мережами та послугами [5–7].

З іншого боку, ML керується даними [8]. Це означає програмування для оптимізації критеріїв продуктивності з використанням прикладів даних або минулого досвіду. Оскільки у ML існує модель, визначена деякими параметрами, то навчання стає виконанням програми для оптимізації параметрів моделі з використанням навчальних даних або минулого досвіду. Випадок минулого досвіду відрізняється від навчання під контролем або без нього, оскільки присвоєння кредитів може затримуватися. Таким чином, не відразу зрозуміло, яку поведінку потрібно підтримувати чи заперечувати. Це питання стосується навчання з підкріпленням. Модель може бути прогностичною, щоб робити прогнози в майбутньому, або вона може бути описовою, щоб отримати знання з даних, або це може бути і те, і інше. ML використовує статистичну теорію, щоб керувати побудовою моделі та зробити висновок на основі навчальних даних або минулого досвіду. Під час навчання ефективні алгоритми потрібні для вирішення проблеми оптимізації, а також для зберігання та обробки даних / минулого досвіду. Крім того, модель, яку вивчають наприкінці навчання, має мати ефективне уявлення та рішення для цілей логічного висновку, можливо, у реальному часі. У деяких застосуваннях ML ефективність моделі навчання та логічного висновку, іншими словами, складність простору та часу, може бути такою ж важливою, як і точність її передбачення. Зростання мережних технологій для легкого доступу до даних, дешевшого доступу до потужності процесора та швидкого доступу до сховища даних уможливило використання алгоритмів ML в керуванні мережами та послугами [9, 10].

Більшість дослідників класифікують алгоритми навчання на три основні типи на основі основних характеристик завдання: (1) контрольоване навчання, (2) неконтрольоване навчання та (3) навчання з підкріпленням. У контрольованому навчанні мета полягає в тому, щоб навчитися відображати вхідні дані на вихідні дані, правильні значення яких надає основна правда (мітка) під час навчання. У неконтрольованому навчанні такої базової істини немає, а є лише вхідні дані. Таким чином, мета полягає в тому, щоб знайти подібність у вхідних даних. У навчанні з підкріпленням, увага зосереджена на ідентифікації системи, яка здатна максимізувати кумулятивну винагороду, отриману при явній взаємодії з середовищем. Однак, незалежно від завдання, ML має такі ключові компоненти, а саме: подання функції вартості та розподіл кредитів. У цьому контексті подання означає навчальну мову, яка використовується для створення розв'язань. Приклади подання нейронної мережі проти подання в індукції дерева розв'язань або інструкцій з простого набору інструкцій. Таким чином, рекурентні подання визначають поточний вихід як функцію попереднього «внутрішнього» стану, а також поточного входу (стану). Якщо подання не підтримує механізми пам'яті, кінцева модель обмежена лише реактивною поведінкою.

Залежно від передбачуваного подання розв'язання може бути важче знайти або дорого оцінити. Підтримка пам'яті була б корисною для завдань, які частково можна спостерігати, але також може зменшити здатність визначити, як приймалися ухвали (прозорість). Функція вартості вказує на те, як оцінюється продуктивність розв'язання, наприклад, класифікація або точність прогнозу, апостеріорна ймовірність або наскільки простим має бути розв'язання. Присвоєння кредиту керує тим, як модифікувати подання, тобто підтримку/заперечення при керуванні процесом пошуку.

Далі ми докладніше обговоримо типи машинного навчання, щоб отримати більше уявлення та зрозуміти їх використання.

Мета контрольованого навчання полягає в тому, щоб навчитися відображати вхідний простір у вихідний простір, де правильні значення надаються мітками, які називаються супервізором.

На рисунку 1.5 показано огляд моделі навчання під наглядом. Якщо вихідні дані є дійсними, то такі задачі ще називають регресією. Інакше їх називають класифікацією, коли система навчання відповідає моделі, яка пов'язує набори (вхідних) екземплярів із мітками, можливо, з відповідною мірою достовірності.

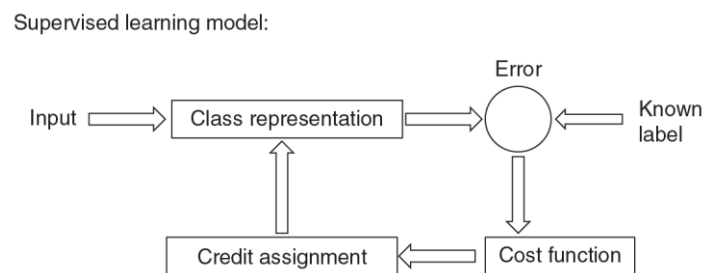


Рисунок 1.5 – Контрольована модель навчання

Після навчання з минулими даними модель вивчає правило класифікації, яке може мати форму If-Then-Else. Наявність такого правила дозволяє нам робити прогнози, якщо майбутнє схоже на минуле. У деяких випадках ми можемо захотіти обчислити ймовірність, тоді класифікація стає вивченням зв'язку між вхідними та вихідними даними. Вивчення правила з даних також дозволяє отримати знання.

У цьому випадку правило є простою (складною) моделлю, що описує дані, і тому модель навчання дає нам уявлення про процес, який лежить в основі даних. Крім того, навчальна модель також виконує стиснення, підбираючи правило до даних. Це дозволяє нам використовувати менше пам'яті для зберігання даних і менше обчислень для обробки. Іншим використанням навчання під наглядом є виявлення артефактів. У цьому випадку після вивчення правила ми зосереджуємося на частинах даних, які не охоплюються правилом. Іншими словами, ми визначаємо випадки, які не

відповідають правилу та/або є винятками з правил. Ці артефакти вказують на аномалії, які потребують подальшого аналізу.

Метою неконтрольованого навчання є виявлення закономірностей у вхідних даних. Припущення полягає в тому, що існує така структура вхідного простору, що певні шаблони трапляються частіше, ніж інші. На рисунку 1.6 показано огляд моделі навчання без нагляду. Таким чином, ми прагнемо ідентифікувати та розрізнити шаблони з різними основними властивостями. Коли це буде досягнуто, ми також зможемо відрізнити типову поведінку від нетипової. Одним із методів досягнення цього є кластеризація. Метою кластеризації є пошук груп вхідних даних. Це можна використовувати для дослідження даних, щоб зрозуміти структуру даних та/або для попередньої обробки даних, де кластеризація дозволяє відобразити дані в новому k -вимірному просторі, де нова розмірність може бути більшою за вихідну розмірність даних.

Однією з переваг неконтрольованого навчання є те, що для нього не потрібні позначені дані.

Позначення даних (отримання фактичної інформації) є дорогавартісним. Таким чином, ми можемо використовувати великі обсяги немічених даних для вивчення параметрів кластера. Ось чому неконтрольоване навчання широко використовується для «виявлення аномалій» при керуванні мережами та послугами [11–13].

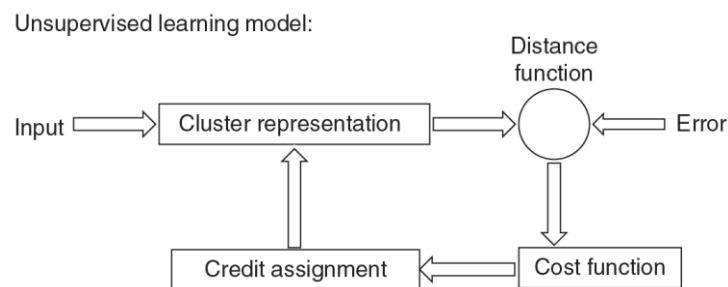


Рисунок 1.6 – Модель навчання без контролю

Мета навчання з підкріпленням полягає в тому, щоб вивчити найкращу послідовність дій (політику) у певному середовищі, щоб максимізувати кумулятивну винагороду. На рисунку 1.7 показано огляд моделі навчання з підкріпленням. У цьому випадку модель навчання з підкріпленням діє як агент прийняття розв’язань, виконуючи дії в середовищі та отримує винагороди/штрафи, намагаючись вирішити проблему. У задачах навчання з підкріпленням середовище перебуває в певному стані (з набору можливих станів) у будь-який момент часу. Інформація про стан може бути повною (марковською) або неповною (немарковською). Агент має набір дій (з набору можливих дій), і коли дія виконується, стан середовища змінюється.

Reinforcement learning model:

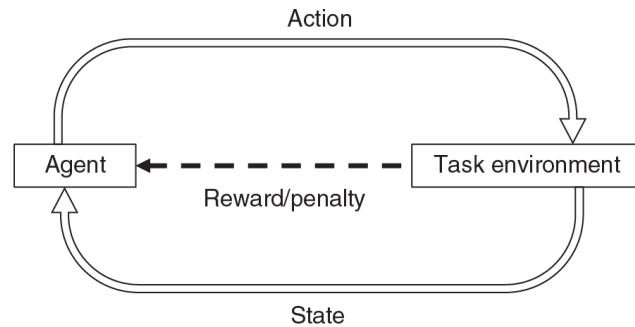


Рисунок 1.7 – Модель навчання з підкріпленням

Таким чином, на відміну від неконтрольованого або контрольованого навчання, навчання з підкріпленням явно взаємодіє з «завданням». Модель будується в інтерактивному режимі з завданням, а не незалежно від нього. На кожному часовому етапі зазвичай передбачається сигнал винагороди, де винагородою може бути просто «ви не зазнали невдачі». Дійсно, може не бути жодної «кінцевої винагороди», окрім максимізації тривалості між відмовами або максимальної кількості маршрутизованих пакетів. У контрольованому навчанні мітка даних чітко вказує нам, що робити. І навпаки, моделі підкріплення можуть спробувати вивчити функцію, що описує відносну «цінність» перебування в кожному стані. Тоді прийняття розв'язань спроститься до визначення дії, яка перемістила поточний стан у наступний стан із найбільшою «цінністю». Таким чином, навчання з підкріпленням також явно бере участь у встановленні порядку, в якому воно піддається стану з завдання. Це знову ж таки відрізняється від навчання під наглядом або без нагляду, у якому, зазвичай, передбачається, що дані відповідають незалежному та ідентично розподіленому припущенню. Крім того, коли повна інформація доступна, агент навчання з підкріпленням може приймати оптимальні рішення лише на основі поточного стану. Однак, коли повна інформація про стан відсутня, агент навчання з підкріпленням додатково має розробити внутрішні моделі стану, які поширюють стан на раніше відвідані значення. Зайве говорити, що ця вимога впливає на прийняте представництво, а також на процес призначення кредиту. Алгоритми навчання з підкріпленням мають ширший спектр застосувань, ніж алгоритми навчання під наглядом, проте їм може знадобитися більше часу для конвергенції, враховуючи, що зворотний зв'язок є менш явним, ніж під час навчання під наглядом і без нього.

Тут потрібно зазначити, що застосування навчання з підкріпленням при керуванні мережами та послугами швидко розвивається, і ми бачимо все більш вражаючі результати в цій галузі [14–16].

Методи штучного інтелекту/ML мають важливий список застосувань у багатьох задачах керування мережами та послугами, охоплюючи (але не обмежуючись цим) класифікацію трафіку/сервісів і прогнозування для

керування продуктивністю; вторгнення, виявлення зловмисного програмного забезпечення та атрибуція для керування безпекою; аналіз першопричини та ідентифікація/прогноз несправності для керування несправностями; і розподіл ресурсів/завдань/призначення для керування конфігурацією. Як обговорювалося раніше, зростання кількості підключених пристроїв, а також нові комунікаційні технології від 5G+ до SDN і NFV переконують дослідників керування мережами та послугами вивчати нові методології в сфері AI/ML [17].

Враховуючи поточний прогрес у мережах/сервісах, штучний інтелект/ML знайшов своє місце в задачах керування продуктивністю завдяки своїй здатності навчатися на основі великих даних для прогнозування різних умов, агрегувати шаблони, визначати тригери для операцій і дій керування. Наприклад, для прогнозування трафіку використовувалися численні програми на основі машинного навчання: від прогнозування часових рядів [18] до нейронних мереж [19, 20] до прихованих моделей Маркова [21] і генетичних алгоритмів [22]. Крім того, у багатьох інших завданнях керування продуктивністю використовуються методи AI/ML для керування трафіком у хмарних і мобільних периферійних обчисленнях, керування мережними ресурсами та розподілом, забезпечення якості обслуговування та контролю перевантаження. Вони використовують можливості методів AI/ML для навчання з часовими і динамічними даними [23–26]. Сучасні приклади таких розробок охоплюють Deep Neural Networks [27], TransferLearning [28], Deep Reinforcement Learning [15, 29] і Stream online learning [30].

Керування безпекою – це ще одна сфера керування мережами/сервісами, яка охоплює широке та раннє схвалення методів AI/ML. Виявлення мережних аномалій є яскравим прикладом, у якому застосовуються методи ML для їхньої здатності автоматично навчатися з даних і «витягувати» шаблони, які можна використовувати для своєчасної ідентифікації мережних аномалій [31]. Для цього застосовуються підходи часової кореляції [32], вейвлет-аналізу [33] та традиційного виявлення точки зміни [34] для створення моделей нормального/зловмисного трафіку, де послідовність дій у часовому вікні використовується для створення профілів за допомогою таких методів кластеризації, як самоорганізовувальні карти [35], K-середні [36] і моделі змішування Гаусса [37]. Крім того, методи AI/ML були застосовані для виявлення мережних вторгнень, охоплюючи, але не обмежуючись цим, дерева розв'язань, еволюційні обчислення, байєсовські мережі, опорні векторні машини та віднедавна глибоке навчання та навчання з підкріпленням [38–43]. Неконтрольоване навчання та потокове онлайн-навчання також використовувалися для завдань безпеки [44, 45].

Інші приклади додатків штучного інтелекту/ML у безпеці – це захист рухомих цілей, виявлення внутрішніх загроз і фільтрація мережного вмісту [46].

При керуванні несправностями прогнозування та діагностика несправностей широке використання отримали методи AI/ML: від онлайн-навчання для виявлення точки зміни до нейронних мереж та прихованих моделей Маркова та дерев прийняття розв'язань, а також кількох алгоритмів неконтрольованого навчання [9–13]. Крім того, інші AI/ML були подані спеціально для прогнозування несправностей, автоматизованого пом'якшення несправностей та аналізу першопричини [54–57].

Застосування методів AI/ML було повільнішим у завданнях керування конфігурацією. Однак, як говорилося раніше, з впровадженням технологій NFV і SDN це змінюється [58–60]. Такі ініціативи, як Intent Based Networking [61] і Zero Touch Networking [62], широко використовують AI/ML у бездротових мережах. Іншими прикладами завдань при керуванні конфігурацією з використанням ML є керування конфігурацією послуг, балансування мережного навантаження та маршрутизація [63–68].

Підводячи підсумок, можна сказати, що за останнє десятиліття методи штучного інтелекту/ML застосовувалися у більшій кількості для кількох завдань керування мережами та послугами [69]. Проте все ще існують проблеми, які потрібно вирішити для успішного використання таких методів у виробничих середовищах. Однією з проблем є отримання високоякісних даних для навчання та оцінювання методів ML для функцій керування мережею та послугами. Навіть незважаючи на те, що мережних/сервісних даних у реальному світі багато і вони різноманітні, здебільшого важко отримати такі дані на основі реальних даних. У свою чергу, це не лише створює труднощі для оцінювання методів AI/ML, але й виступають на перше місце проблеми щодо конфіденційності та довіри. Ще одна проблема полягає в тому, що в сучасних мережах/сервісах дані генеруються неперервно у великому обсязі та швидкості. Вони характеризуються як стаціонарною, так і нестаціонарною поведінкою; постійно розвиваються, оскільки з часом впроваджуються нові протоколи та технології. Усе це відображається в даних в одній формі як поступові зміни в поведінці користувача/системи або як раптові зміни, можливо, через несправний пристрій або атаку на відмову на обслуговування певної мережі чи служби. Це означає, що методи AI/ML потребують враховувати цю динаміку та зміни, навчатися за вищезазначених умов, щоб забезпечити успішне впровадження.

Ще одна проблема полягає в потребі експертів-людей (від мережних інженерів до аналітиків безпеки та менеджерів мереж/сервісів) для вивчення системи та інструментів на основі AI/ML. Це потребує прозорих методів AI/ML для залучення експертів. Це надзвичайно важливо для широкого та успішного впровадження методів AI/ML при керуванні мережами та послугами.

Нарешті, ці виклики також створюють можливості у вигляді потреби в прозорих і надійних методах на основі AI/ML для керування мережею та послугами. З цією метою ми вже почали спостерігати застосування

потокowego навчання, змагального навчання та перенесення навчання в мережу й рішення для керування послугами. Крім того, дослідження прозорих, безпечних і надійних методів AI/ML отримали значний імпульс в ML. Враховуючи масштаб і динаміку сучасних мереж/сервісів, ми передбачаємо, що застосування методів AI/ML ставатиме все більш і більш повсюдним і центральним для операцій і керування майбутніми сервісами та мережами. У наступних підрозділах ми познайомимося з поточним станом і новими тенденціями застосування AI/ML при керуванні мережами та послугами.

2 ПРОГРАМНІ ТЕХНОЛОГІЇ В РАДІОМЕРЕЖАХ

2.1 Керування віртуалізованими мережами та службами за допомогою машинного навчання

Віртуалізація викликає революційні зміни в мережній індустрії, подібній комп'ютерній індустрії 1980-х років. Дійсно, перед появою сумісних з IBM і Windows індустрія мейнфреймів наприкінці 1970-х і на початку 1980-х років була закрита вертикально інтегрованим спеціалізованим апаратним забезпеченням, операційною системою та програмами – усе від одного постачальника. Революція сталася, коли почали з'являтися відкриті інтерфейси, індустрія стала горизонтальною, а інновації вибухнули. Подібна революція відбувається в мережній індустрії, яка раніше мала «мейнфрейм» мислення, що покладалося на приватні та вертикально інтегровані вирішення від постачальників. Очікується, що віртуалізація мережі (NV) і надання відкритих інтерфейсів для мережного програмування сприятимуть інноваціям і швидкому розгортанню нових мережних послуг.

Ідея NV набрала обертів для вирішення проблеми деградації Інтернету шляхом створення радикально інших архітектур [1]. Сучасний Інтернет страждає від деградації, оскільки розмір і жорсткість Інтернету ускладнюють впровадження нових мережних технологій [2]. Наприклад, перехід від Інтернет-протоколу версії 4 (IPv4) до IPv6 розпочався більше десяти років тому, тоді як рівень впровадження IPv6 все ще є значно низьким, як повідомляють основні постачальники послуг (тобто менше 30% користувачів Google перейшли на IPv6 [3]). У застарілому Інтернеті стає дедалі важче дотримуватись нових вимог щодо якості обслуговування (QoS) додатків щодо пропускнуєї спроможності, надійності та затримки. NV вирішує проблему деградації, дозволяючи співіснувати декільком віртуальним мережам (VN), кожна з яких налаштована на певну мету в спільному Інтернеті. Незважаючи на те, що ідея NV виникла для вирішення проблеми деградації Інтернету, NV було прийнято як диверсифікувальний атрибут різних мережних технологій, наприклад, бездротові [4], радіодоступ [5], оптичні [6], центри обробки даних (DC) [7], хмарні обчислення [8], сервіс-орієнтовані [9], програмно-визначені мережі (SDN) [10, 11] та Інтернет речей (IoT) [12].

Іншим плідним застосуванням віртуалізації в мережі є впровадження віртуалізованих мережних служб через віртуалізацію мережних функцій (NFV).

NFV відокремлює мережні чи сервісні функції від основного апаратного забезпечення та реалізує їх у вигляді програмних пристроїв, які називаються віртуальними мережними функціями (VNF), на віртуалізованому стандартному апаратному забезпеченні. Численні сучасні

VNF вже показали потенціал для досягнення майже апаратної продуктивності [13, 14].

Крім того, NFV надає широкі можливості для оптимізації мережі та зниження витрат. По-перше, мережні або сервісні функції на основі апаратного забезпечення супроводжуються високими капітальними витратами, які можна зменшити шляхом розгортання VNF на стандартних серверах. По-друге, апаратні пристрої зазвичай розміщуються у фіксованих місцях, тоді як у NFV VNF можна розгорнути на будь-якому сервері в мережі. Розташування VNF можна визначати інтелектуально, щоб задовольнити динамічний попит на трафік і краще використовувати мережні ресурси. NFV відкриває можливість одночасно оптимізувати розташування VNF і шляхи маршрутизації трафіку, що може значно знизити експлуатаційні витрати мережі. Нарешті, апаратні функції важко масштабувати, тоді як NFV пропонує економічно ефективне масштабування VNF за вимогою.

Сервісно-функціональний ланцюг (SFC) – це впорядкована послідовність VNF, що входить до складу конкретного сервісу [15]. Наприклад, у типовій стаціонарній мережі трафік із сервера проходить через систему виявлення вторгнень (IDS), брандмауер і транслятор мережних адрес (NAT), перш ніж досягти Інтернету.

Віртуалізація мереж і послуг полегшує нову бізнес-модель, а саме: мережу як послугу (NaaS), яка забезпечує поділ між додатками, послугами та мережами, які їх підтримують [16]. Оператори мереж можуть прийняти модель NaaS, щоб розділити свої фізичні мережні ресурси на кілька VN (також званих мережними зрізами) і здати їх в оренду постачальникам послуг [17].

У свою чергу, постачальники послуг використовують віртуальні мережі, щоб пропонувати послуги з різноманітними вимогами до QoS без будь-яких інвестицій у створення та управління фізичною інфраструктурою.

Ідеальним втіленням моделі NaaS є нарізка мережі для мобільних мереж п'ятого покоління (5G). Використовуючи нарізку мережі, одну фізичну мережу 5G можна розділити на кілька ізольованих логічних мереж різного розміру та структури, призначених для різних типів послуг. Ці «самодостатні» віртуальні мережі мають бути достатньо гнучкими, щоб одночасно враховувати різноманітні бізнес-випадки використання від кількох постачальників послуг у загальній мережній інфраструктурі та створюватися на вимогу (відповідно до вимог) постачальників послуг.

Переваги віртуалізованих мереж і послуг пов'язана з додатковими проблемами керування для мережних операторів. По-перше, мережний оператор має організувати VN/мережні сегменти таким чином, щоб вони могли співіснувати в одній інфраструктурі, не впливаючи один на одного. Отже, потрібно прийняти розумні рішення щодо оркестровки, щоб створити VN, які задовольняють вимоги різноманітних користувачів і програм,

забезпечуючи при цьому бажане використання ресурсів. Це також передбачає налаштування великої кількості віртуальних екземплярів та їхніх робочих параметрів.

Початкову оркестровку та конфігурацію потрібно адаптувати, щоб впоратися зі змінними в часі вимогами трафіку та зміною станів мережі. По-друге, доданий рівень віртуалізації вводить нові поверхні атак і збоїв у різних адміністративних і технологічних сферах. Наприклад, будь-який збій базового фізичного ресурсу може поширюватися на розміщені віртуальні ресурси, хоча обернене не завжди достовірно. Подібним чином механізм усунення та пом'якшення наслідків для однієї VN не має загрожувати роботі співіснуючих VN. Ці різноманітні виклики потребують автоматизованого керування, яке не може задовольнитися традиційним реактивним підходом до керування людиною в циклі. Керування віртуальними мережами має бути розумним, щоб використовувати величезний обсяг оперативних даних, створених у живій мережі, і приймати автоматизовані рішення для різних операційних і управлінських дій. Таким чином, штучний інтелект (AI) і машинне навчання (ML) можуть відігравати ключові ролі для реалізації автоматизації контролю та керування для віртуальних мереж і їхніх послуг [18, 19].

Методи AI та ML широко використовувалися для вирішення мережних проблем в останні кілька десятиліть [18, 19]. Однак, коли справа доходить до керування віртуалізованою мережею, відсутність реального розгортання віртуалізованих служб перешкоджає застосуванню методів штучного інтелекту та машинного навчання. Незважаючи на це, останнім часом відбувся сплеск дослідницьких зусиль, спрямованих на використання ML для вирішення складних проблем у середовищі NV. У цьому підрозділі підсумовуються сучасні дослідження та окреслюються потенційні шляхи застосування методів штучного інтелекту та машинного навчання у віртуалізованій мережі та керуванні послугами. Решта організована таким чином. Ми надаємо детальний огляд технологій віртуалізованих мереж і послуг у підрозділі 2.2. У підрозділі 2.3 ми наводимо найсучасніші дослідження, які застосовують штучний інтелект і машинне навчання у трьох основних підсферах віртуалізованих мереж і послуг, а саме: NV, NFV і нарізку мережі. Ми завершуємо даний розділ підрозділом 2.4 з коротким підсумком і окреслюємо можливі шляхи дослідження для вдосконалення сучасного рівня застосування штучного інтелекту та машинного навчання для керування віртуалізованими мережами та послугами.

Віртуалізація в мережах не є новою концепцією. Віртуальні канали в телекомунікаційних мережах на основі X.25 (наприклад, мережах банкоматів) дозволяють кільком користувачам спільно використовувати великий фізичний канал. Віртуальні локальні мережі (VLAN) поділяють фізичну локальну мережу (LAN) на кілька логічних локальних мереж з підвищеним рівнем довіри, безпеки та ізоляції. Подібним чином віртуальні приватні мережі (VPN) пропонують спеціальний зв'язок, який з'єднує

кілька територіально розподілених сайтів через приватні та безпечні тунелі через загальнодоступні мережі зв'язку (наприклад, Інтернет). Накладені мережі (наприклад, PlanetLab) створюють віртуальні топології поверх фізичної топології іншої мережі. Накладення зазвичай реалізуються на прикладному рівні, хоча існують різні реалізації на нижніх рівнях мережного стеку. Ці технології розгортають вузькі виправлення конкретних проблем без цілісного уявлення про взаємодію між співіснуючими віртуальними мережами. Тому в цьому підрозділі ми пропонуємо вичерпний огляд різних технологій, які дозволяють віртуалізувати мережі та служби.

Мережна функція (NF) – це функціональний блок у мережній інфраструктурі, який має чітко визначені зовнішні інтерфейси та функціональну поведінку [13]. NF у традиційних дротових мережах можна класифікувати за двома категоріями: функції пересилання та додаткові функції. Функції пересилання, такі як маршрутизатори, комутатори та транспондери, забезпечують пересилання даних по мережному шляху.

З іншого боку, додаткові функції, такі як протокол динамічної конфігурації хоста (DHCP), трансляція мережних адрес (NAT), універсальна функція Plug and Play (UPnP), брандмауер, оптимізатор і глибокі інспектори пакетів (DPI), пропонують додаткові можливості до шляху пересилання даних. Подібним чином NF в мобільних мережах класифікуються на два класи: функції мережі радіодоступу (RAN) і основні функції. Ми розглянемо детальніше RAN і основні функції пізніше, коли будемо розглядати нарізку мережі. У цьому підрозділі ми розглядаємо два популярних методи віртуалізації NF, наведені нижче (на рисунку 2.1).

Розбиття – це зручний спосіб створення кількох віртуальних об'єктів на одному мережному пристрої (наприклад, маршрутизаторах і комутаторах), які забезпечують функції пересилання.

Розподіл ресурсів може бути досягнутий шляхом жорсткого розподілу (тобто виділені порти комутатора, ядра ЦП, карти) або м'якого розділення (тобто обмеження виконання ЦП, маршрутизація та розділення таблиці пересилання). Жорстке розділення забезпечує чудову ізоляцію, але для його реалізації потрібна велика кількість обладнання, а м'які розділені екземпляри можуть не забезпечувати найвищий рівень ізоляції та безпеки через їх спільний характер.

Маршрутизатор із жорстким розділенням, який називається логічним маршрутизатором (LR), може працювати на різних процесорах та різних картах пристрою маршрутизатора. Усі базові апаратні та програмні ресурси, зокрема мережні процесори, інтерфейси та таблиці маршрутизації і пересилання, призначені для LR. Прикладами LR є «захищені системні домени» від Juniper Networks або «логічні маршрутизатори» від Cisco Systems. Апаратно розділені маршрутизатори розгортаються, в основному, в точках присутності (PoP) операторів мережі, щоб заощадити простір і енергію, а також зменшити витрати на керування.

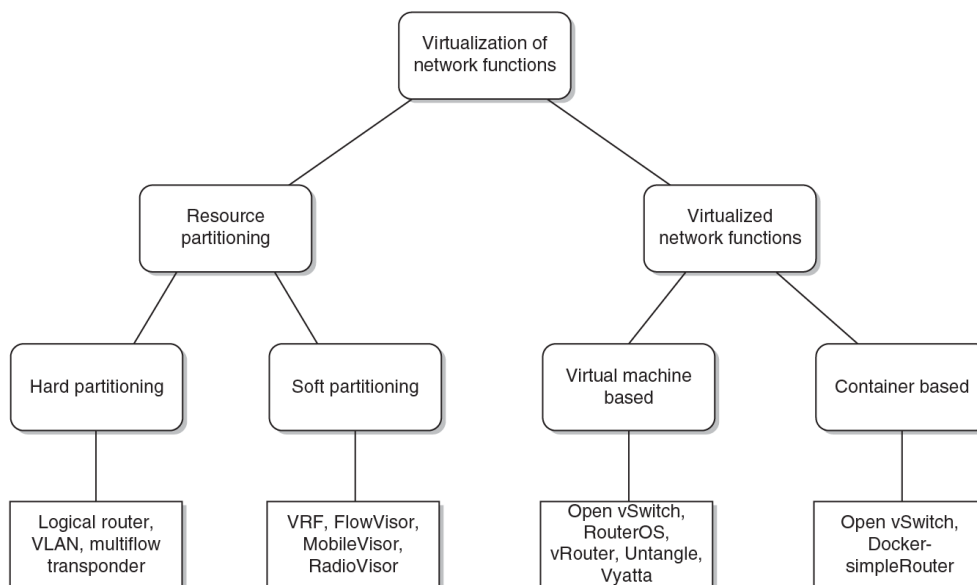


Рисунок 2.1 – Технології віртуалізації мережних функцій з прикладами

Подібним чином мережі VLAN поділяють фізичний комутатор на кілька логічних комутаторів шляхом групування портів на комутаторі. Спосіб групування комутатором залежить від реалізації, але поширеним рішенням є тегування комутатором кожного кадру ідентифікатором VLAN, коли він надходить на порт. Коли кадр надсилається до іншого порту, лише порти, налаштовані за допомогою ідентифікатора VLAN, що міститься у кадрі, виведуть пакет. VLAN також може охоплювати кілька взаємопов'язаних комутаторів за стандартом IEEE 802.1Q. Обмеженням VLAN є його низька масштабованість, насамперед через максимум 4094 VLAN у мережі рівня 2.

Для підтримки більшої кількості VLAN у ширококомовному домені розроблено VXLAN для великих середовищ DC з декількома клієнтами. В оптичному діапазоні багатопотокові транспондери можна використовувати для створення низки субтранспондерів із пулу апаратних ресурсів [6]. Ці субтранспондери можна використовувати для передачі різних потоків, що надходять від одного інтерфейсу маршрутизатора, використовуючи ідентифікатори потоку.

Приклади м'якого розділення містять віртуальну маршрутизацію та переадресацію (VRF), яка дозволяє одночасно існувати декільком екземплярам таблиць маршрутизації та пересилання в межах одного маршрутизатора. Різні таблиці маршрутизації та пересилання можуть підтримуватися одним процесом або кількома процесами (наприклад, один процес для кожної таблиці маршрутизації та пересилання).

Протоколи маршрутизації мають враховувати, що певні маршрути можуть бути розміщені лише в певних VRF. Протоколи маршрутизації керують цим шляхом однорангового перегляду в обмеженій топології, де екземпляр протоколу маршрутизації у VRF взаємодіє з іншими

екземплярами в тому самому VN. Іншим прикладом м'якого розділення є FlowVisor, який розрізає простір потоку комутаторів OpenFlow на основі полів відповідності OpenFlow, таких як порт комутатора, MAC-адреси та IP-адреси. FlowVisor діє, в основному, як проксі між комутаторами та контролерами OpenFlow і перехоплює повідомлення між ними. Абстрагуючи канал керування OpenFlow, FlowVisor забезпечує механізми для пропускнуої спроможності, перемикання центрального процесора та ізоляції потокового простору.

Основна ідея VNF полягає в тому, щоб відокремити фізичне мережне обладнання від функцій, які на ньому виконуються. VNF – це реалізація NF, яка розгорнута на таких віртуальних ресурсах, як віртуальна машина (VM) або контейнер [13]. Один VNF може складатися з кількох внутрішніх компонентів, і, отже, його можна розгортати на кількох віртуальних машинах/контейнерах, у цьому випадку кожна віртуальна машина/контейнер містить один компонент VNF.

Наприклад, віртуальний маршрутизатор (vRouter) – це програмна функція, яка реалізує функціональні можливості IP-маршрутизації рівня 3 у програмному забезпеченні. Базові фізичні ресурси використовуються спільно з іншими спільно розміщеними віртуальними машинами. У добре реалізованому vRouter користувачі можуть бачити та змінювати лише конфігурацію та статистику для «свого» маршрутизатора.

Приклади vRouter охоплюють Alpine Linux, Mikrotik RouterOS, Brocade vRouter, Untangle і Vyatta. Так само віртуальний комутатор (vSwitch) – це програмна емуляція фізичного комутатора, який виконує такі функції, як комутація трафіку, мультиплексування та планування. Він визначає, які віртуальні машини логічно підключені до кожного з його віртуальних портів, і використовує цю інформацію для перенаправлення трафіку на правильні віртуальні машини. Приклади vSwitch охоплюють Open vSwitch, Cisco Nexus 1000v і віртуальний комутатор VMware.

Через різноманітність доданої вартості NF можуть формувати різні види VNF на основі різних мережних рівнів. Навіть для кожного типу NF може бути кілька реалізацій з різними функціями від різних постачальників. Наприклад, віртуальний NAT, реалізований VMware, надає віртуальним машинам можливість спілкуватися з хостом, тоді як NAT, реалізований NFWare, розширено до рівня оператора. Повний перелік продуктів VNF можна знайти в [14].

Існують плюси та мінуси розгортання VNF поверх віртуальної машини або контейнера. У випадку віртуальних машин уся операційна функція віртуальної машини повністю ізольована від функції хоста та інших гостьових віртуальних машин. Таким чином, віртуалізація на основі віртуальних машин забезпечує сильнішу ізоляцію між віртуальними машинами та фізичною машиною й вважається більш безпечним і надійним рішенням. Однак віртуалізація на основі VM страждає від масштабованості та проблем із продуктивністю через накладні витрати на емуляцію повної

комп'ютерної машини у VM. На відміну від цього, контейнери не потребують апаратного опосередкування та працюють більш ефективно поверх ОС хоста, тоді як кожна віртуальна машина працює як незалежна ОС. Отже, контейнери можна використовувати для розгортання VNF більш гнучким способом, але зі зниженим рівнем ізоляції та безпеки.

Технології віртуалізації каналів дозволяють створювати віртуальні канали, які можуть з'єднувати фізичні або віртуальні NF. Віртуальне посилення може складатися з одного фізичного посилення або може охоплювати послідовність фізичних посилення, що утворюють шлях. У цьому підрозділі ми розглядаємо дві популярні технології віртуалізації мережних посилення (зображені на рисунку 2.2).

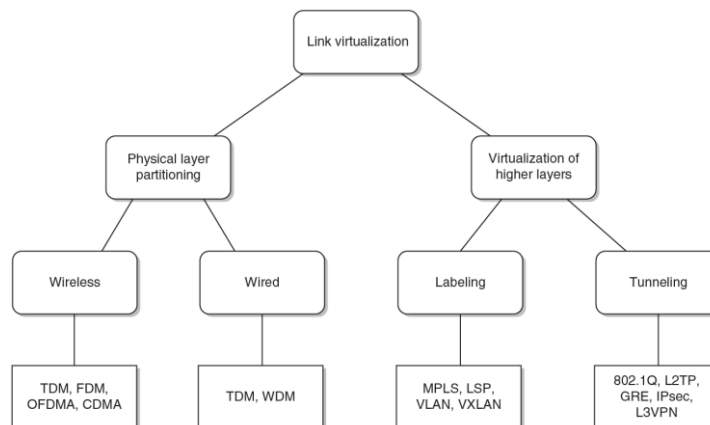


Рисунок 2.2 – Технології віртуалізації потоків даних з прикладами

Використовуючи різні технології мультиплексування, дротове (наприклад, оптоволокло, мідний кабель) або бездротове (наприклад, бездротовий спектр) фізичне середовище можна розділити на окремі канали або часові інтервали. Набір каналів або часових інтервалів потім призначається віртуальному каналу з певною швидкістю передачі даних, щоб у відправника та одержувача віртуального каналу створювалася ілюзія, що вони володіють фізичним носієм. Тип техніки мультиплексування залежить від властивостей фізичного середовища, пов'язаних обмежень і недоліків. Наприклад, бездротове з'єднання може бути розділене за допомогою мультиплексування з часовим поділом (TDM), мультиплексування з частотним поділом (FDM) або за допомогою множинного доступу з кодовим поділом (CDMA). Комбінація різних методів мультиплексування також може бути застосована для досягнення більшої пропускної спроможності, наприклад, для широкосмугових бездротових мереж. Наприклад, множинний доступ з ортогональним частотним поділом каналів (OFDMA) можна описати як комбінацію множинного доступу FDM і TDM, де ресурси розділені як у часовій, так і в частотній областях, а слоти призначаються вздовж індексу символу OFDM, а також індексу піднесучої OFDM.

У волоконно-оптичних комунікаціях мультиплексування за довжиною хвилі (WDM) – це технологія, яка мультиплексує кілька сигналів оптичної несучої на світловий промінь (тобто набір об'єднаних оптоволоконних каналів) за допомогою різних довжин хвиль (тобто кольорів) лазерного світла. Це схоже на FDM, оскільки довжина хвилі та частота передають однакову інформацію. Мультиплексування фізичного рівня забезпечує жорстке розділення та кращу ізоляцію між віртуальними з'єднаннями, оскільки ресурси фізичного середовища призначаються віртуальним з'єднанням спеціально.

На вищих рівнях (наприклад, рівнях зв'язку, мережі або додатків) розділення ресурсів зв'язку досягається шляхом виділення певної смуги пропускання (тобто швидкості передачі, пропускну здатності каналу) для віртуального каналу. Таке розділення може бути забезпечене обмеженням швидкості або виділенням відповідної кількості черг каналів і буферів каналів. Оскільки віртуалізація на вищих рівнях досягається за допомогою м'якого розподілу ресурсів зв'язку, ізоляція між віртуальними з'єднаннями є особливо важливою. Для забезпечення ізоляції між віртуальними посланнями використовуються два популярні методи: (1) маркування та (2) тунелювання.

Позначення передбачає вказання певних полів (наприклад, тегів, ідентифікаторів тощо) у заголовку пакета, які служать для ідентифікації та ізоляції віртуальних послань. Наприклад, в VLAN застосовують теги до мережних пакетів, і обробляють ці теги в комутаторах, створюючи зовнішній вигляд і функціональність мережного трафіку, який фізично знаходиться в одній мережі, але діє так, ніби він розділений між окремими VN. Подібним чином технології багатопроTOCOLЬНОЇ комутації міток (MPLS) і комутації міток (LSP) можуть використовуватися для визначення шляху пакетів даних. У MPLS мітки ідентифікують віртуальні зв'язки (шляхи) між несуміжними NF. Це потребує, щоб маршрутизатори з підтримкою MPLS (наприклад, маршрутизатори з комутацією міток) пересилали пакети до вихідного інтерфейсу лише на основі значення мітки, на відміну від використання IP-адрес у традиційних маршрутизаторах.

Тунелювання – це популярний метод віртуалізації каналів, який використовується багатьма такими різними технологіями, як VPN і VLAN. Це забезпечує ізоляцію трафіку від кількох віртуальних мереж, що транспортуються через спільну мережу. При цьому також забезпечується пряме з'єднання між мережними пристроями, які фізично не є суміжними. Тунелювання виконується за допомогою інкапсуляції та інколи методів шифрування. Існують різні технології тунелювання, зокрема IEEE 802.1Q, протокол тунелювання рівня 2 (L2TP), загальна інкапсуляція маршрутизації (GRE), безпека Інтернет-протоколу (IPsec) і віртуальна приватна мережа рівня 3 (L3VPN).

Як повідомлялося в попередніх підрозділах, і NF, і послання можуть бути незалежно віртуалізовані, не звертаючи уваги один на одного. Також

можна віртуалізувати лише NF та використовувати невіртуалізовані послання для підключення VNF і навпаки. При цьому NV прагне створити фрагменти мережі, тобто VN на конкретному мережному рівні. Наприклад, VN на IP-рівні складається з vRouters/vSwitch і накладених IP-каналів, що з'єднують їх, тоді як VN на оптичному рівні з'єднує мультипотоківі транспондери через оптичні шляхи світла. Потрібно вказати, що дана віртуальна мережа має мати власні ресурси, зокрема власне уявлення про топологію мережі, власні частини пропускної спроможності каналу, виділені ресурси ЦП у NF та власні фрагменти ЦП, таблиці пересилання/маршрутизації в комутаторах і маршрутизатори.

Такого цілісного NV можна досягти за допомогою мережних гіпервізорів, які абстрагують фізичну мережу (наприклад, канали зв'язку, елементи мережі та функції керування) у логічно ізольовані VN [11]. Низка таких мережних гіпервізорів, як OpenVirteX, FlowVisor, OpenSlice, MobileVisor, RadioVisor і Hyper-Flex, були розроблені для різних мережних технологій. Більш детальний огляд гіпервізорів NV здійснено в [11].

Розрізання мережі розширює концепцію NV у контексті мобільних мереж 5G з двох точок зору. По-перше, сегмент мережі 5G – це наскрізна (E2E) VN, яка охоплює кілька технологічних і адміністративних сегментів мережі (наприклад, бездротове радіо, транспортні мережі доступу/основні, граничні обчислення з множинним доступом [MEC] і центральні DC), тоді як традиційний VN стосується лише однієї такої конкретної мережної технології, як дротовий транспорт або бездротова мережа. Приклади сегментів мережі показані на рисунку 2.3, де темно-сірий фрагмент мережі проходить до центрального DC, а пунктирний світло-сірий фрагмент мережі закінчується в центральному офісі мобільної мережі.

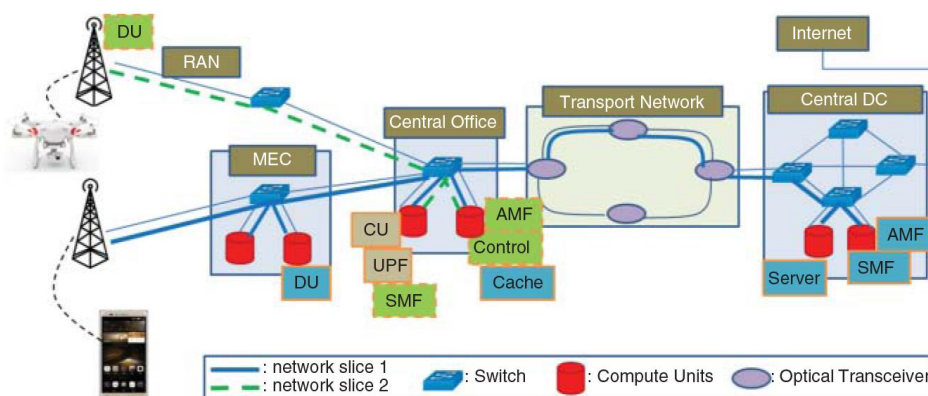


Рисунок 2.3 – Приклади зрізів мережі

Точка зору мережних сегментів E2E пропонує більше можливостей для оптимізації розгортання мережних сегментів і відповідає детальним вимогам QoS. По-друге, нарізка мережі дозволяє віртуалізувати RAN і базові NF і внести їх у мережний зріз, який зазвичай не враховується звичайними VN.

Віртуалізація RAN і основних NF забезпечує більш гнучкий спосіб створення, роботи, керування та видалення сегментів мережі на вимогу. Це також дозволяє розгортати ці VNF з відповідною потужністю в потрібному місці, щоб відповідати суворим вимогам (наприклад, затримка E2E), які висуваються послугами 5G.

Давайте тепер розглянемо детальніше RAN і основні NF. Найпоширенішими функціями RAN, відповідальними за обробку базової смуги, є: протокол адаптації даних служби (SDAP), керування радіоресурсами (RRC), протокол конвергенції пакетних даних (PDCP), керування радіолінією (RLC), керування доступом до середовища (MAC) і (PHY) функції фізичного рівня. У традиційних мережах мобільного зв'язку блоки базової смуги частот (BBU), розташовані разом з антенами, відповідають за виконання RAN NF. Однак в архітектурі 5G RAN передбачається, що ці NF будуть віртуалізовані та розміщені на стандартних серверах, розгорнутих на антенних майданчиках або MEC. Через суворі вимоги щодо синхронізації деяких NF, NF RAN згруповані в дві сутності: центральний блок (CU) і розподілений блок (DU) [20]. DU містить такі критично важливі за часом функції, як MAC, RLC і PHY, і обслуговує низку мобільних користувачів у зоні покриття DU. З іншого боку, CU може приймати такі стійкі щодо часу функції, як SDAP, PDCP і RRC, і може обслуговувати кілька DU. CU можна також розглядати як агреговані VNF і розгортати на віртуальних машинах/контейнерах та серверах, розташованих на антенних майданчиках або MEC.

Подібним чином була запропонована нова базова мережна архітектура для мобільних мереж 5G, а саме: ядро наступного покоління (NG), яке розділяє поточні функції Evolved Packet Core (EPC) на більш детальні NF [20]. Найпомітніші NF в ядрі NG такі: функція керування доступом і мобільністю (AMF), функція керування сеансами (SMF), функція контролю політики (PCF), функція площини користувача (UPF) і уніфіковане керування даними (UDM). Ці основні NF можна також розглядати як VNF і легко розгортати у віртуалізованому середовищі. Перевага цієї сервісо-орієнтованої RAN і архітектури ядра полягає в тому, що вона дозволяє спільно використовувати детальні VNF між сегментами мережі без шкоди для вимог до продуктивності та якості обслуговування. Наприклад, на рисунку 2.3 темно-сірий і пунктирний світло-сірий сегменти мережі спільно використовують CU і UPF NF, задіюючи повністю виділені RAN і базові NF а також їхні прикладні функції (наприклад, кеш, керування або сервер). Подібним чином, такі функції рівня керування, як RLC, MAC, AMF і PCF, можуть бути спільними між сегментами під час використання виділених UPF, зокрема PDCP і UPF. Нарешті, сегменти мережі, які потребують найвищого рівня безпеки (наприклад, громадська безпека або сегмент першої служби реагування), можуть використовувати спеціальні VNF, які не надаються іншим.

SDN має потенціал для спрощення конфігурації мережі та зменшення складності керування. На відміну від сучасних мереж, де функції керування та пересилання тісно пов'язані та вбудовані в кожний мережний пристрій (тобто комутатори та маршрутизатори), SDN акумулює функції керування в логічно централізованій і програмованій площині керування, яка відокремлена від площини пересилання. Площина керування реалізована в програмному забезпеченні (тобто контролері SDN) на одному або кількох виділених комп'ютерних серверах, має вигляд глобальної мережі та забезпечує уніфікований інтерфейс для налаштування та керування мережею. З іншого боку, пересилання пакетів залишається відповідальністю комутаторів/маршрутизаторів і реалізовано на звичайному обладнанні.

Керування та оркестровка (MANO) є квінтесенцією для розкриття повного потенціалу NV, який охоплює безперебійну роботу та ефективне надання послуг. OpenStack – це платформа хмарних обчислень з відкритим вихідним кодом, яка контролює великі пули віртуальних ресурсів для створення та керування приватними/публічними хмарами.

Однак із появою NFV OpenStack став ключовим компонентом у NFV MANO як менеджер віртуальної інфраструктури (VIM). Він відповідає за динамічне керування апаратними ресурсами інфраструктури віртуалізації мережних функцій (NFVI) (тобто обчислювальними, сховищем і мережними) і програмними ресурсами (тобто гіпервізорами), пропонуючи високу доступність і масштабованість. OpenStack також підтримує додаткові функції в NFVI, зокрема ланцюжок функцій служби та нарізку мережі. Відкрита платформа для віртуалізації мережних функцій (OPNFV), платформа операторського рівня з відкритим кодом, також використовує OpenStack як рішення VIM [21].

Платформа автоматизації відкритих мереж (ONAP) і MANO з відкритим кодом (OSM) є двома відомими ініціативами NFV MANO. ONAP, проєкт з відкритим вихідним кодом, організований Linux Foundation, пропонує керувану політикою оркестровку як фізичних, так і віртуалізованих NF у реальному часі, щоб полегшити ефективне й автоматизоване надання послуг на вимогу та підтримувати керування їхнім життєвим циклом. Усі компоненти ONAP пропонуються як контейнери Docker, що дозволяє інтегрувати їх у різні операторські середовища. Він також забезпечує інтеграцію з кількома VIM, менеджерами VNF і контролерами SDN. ONAP складається, в основному, з двох компонентів: (1) часу розробки та (2) часу виконання, кожен з яких має підкомпоненти.

Компонент часу розробки ONAP пропонує середовище проєктування та створення послуг (SDC), яке підтримує специфікацію топології та оркестровки OASIS для хмарних додатків (TOSCA), для опису ресурсів і послуг (тобто активів), а також пов'язаних із ними політик і процесів. Його компонент часу виконання виконує політики, підготовлені під час проєктування, які стосуються моніторингу, збирання даних, аналітики, оркестровки послуг тощо. ONAP використовує платформу керування

замкнутою автоматизацією (CLAMP), щоб забезпечити керування життєвим циклом VNF та автоматизувати процеси розгортання E2E. Навпаки, OSM є ініціативою Європейського інституту телекомунікаційних стандартів (ETSI), яка пропонує економічно ефективно та автоматизоване надання послуг. І ONAP, і OSM відповідають еталонній архітектурі ETSI NFV. Порівняльна оцінка ONAP і OSM щодо функцій і прогалін у продуктивності наведена в [22]. Автори в [23, 24] пропонують архітектуру для керування сегментами мережі на основі ONAP, тоді як [25] вдосконалюють OSM (разом з контролером OpenStack і OpenDaylight SDN), щоб забезпечити розгортання сервісу в багатодоменні інфраструктури.

Вбудовування VN в субстратні мережі є критичним аспектом NV. Вбудовування віртуальної мережі (VNE) – це проблема розподілу ресурсів, яка охоплює вбудовування віртуальних вузлів і посилок у вузли субстрату та посилок відповідно. Для успішного вбудовування в мережу найважливішим є ефективний розподіл ресурсів. VNE є добре вивченою проблемою, яка, як було доведено, є NP-складною [26, 27].

У результаті в науковій літературі було запропоновано кілька алгоритмів лінійного програмування, алгоритмів змішаного цілочислового програмування, а також евристичних алгоритмів. Більшість запропонованих евристичних алгоритмів вирішує проблему в два етапи: (1) спочатку вбудовування вузла та (2) потім вбудовування посилок. На першому етапі вузли субстрату ранжуються на основі певної метрики (наприклад, доступності), і застосовується стратегія відображення жадібних вузлів, де відображення визначається результатами рангу. На другому етапі віртуальні канали зазвичай відображаються на найкоротший шлях, який має достатньо ресурсів пропускної спроможності між вузлами. З іншого боку, алгоритми лінійного програмування та змішаного цілочислового програмування використовуються для вирішення проблеми VNE в один етап шляхом одночасного відображення вузлів і зв'язків.

Більшість рішень VNE виконує статичні відображення та розподіл ресурсів, тобто вони не враховують перевідображення вбудованих VN шляхом міграції віртуальних вузлів і/або посилок або коригування ресурсу, виділеного для VN, у міру отримання нових запитів або навантаження на мережу, трафіку зміни шаблону. Дійсно, це суперечить інтуїції, враховуючи динамічний характер Інтернет-трафіку. Доведена неефективність статичного розподілу ресурсів спонукала до появи динамічних рішень. ML, зокрема навчання з підкріпленням (RL), було визнано особливо ефективним для вирішення проблеми динамічного розподілу ресурсів, враховуючи вищу складність проблеми порівняно зі статичним VNE. У таблиці 2.1 наведено огляд сучасного стану, який стосується VNE та розподілу ресурсів.

Таблиця 2.1 – Огляд сучасного стану вбудовування віртуальної мережі

Problem/objective	Features	ML technique
Dynamic resource allocation to achieve high resource utilization and QoS	Virtual resource substrate resource	RL with Q-learning
Dynamic resource allocation to achieve high resource utilization and QoS	Virtual resource substrate resource	RL with ANN
Node mapping to achieve high revenue-to-cost ratio	CPU bandwidth topological features	RL with ANN
Node mapping to achieve high revenue-to-cost ratio	CPU bandwidth degree	RL with RNN
VNE admission control	CPU bandwidth topological features	RNN
Substrate subgraph extraction to speed up VNE process	CPU bandwidth topological features	Hopfield network
Node mapping to achieve high acceptance ratio, high revenue-to-cost ratio, and load balancing	CPU bandwidth embedding status	Deep RL with GCN

Міджумбі та ін. [28] розглядають проблему динамічного розподілу ресурсів, використовуючи підхід на основі RL. Вони моделюють мережу як децентралізовану систему агентів Q-навчання, пов'язаних із вузлами та зв'язками. Агенти використовують Q-навчання, щоб вивчити оптимальну політику динамічного розподілу мережних ресурсів для віртуальних вузлів і посилань. Функція винагороди заохочує високе використання віртуальних ресурсів, водночас штрафуючи скидання пакетів і великі затримки. Агенти гарантують, що хоча віртуальні мережі мають необхідні їм ресурси, у будь-який момент часу для цієї мети резервуються лише необхідні ресурси. Симуляції показують, що динамічний розподіл ресурсів на основі RL значно покращує коефіцієнт прийняття VN і максимальну кількість прийнятих запитів VN у будь-який час порівняно зі статичним підходом. Цей підхід також гарантує, що такі вимоги QoS віртуальної мережі, як швидкість відкидання пакетів і затримка віртуального з'єднання, не впливають на розподіл ресурсів.

У наступній роботі [29] Mijumbi et al. вирішив використовувати штучні нейронні мережі (ANN) і запропонував адаптивний механізм розподілу ресурсів, який, на відміну від вирішення на основі Q-навчання в [28], не обмежує простір станів-дій.

Подібно до [28], вирішення проблеми розподілу ресурсів приймається децентралізованим способом агентами RL, пов'язаними з кожним вузлом

субстрату та ланкою. Кожен агент покладається на ANN, вхідними даними якого є статус вузла субстрату (відповідно зв'язку) і вбудованих віртуальних вузлів (відповідно зв'язків), і яка виводить дію розподілу.

Функція помилок, яка оцінює бажаність виходу ШНМ, використовується для цілей навчання. Мета функції помилки полягає в тому, щоб заохочувати використання віртуальних ресурсів, водночас штрафуючи скидання пакетів і великі затримки. Симуляції показують, що рішення RL на основі ШНМ перевершує рішення на основі Q-навчання, якому приписують простір станів-дій, виражений у дрібнішій деталізації.

У [30, 31] Yao et al., спираючись на інтуїцію, що мережні запити дотримуються незмінного розподілу таким чином, якщо алгоритм вбудовування добре працює для історичних запитів VN, він, імовірно, матиме таку саму продуктивність для вхідних запитів VN. У [30] вони пропонують двофазний алгоритм VNE, тобто фазу відображення вузлів на основі градієнта політики RL, за якою слідує пошук найкоротших шляхів між вибраними хост-вузлами на фазі відображення зв'язку. Агент відображення вузлів реалізований як ANN. Він навчається з історичними даними мережі та налаштовується за допомогою градієнта політики на основі показника середнього співвідношення доходів і витрат. Метою агента є спостереження за поточним станом мережі та рішеннями про відображення вихідних вузлів. Стан мережі подано матрицею, яка поєднує топологічні характеристики та використання ресурсів, отримані з кожного вузла. У [31] цю матрицю додатково зменшено за допомогою методу спектрального аналізу. Скорочена матриця поєднується зі скороченою формою матриці суміжності мережі підкладки. Збурення застосовуються до підсумкової матриці кожного разу, коли відбувається вбудовування, замість систематичних оновлень для зменшення складності. Моделювання показує, що модель, розроблена в [31], перевершує вихідну модель з [30].

Нещодавно Yao et al. [32] дослідили заміну агента відображення вузлів ANN рекурентною нейронною мережею (RNN) після формулювання VNE як проблеми часових рядів. Інтуїція підказує, що вбудовування вузлів є неперервним процесом прийняття рішень.

Агент RNN, реалізований як модель seq2seq, навчається з історичними даними мережі та точно налаштовується за допомогою алгоритму градієнта політики на основі показника довгострокового середнього співвідношення доходів і витрат. Результати моделювання показують покращення порівняно з оригінальною моделлю з [30] щодо коефіцієнта прийнятності запитів, довгострокового доходу та довгострокового співвідношення доходів і витрат.

У [33] Vlenk та ін. вивчили онлайн-проблему виконання VNE. Вони пропонують класифікатор на основі RNN, який для даного запиту VN виводить можливе вбудовування чи ні. Модель призначена для запуску перед алгоритмом VNE як такої, як процедура контролю допуску. Мета полягає в тому, щоб заощадити час і ресурси, які можуть бути витрачені

даремно, намагаючись задовольнити запит на вбудовування, який неможливо задовольнити, принаймні в прийнятний час, у поточному стані мережі.

Крім того, автори розробили нове подання мережі з відносно низькою складністю, а також запити VN, які поєднують топологічні особливості та використання ресурсів. Моделювання показує, що їхній класифікатор є високоточним і значно скорочує загальний час обчислення для онлайн-проблеми VNE, несуттєво впливаючи на продуктивність вбудовування.

У своїх постійних зусиллях пришвидшити та вдосконалити строги онлайн-алгоритми VNE, Бленк та інші [34] використовують мережі Хопфілда, щоб розробити механізм попередньої обробки VNE, який виконує скорочення простору пошуку та виділення підграфу-кандидата. Точніше, розроблена мережа Хопфілда обчислює ймовірність того, що кожен вузол буде частиною підграфу-кандидата для даного запиту на вбудовування. Потім використовується строгий алгоритм VNE, щоб знайти остаточне рішення для вбудовування в межах виділеного підграфу. Моделювання показує, що запропонований етап попередньої обробки покращує час виконання та/або продуктивність більшості протестованих онлайн-алгоритмів VNE, залежно від параметрів мережі Хопфілда, які потрібно визначити заздалегідь.

Ян та ін. [35] базуються на останніх досягненнях у глибокому навчанні та пропонують застосування глибокого RL для проблеми відображення вузлів, щоб зменшити загальний час виконання алгоритму VNE. Автори акцентують увагу на статичному розподілі ресурсів. Вони використовують графові згорткові мережі (GCN) для того, щоб навчальний агент виділяв просторові характеристики в мережі та знаходив оптимальне відображення вузлів. Агент навчання навчається з використанням підходу паралельного градієнта політики, який, як показано, збігається швидше та працює краще, ніж послідовне навчання. На додаток до винагороди за вищий коефіцієнт прийняття та співвідношення доходів і витрат, використовуваний сигнал винагороди також заохочує вивчення політики та, як показано, призводить до вищої продуктивності, ніж більш традиційні функції винагороди. Показано, що запропоноване глибоке задіяння RL перевершує найсучасніші алгоритми вбудовування без ML.

Розміщення SFC може мати такі різні цілі, як мінімізація вартості розміщення, собівартості експлуатації (наприклад, плата за ліцензію, споживання енергії); угода про рівень обслуговування (SLA) і вимоги до якості обслуговування. Ця проблема, як відомо, є NP-складною, що ускладнює або навіть забороняє її оптимальне вирішення для великих екземплярів проблеми. Крім того, евристика, як правило, неефективна в умовах великої кількості обмежень і змін у динаміці мережі [36, 37]. Нещодавно було досліджено RL для полегшення розміщення SFC у віртуалізованих середовищах.

Традиційний RL підтримує Q-таблицю для зберігання політик (тобто Q-значень), а агент RL використовує зворотний зв'язок із середовища, щоб дізнатися про найкращу послідовність дій або політику для оптимізації сукупної винагороди. Однак він не масштабується для великого простору стану-дій [38]. На відміну від цього, глибокий RL використовує нейронні мережі (NN) для вивчення Q-функції, яка відображає стани, дії на Q-значення. Глибинний RL можна класифікувати на підходи, засновані на таких цінностях, як мережа глибокого Q-навчання (DQN) і підходи на основі політики. У таблиці 2.2 наведено огляд сучасного стану, який стосується розміщення NFV.

У [39] Pei та ін. транслювали вимоги QoS як покарання, якщо не вдається обслуговувати запит ланцюжка сервісних функцій (SFCR) у розміщенні VNF. Вони використовують мережу подвійного глибокого Q-навчання (DDQN), яка містить дві NN, одну для вибору стану, дії, а іншу для оцінювання Q-значення. Після навчання DDQN його можна використовувати для розміщення VNF. Кожна дія має відповідну винагороду, яка відображає вплив дії на мережу. Після розгортання DDQN оцінює продуктивність дій і вибирає дію з найвищою винагородою відповідно до політики на основі порогового значення, щоб запустити горизонтальне масштабування. Після розміщення VNF автори використовують SFC-MAP [42] для побудови шляхів маршрутизації для впорядкування, необхідного в SFCR.

Щоб уникнути дорогого споживання пропускнуої спроможності, Zheng et al. [40] спільно оптимізують вартість надання VNF на багатоядерних серверах (тобто призначення VNF ядру ЦП). Однак у розгортанні VNF все ще є непередбачуваність, наприклад випадкове надходження SFCR, спожиті ресурси та вартість надання. Автори використовують Q-навчання, щоб полегшити потребу у визначенні апріорних переходів станів. Вони використовують ітерацію вартості, щоб вибрати однаково випадкову дію, реалізувати її та оцінити винагороду. Таким чином, їхній підхід оновлює Q-таблицю, щоб ідентифікувати переходи станів і бути стійкими до змін швидкості SFCR. Автори використовують ϵ -жадібний алгоритм, який встановлює баланс між розвідкою та експлуатацією і контролює вплив історичного досвіду. З іншого боку, Quang et al. [36] використовують глибоке Q-навчання (DQL), щоб максимізувати кількість SFC у мережі, дотримуючись при цьому обмежень інфраструктури. Вони використовують глибокий детермінований градієнт політики (DDPG), де глибокі NN (DNN), тобто актор і критик, окремо вивчають політику та Q-значення відповідно. Автори покращують DDPG, використовуючи мережу кількох критиків (MCN) для дії, де актор NN оновлюється градієнтом найкращого критика в MCN, таким чином покращуючи час конвергенції.

Таблиця 2.2 – Огляд сучасного стану розміщення на основі ML у NFV.

Problem/objective	Features	ML technique
Minimize operational cost and penalty for rejecting SFCR	CPU, memory, bandwidth	Deep RL with DDQN
Minimize cost of provisioning VNFs on multi-core servers for SFCRs	CPU	RL with Q-learning and ϵ -greedy policy
Maximize the number of SFCs based on QoS requirements	CPU, memory, storage, bandwidth	Deep RL with DDPG and MCN
Minimize infrastructure power consumption	CPU, storage, bandwidth, propagation delay	NCO with stacked LSTM and policy gradient
Minimize operational cost and maximize QoS w.r.t. total throughput of accepted SFCR	CPU, memory, bandwidth, latency	Deep RL with policy gradient
Minimize discrepancy in predicted and actual total response time	Transmission, propagation, processing times, CPU, storage	RL with Q-learning and ϵ -greedy policy

Парадигма нейронної комбінаторної оптимізації (NCO) розширена Solozabal та ін. [37], щоб оптимізувати розміщення VNF. Їх NCO використовує NN для моделювання зв'язку між екземплярами проблеми (тобто станами) і відповідними вирішеннями (тобто діями), де ваги моделі вивчаються ітеративно за допомогою RL, зокрема методом градієнта політики. Як тільки агент RL завершує роботу, враховуючи екземпляр проблеми, він повертає розв'язок. Це дозволяє зробити висновок про політику розміщення для даного SFCR, яка мінімізує загальне енергоспоживання інфраструктури (тобто функцію витрат або винагороду), враховуючи такі обмеження, як доступність віртуального ресурсу та порогові значення затримки обслуговування. Обмеження входять у функцію витрат за допомогою релаксації Лагранжа, яка вказує на ступінь незадоволення обмежень. Для NN автори використовують стекову довготривалу короткочасну пам'ять (LSTM), яка дозволяє розміщувати SFC різного розміру. Автори показують, що запропонований агент при спільному використанні покращує продуктивність жадібної евристики First-Fit.

У [38] Xiao et al. спільно вирішували такі проблеми розгортання SFC: (1) фіксація динамічного характеру запиту на обслуговування та стану мережі; (2) обробка різних характеристик трафіку запиту на мережний сервіс (наприклад, швидкість потоку) і вимог QoS (наприклад, пропускна спроможність і затримка) і (3) задоволення цілей як постачальника, так і клієнта, тобто мінімізація операційних витрат і максимізація QoS

відповідно. Для першого завдання автори використовують процес прийняття розв'язань Маркова (MDP) для моделювання динамічних змін стану мережі, де стан подано як поточне використання мережних ресурсів (тобто ЦП, пам'ять і пропускна спроможність) і вплив поточних SFC, тоді як дія відповідає розгортанню SFC, що відповідає отриманому запиту на службу. Для другого завдання автори використовують глибокий RL на основі градієнта політики для автоматичного розгортання SFC. Після конвергенції RL він надає рекомендації для розгортання SFC для кожного запиту, що надходить, дотримуючись обмежень ресурсів. Далі вони вирішують третє завдання, спільно максимізуючи зважену загальну пропускну спроможність прийнятих запитів на обслуговування (тобто дохід) і мінімізуючи зважену загальну вартість зайнятих серверів (тобто витрати) як функцію винагороди MDP (тобто дохід мінус витрати). Завдяки моделюванню, керованому трасуванням, автори демонструють свій підхід, що дозволяє перевершити жадібні та байєсовські підходи, засновані на навчанні, забезпечуючи в середньому вищу пропускну спроможність і нижчі експлуатаційні витрати.

Буньякітанон та ін. [41] визначають показники рівня обслуговування E2E (наприклад, час обробки VNF, затримку мережі тощо) для підтримки розміщення VNF. Вони враховують різномірні вузли з різними можливостями та доступністю. Автори стверджують, що їх модель на основі Q-навчання добре узагальнює різномірні вузли та умови мережі, що динамічно змінюються. Вони прогнозують показники рівня обслуговування та вживають заходів, які максимізують винагороду за правильні прогнози. Значення Q оновлюються з використанням середньозваженого нового та минулого значень Q. Винагорода містить прийнятну похибку з найвищою винагородою за прогнозування значення, яке дорівнює фактичному значенню. Вони використовують ϵ -жадібну політику, щоб знайти баланс між розвідкою та розробкою, починаючи з однакової ймовірності досліджувати чи розробляти. Потім вони генерують випадкове число та порівнюють його з ϵ -жадібним значенням, щоб скерувати до дослідження чи експлуатації. Автори показують, що їх модель має найкращу продуктивність з приблизно 94% у розвідці та 6% у експлуатації.

Масштабування ресурсів VNF передбачає початкове розгортання SFC, причому головною метою пристосування до змін є попит на обслуговування. Масштабування на основі статичного порогу є відносно простим у реалізації, коли попередньо визначені пороги використовуються для метрики продуктивності, наприклад, використання ЦП, використання пропускної спроможності тощо. Наприклад, Ceilometer у OpenStack Heat можна використовувати для створення сигналів тривоги на основі порогів використання ЦП для запуску або завершення віртуальних мережних функцій (VNFI) [43]. Однак вибрати ці порогові значення потрібно не лише нетривіально, вони також можуть потребувати частих оновлень, щоб

відповідати різноманітним вимогам до послуг. У таблиці 2.3 наведено підсумок сучасного стану, який стосується масштабування NFV.

Масштабування на основі статичного порогового значення є реактивним і не в змозі впоратися з раптовими змінами попиту на обслуговування, що призводить до втрати ресурсів і порушень SLA.

Більш того, надмірне забезпечення може призвести до низького використання ресурсів і високих операційних витрат, тоді як недостатнє забезпечення може призвести до збоїв у наданні послуг і навіть припиненні роботи. Тан та ін. [44] пропонують альтернативу статичним механізмам масштабування на основі порогових значень, які враховують SLA та ресурсоефективність. Вони моделюють масштабування VNF як MDP і використовують Q-навчання для прийняття рекомендацій щодо політики масштабування. Під час оцінювання щоденних сценаріїв завантаженості, простою та різкого трафіку їхній підхід перевершує підходи на основі статичних порогових значень і підходів, заснованих на політиці голосування (наприклад, більшість показників ефективності мають узгодити дію масштабування на основі їхніх відповідних порогових значень), дотримуючись компромісу між гарантією SLA для мережних послуг і споживанням ресурсів VNF.

Проактивне масштабування використовує попит на обслуговування та/або порогові прогнози для динамічного розподілу ресурсів між SFC. ML і є ідеальною технікою для прогнозування на основі історичних даних, а функції ML відіграють ключову роль у його продуктивності. Цао та ін. [45] використовують нові функції ML для масштабування, які містять VNF та показники рівня інфраструктури. Вони навчають NN на позначених даних, щоб охопити складні зв'язки між розподілом ресурсів, продуктивністю VNF і попитом на послуги. Однак маркування не тільки громіздке, виснажливе та схильне до помилок, але також потребує експертних знань домену NFV. Автори визначають пріоритетність розподілу ресурсів для VNF на основі терміновості та намагаються розподілити навантаження між усіма екземплярами VNF, використовуючи правила пересилання трафіку. Однак, якщо існуючі екземпляри VNF не можуть задовольнити попит на послуги, нові екземпляри мають бути створені за допомогою алгоритмів розміщення VNF. Хоча Сао та ін. [45] показують переваги складених функцій (тобто VNF та рівня інфраструктури), Schneider et al. [46] сприяють використанню ML для створення профілів продуктивності, які точно відображають складні відносини між продуктивністю VNF і вимогами до ресурсів.

З іншого боку, Shi et al. [47] вирішив використовувати MDP для масштабування компонентів віртуальної мережі (VNFC). Щоб покращити продуктивність MDP, автори використовують байєсовське навчання та використовують історичне використання ресурсів VNFC для прогнозування майбутньої надійності ресурсу. Ці передбачення використовуються в MDP для динамічного розподілу ресурсів для VNFC і сприяють роботі системи без збоїв.

Їхній підхід перевершує жадібні методи за загальною вартістю.

Таблиця 2.3 – Огляд сучасного масштабування на основі ML у NFV

Problem/objective	Features	ML technique
Trade-off between SLA and VNF resource consumption	CPU, memory, storage, bandwidth, network users and requests	RL with Q-learning
Learning resource allocation and VNF performance relationship	VNF internal statistics (e.g. request queue size) and resource utilization	NN, decision table, random forest, logistic regression, naïve bayes
Meet service demands	Performance measurements (e.g. max sustainable traffic load) and resource requirements (e.g., CPU, memory)	Support vector regression, decision tree, multi-layer perceptron, linear regression, ensemble
Predict VNFC resource reliability	QoS requirements	Bayesian learning
Predict VNFC resource reliability	CPU, memory, link delay	GNN with FNNs
Predict VNFIs, and minimize QoS violations and operational cost	Time of day, measured traffic load at different time units, and changes in traffic	Multi-layer percep., bayesian network, reduced error pruning tree, random and C4.5 decision trees, random forest, decision table
Minimize average oper. cost, SLA violation and VNF latency w.r.t. resizing, deployment, off-loading	CPU, memory, QoS	Deep RL with twin delayed DDPG and DNN

Міджумбі та ін. [43] створюють логічні зв'язки між VNFC у SFC, щоб прогнозувати майбутні потреби в ресурсах. Новизна полягає в ідентифікації зв'язків між VNFC, які можуть бути або не бути впорядкованими в межах VNF. Автори використовують граф NN (GNN), щоб змодельовати кожен VNFC у SFC як дві параметричні функції, кожен з яких змодельовано як пряму NN (FNN). Ці пари FNN відповідають за вивчення вимог до ресурсів VNFC, використовуючи історичну інформацію про використання ресурсів від VNFC та сусідніх VNFC (тобто, з використанням першого FNN), з подальшим прогнозуванням майбутніх потреб у ресурсах VNFC (тобто, з використанням другого FNN). Автори використовують зворотне поширення через час, щоб оновити вагові коефіцієнти NN і покращити ефективність прогнозування. Подібно до [45], вони також використовують функції VNF (наприклад, використання процесора, пам'ять, затримка обробки) та функції рівня інфраструктури (наприклад, посилення, ємність, затримка). Їхня модель дає найменшу середню абсолютну відсоткову похибку, коли розмір

вікна передбачення знаходиться в межах розміру вікна навчання. Інакше точність передбачення страждає, що потребує збереження моделі.

У [8] Rahman et al. використовує вимірювання трафіку та рішення про масштабування протягом періоду часу, щоб виділити функції та визначити класи для класифікаторів ML. Функції подають вимірний попит на послуги та його зміну порівняно з недавньою історією, тоді як класи подають кількість VNFI. Ці функції та класи використовуються для навчання класифікаторів ML і прогнозування майбутніх дій щодо масштабування. Автори використовують два класифікатори: перший передбачає масштабування, щоб уникнути порушень QoS, а другий передбачає масштабування, щоб зменшити експлуатаційні витрати. З огляду на неточні прогнози масштабування та/або затримки часу запуску віртуальної машини, класифікатори ML, навчені зменшувати порушення QoS, залишаються в стані погіршеного QoS протягом коротших періодів часу. Показано, що контейнеризація скорочує час запуску для VNFI і значно покращує QoS.

Roig та ін. [9] використовують немарковані дані для прийняття висновків щодо вертикального, горизонтального масштабування або перевантаження в хмару на основі запитів на обслуговування, операційної вартості, вимог до QoS і затримки, сприйнятої кінцевим користувачем. Автори використовують MDP параметризованої дії, де набір неперервних параметрів асоціюється з кожною дією. Дії відповідають призначенню користувач-сервер, тоді як параметри визначають масштабування ресурсів сервера VNF (тобто обчислення та зберігання). Це дозволяє вибирати різні сервери для користувачів, які запитують ту саму послугу VNF, щоб підвищити чутливість до затримки, яку сприймає кінцевий користувач, і забезпечити асинхронне маніпулювання ресурсами сервера. Автори використовують глибокий RL, який параметризує політику, і використовують NN акторів і критиків, щоб взяти політику за допомогою подвійної затримки DDPG.

DNN використовується для наближення політики оптимізації середньозваженої затримки, експлуатаційних витрат та QoS. Оскільки ваги можна регулювати та використовувати для оновлення політики, вона не тільки добре працює за постійного попиту на послуги, але й швидко адаптується до змін у запитах на послуги та є стійкою до змін у динаміці мережі.

Контроль доступу визначає, потрібно надавати чи відхилити новий вхідний запит на зріз на основі доступних мережних ресурсів, вимог QoS нового запиту та його наслідків для існуючих послуг, а також забезпечення доступних ресурсів для майбутніх запитів. Очевидно, що прийняття нового запиту приносить дохід провайдеру мережі. Однак це може погіршити QoS існуючих сегментів через дефіцит ресурсів, що, як наслідок, порушує SLA та загрожує штрафами й втратою доходу. Таким чином, існує невід'ємний компроміс між прийняттям нових запитів і підтриманням або дотриманням QoS. Контроль допуску вирішує цю проблему та має на меті максимізувати

кількість прийнятих запитів без порушення SLA. Кілька дослідницьких робіт, як описано нижче, розглядали проблему контролю допуску зрізів з різних точок зору за допомогою ML. У таблиці 2.4 наведено короткий виклад найсучасніших підходів до контролю доступу на основі ML у нарізці мережі.

Бега та ін. [5] презентує алгоритм контролю доступу до сегментів мережі, який максимізує монетизацію постачальника інфраструктури, одночасно забезпечуючи SLA для сегментів. Алгоритм досягає мети шляхом автономного вивчення оптимальної політики контролю доступу, навіть якщо поведінка зрізу невідома, а дані не позначені. Автори розглядають два типи зрізів: (1) нееластичні, пропускна спроможність яких завжди має бути вищою гарантованої швидкості, і (2) еластичні, пропускна спроможність яких може падати нижче гарантованої швидкості протягом деяких періодів, доки середня залишається вищою вказаної ставки. Оскільки тип зрізу, його прихід і вихід заздалегідь невідомі, неможливо встановити основну істину для задачі пропускнуго контролю. Таким чином, автори пропонують глибокий підхід RL, який взаємодіє з навколишнім середовищем і вирішує як діяти в заданому стані, отримуючи при цьому зворотний зв'язок з минулого досвіду. Їх глибока структура RL використовує дві різні NN: одну для оцінювання доходу для кожного стану під час прийняття запиту на зріз, а іншу – для відхилення запиту. Потім структура вибирає дію з найбільшим очікуваним доходом, і винагорода за дію надсилається назад до RL. За допомогою оцінювання автори показали, що запропонований ними алгоритм працює близько до оптимального в широкому діапазоні конфігурацій і перевершує наївні підходи та інтелектуальну евристику.

Таблиця 2.4 – Опис найсучасніших підходів до контролю доступу на основі ML у нарізці мережі

Problem/objective	ML technique
Maximize monetization of infrastructure provider, while ensuring slice SLAs	Deep RL framework with two different NNs
Minimize loss of revenue and loss due to penalties in service degradation	Resource prediction and RL
Maximize resource utilization while respecting slice priorities	RL with Q-learning

Раза та ін. [1] вирішує проблему контролю доступу до сегментів мережі, беручи до уваги доходи від прийнятих сегментів і штрафи, пропорційні зниженню продуктивності, якщо допущений сегмент неможливо збільшити пізніше через суперечку за ресурси. Автори пропонують алгоритм навчання під керівництвом (SL) та алгоритм на основі RL для контролю допуску до зрізів. Рішення на основі SL використовує

прогнозування вимог до вхідного фрагмента та майбутні зміни у вимогах для вхідного фрагмента та всіх інших сегментів, які наразі надані. Це полегшує ідентифікацію можливого погіршення продуктивності під час прийому для вхідного зрізу або поточних наданих зрізів, що призводить до відхилення зрізу. З іншого боку, алгоритм на основі RL вивчає зв'язок між вимогами до фрагментів і поточним розподілом ресурсів разом із загальним прибутком. Цей зв'язок керує політикою прийому фрагментів, дозволяючи приймати лише ті фрагменти, які, ймовірно, мінімально або взагалі не погіршать продуктивність. Метою політики допуску є мінімізація втрат, де втрата має два компоненти: (1) втрата доходу через відхилення запитів на зрізи та (2) втрата через штрафи за погіршення якості обслуговування, як описано в [3].

Вирішення на основі RL проблеми контролю перевантаження між сегментами в мережах 5G, яке впливає на процес контролю доступу до сегментів, запропоновано Ханом та ін. [2].

Їхня пропозиція визначає активні сегменти з невиправданими вимогами, тобто кількість виділених ними ресурсів може бути зменшена на основі доступності ресурсів, вимог до сегментів і стану черги. Ресурси ідентифікованих сегментів потім зменшуються, щоб звільнити місце для більшої кількості сегментів з вищим пріоритетом. Щоб досягти цього, автори використовують Q-навчання, яке може вивчати оптимальну стратегію перерозподілу ресурсів шляхом спільного максимального використання ресурсів і дотримання пріоритетів зрізів. Результати оцінювання показують, що запропоноване здатне збільшити відсоток прийнятих запитів на зрізи без негативного впливу на продуктивність зрізів з високим пріоритетом.

Мережа E2E може одночасно потребувати радіо, мережних, обчислювальних ресурсів і ресурсів зберігання з кількох сегментів мережі. Проблема, що постає перед провайдером мережі, полягає в тому, як одночасно керувати кількома взаємопов'язаними ресурсами. У зв'язку з динамічним попитом на послуги частота запитів на зрізи, час їх виконання та вимоги невідомі апріорі, а ресурси обмежені. Отже, динамічний розподіл ресурсів у режимі реального часу для максимізації конкретної мети є ще одним викликом для мережного провайдера.

У таблиці 2.5 наведено короткий виклад найсучасніших підходів до розподілу ресурсів на основі машинного навчання для нарізки мережі.

Wang і Zhang [4] пропонують двоетапну структуру розподілу ресурсів зрізів мережі на основі RL (тобто Q-навчання). Перший етап виконує відображення функцій віртуального стека протоколів сегмента мережі на фізичний вузол сервера. Другий етап керує асоціацією віддаленого радіоблока (RRU), підканалом і розподілом потужності. Замість того, щоб застосовувати одну модель Q-навчання для вирішення спільної проблеми, автори використовують дві ϵ -жадібні моделі Q-навчання послідовно, щоб зберегти модель масштабованою. Метою оптимізації запропонованої моделі

є максимізація корисності послуги (тобто різниці між доходом і витратами) усієї мережі, де дохід походить від тарифу обслуговування, а витрати – від вартості розгортання віртуальної функції та втрати затримки E2E.

Таблиця 2.5 – Опис найсучасніших підходів до розподілу ресурсів на основі машинного навчання для нарізки мережі

Resource type	Problem/objective	ML technique
Virtual protocol stack functions, RRU association, sub-channel and power allocation	Maximize service utility in terms of the difference between revenue and expense	RL with ϵ -greedy Q-learning
VMs and bandwidth	Minimize processing delays for received requests and resource usage costs	RL with policy gradient methods
Service capacity requirement	Maximize revenues in short- and long-term resource reallocation	ANN-based deep learning prediction
Slice bandwidth allocation and scheduling of SFC flows	Maximize the weighted sum of spectrum efficiency and QoE	RL with Deep Q-Learning
Bandwidth or time-slots	Maximize SSR and spectrum efficiency	Dueling GAN-DDQN
Computing, storage, and radio resources	Maximize the long-term average reward	RL (Q-learning, DQL, deep double Q-learning, and deep dueling)

Результати моделювання показують, що, порівняно з базовими схемами (наприклад, розгортання функції з мінімальними витратами та розподіл радіоресурсів, що максимізує співвідношення сигнал/шум), запропонований алгоритм може збільшити корисність усієї системи. Однак існує верхня межа через обмежені ресурси вузла, тоді як моделювання виконується лише на кількох десятках користувачів у системі.

Глибокий підхід RL, запропонований Коо et al. [5], вирішує проблему розподілу ресурсів зрізу мережі, враховуючи невідомі характеристики надходження зрізу, неоднорідний SLA та вимоги до ресурсів (наприклад, віртуальні машини, пропускна спроможність і пам'ять). Розподіл ресурсів сегмента стосується розподілу віртуальних машин і пропускної спроможності для кожного фрагмента з метою мінімізації затримок обробки отриманих запитів і витрат на використання ресурсів. Автори формулюють задачу розподілу ресурсів як MDP, де обмежена проблема оптимізації багатьох ресурсів формулюється для кожної послуги після надходження партії послуг. Для обох типів запитів моделі RL навчаються в автономному режимі, щоб вивчати ефективні політики розподілу ресурсів, які використовуються в розподілі ресурсів у реальному часі.

Політики є стохастичними та визначають розподіл реальних ресурсів для кожного фрагмента, який має великий і неперервний простір для дій. Автори використовують методи градієнта політики на відміну від Q-навчання, яке не може подавати стохастичні та неперервні простори дій. Симуляції з використанням як змодельованих, так і реальних трас показують, що модель перевершує базову стратегію рівномірного розподілу, яка справедливо розподіляє ресурси між кожним сегментом.

Бега та ін. [6] презентують DeepCog, інструмент аналітики даних для когнітивного управління ресурсами в сегментах мережі 5G. DeepCog прогнозує пропускну спроможність, необхідну для задоволення майбутніх потреб трафіку окремих сегментів мережі, враховуючи при цьому бажаний оператором баланс між надлишковими ресурсами (тобто розподілом ресурсів, що перевищує попит) і порушеннями SLA (тобто виділенням менше ресурсів, ніж потрібно). DeepCog використовує механізм передбачення глибокого навчання на основі ШНМ, який складається з кодера з трьома шарами тривимірних згорткових НМ і декодера, реалізованого багат шаровими перцептронами. Показано, що структура кодера-декодера прогнозує потребу в обслуговуванні з високою точністю на основі даних вимірювань, зібраних у робочій мобільній мережі. Автори стверджують, що структура є достатньо загальною, щоб її можна було навчити вирішувати задачу прогнозування потужності для різних сегментів мережі з різними моделями попиту. Прогноз потужності, наданий DeepCog, може бути використаний операторами для прийняття коротко- та довгострокових рішень щодо перерозподілу ресурсів і максимізації доходів.

У [7] Лі та ін. досліджували керування ресурсами адреси для нарізки мережі незалежно для нарізки радіоресурсів і нарізки базової мережі на основі пріоритетів. У радіочастині керування ресурсами стосується розподілу смуги пропускання для максимізації зваженої суми ефективності використання спектра та якості послуг (QoE). При нарізці основної мережі метою є планування потоків до SFC, які потребують прийняттого часу очікування. Для обох цих проблем автори використовують DQL, щоб знайти оптимальну політику розподілу ресурсів, яка підвищує ефективність і гнучкість нарізки мережі в сценарії з обмеженими ресурсами. Однак їхній підхід не враховує вплив випадкового шуму на обчислення ефективності використання спектра та QoE для нарізки радіоресурсу. Щоб подолати це обмеження, Hua et al. [8] вирішили об'єднати розподільний RL і генеративну змагальну мережу (GAN), щоб запропонувати мережу глибокого розподілу Q (GAN-DDQN) на базі GAN. Крім того, автори приймають схему обмеження винагороди та вводять дуельну структуру в GAN-DDQN (тобто Dueling GAN-DDQN), щоб відокремити розподіл цінності стану та функцію переваги дії від розподілу вартості дії. Це дозволяє обійти притаманну GAN-DDQN проблему навчання. Результати моделювання показують ефективність GAN-DDQN і Dueling GAN-DDQN порівняно з класичними алгоритмами DQL.

Van Huynh та ін. [9] пропонують модель керування ресурсами, яка дозволяє провайдеру мережі одночасно розподіляти обчислювальні ресурси, сховище та радіоресурси для різних запитів на зрізи в режимі реального часу. Щоб мати справу з динамікою, невизначеністю та неоднорідністю запитів на зрізи, автори використовують напів-MDP. Потім використовуються кілька алгоритмів RL, наприклад Q-навчання, DQL, глибоке подвійне Q-навчання та глибоке дуельне, щоб максимізувати довгострокову середню винагороду для мережного провайдера. Ключова ідея алгоритму глибокої дуелі полягає у використанні двох потоків повністю пов'язаних прихованих шарів для одночасного навчання функцій цінності та переваги, таким чином покращуючи процес навчання. Результати моделювання показують, що запропонована модель із використанням глибоких дуелей може принести до 40% вищої довгострокової середньої винагороди та в кілька тисяч разів швидша порівняно з іншими підходами до нарізки мережі. Перевага запропонованої моделі полягає в тому, що вона може враховувати додавання додаткових ресурсів або видалення деяких ресурсів (тобто масштабування), враховуючи деякі нові події в просторі стану системи. Однак у роботі [9] не розглядаються мережні ресурси, які необхідні для забезпечення зрізу E2E.

Віртуалізовані мережі та послуги створюють невід'ємні проблеми для мережних операторів, що потребує автоматизованого керування, яке не може задовольнитися традиційним реактивним підходом до керування людиною в циклі. Крім того, вимога до вищої якості обслуговування та послуг із наднизькою затримкою потребує інтелектуального керування, яке має використовувати величезний обсяг даних у мережі та приймати автоматизовані управлінські рішення. Таким чином, AI і ML можуть відігравати ключову роль у реалізації автоматизації керування віртуалізованими мережами та послугами.

Моніторинг потребує ідентифікації таких ключових показників ефективності (KPI), як очікувана затримка, тривоги та використання віртуалізованих мережних компонентів [6]. Вони відіграють вирішальну роль в аналітиці для полегшення автоматизованого прийняття рішень щодо керування віртуалізованими мережами та службами. Квінтесенцією є те, що використовувані методи вимірювання збирають телеметричні дані з високою точністю, мінімізуючи накладні витрати. Однак вимірювання може збільшити витрати (наприклад, споживана пропускна спроможність мережі, пам'ять комутатора через зондування та зберігання), коли велика кількість віртуалізованих мережних компонентів відстежується через регулярні проміжки часу. Це викликає потребу в адаптивних схемах вимірювання, які можуть динамічно налаштовувати швидкість моніторингу та вирішувати, що контролювати.

Такі методи ML, як регресія, можуть полегшити адаптивний моніторинг шляхом прогнозування даних телеметрії, які в іншому випадку були б виміряні. Іншим завданням є розробка механізмів своєчасного та

високоточного інструментарію моніторингу KPI віртуалізованих мереж із високими вимогами до якості обслуговування, особливо для служб із наднизькою затримкою.

Прогнозне обслуговування на основі ML може забезпечити безперебійну роботу віртуалізованих мереж [19]. Це передбачає визначення майбутніх подій на основі вимірних KPI, виявлення причин погіршення продуктивності та проактивне вжиття превентивних заходів. Прикладом є визначення того, чи призведе погіршення продуктивності (наприклад, збільшення втрати пакетів, тривалий час простою) до майбутніх порушень QoS.

Також важливо визначити причини (наприклад, неправильна конфігурація, збій) зниження продуктивності у зв'язку з потенційними сигналами тривоги. Однак усвідомити це на основі величезного обсягу телеметричних даних і стохастичного характеру мережних подій є складним завданням. Для вирішення цих проблем можна досліджувати підходи на основі даних, зокрема ML. Після визначення причини погіршення продуктивності потрібно виконати робочі процеси для зменшення впливу на KPI. Виведення цих робочих процесів і оптимальне планування їх виконання з мінімальним перериванням існуючого трафіку є нетривіальним. Однак RL, здається, добре підходить для вирішення проблеми, і його потрібно дослідити, щоб знайти оптимальні робочі процеси пом'якшення.

У мобільних мережах 5G зріз E2E VN охоплює кілька сегментів мережі, кожен з яких може мати різні технологічні та фізичні обмеження. Наприклад, мережа доступу може мати обмежену пропускну спроможність і масштабованість для мінімізації витрат і енергії, тоді як базова мережа може не мати цих проблем щодо пропускну спроможності або масштабованості.

Однак базова мережа може мати вищу затримку та енергоспоживання через великі географічні відстані та складніші мережні пристрої. Подібні компроміси існують між периферійними та центральними DC щодо потужності обробки, затримки та споживання енергії. Таким чином, буде непрактично створювати сегмент мережі для пікового попиту на трафік. Отже, потрібно досліджувати алгоритми динамічного забезпечення сегментів, де рішення про оркестрування ресурсів полегшуються моделями ML для прогнозування обсягу трафіку сегментів з часовими, просторовими міркуваннями та вимогами до якості обслуговування. Таке динамічне надання зрізу буде ввімкнено за допомогою NFV, що дозволяє створювати віртуалізовані NF на вимогу, і контролери SDN, які можуть направляти трафік до щойно створених NF.

Навіть за прогнозованого технічного обслуговування деякі збої, такі як порізи оптоволокна та пошкодження пристрою, неминучі. Здатність мережного провайдера швидко усунути збій має вирішальне значення для підтримки роботи мережі. Керування відмовами має три етапи: виявлення

відмов, локалізацію та ідентифікацію. Мета виявлення збою полягає в тому, щоб викликати попередження після того, як стався збій. Після виявлення несправний елемент (наприклад, вузол або зв'язок, відповідальний за збій) потрібно локалізувати в мережі, щоб звузити коло причини збою. Навіть після локалізації може бути складно зрозуміти точну причину збою. Наприклад, усередині вузла мережі погіршення може бути спричинене неправильною конфігурацією або несправністю. Щоб прискорити процес усунення несправностей, всі три етапи усунення несправностей мають бути автоматизовані. Цікавим напрямком дослідження є розробка моделей ML і алгоритмів для автоматичного виявлення, локалізації та ідентифікації збоїв на основі даних, згенерованих у виробничих мережах. Ці моделі зменшать середній час відновлення після подій збою, таким чином підвищуючи доступність сегмента мережі або віртуалізованої мережі/послуги.

Вимоги до трафіку та/або вимоги до QoS віртуалізованої мережі або її фрагмента можуть змінюватися з часом через зміну кількості користувачів і моделей зв'язку [6]. Отже, початковий розподіл ресурсів потрібно адаптувати до таких змін, спричиняючи при цьому мінімальні завади для існуючого трафіку. Це потребує, щоб моделі ML передбачали зміни у вимогах своєчасно й точно, а також сприяли динамічній адаптації розподілу ресурсів.

Крім того, з часом прихід і вихід віртуалізованих мереж або мережних фрагментів може призвести до фрагментації та спотвореного використання посилань і серверів обробки. Це, у свою чергу, може вплинути на прийняття майбутніх запитів і призвести до непотрібного споживання енергії. Один із способів пом'якшити це – повторна оптимізація розподілу смуги пропускання та періодична консолідація віртуальних машин або контейнерів. Все зроблене також має виводити послідовність операцій (наприклад, міграція віртуальної машини, міграція віртуального каналу та перерозподіл пропускнуої спроможності), які приводять до стану збалансованого навантаження. RL є ідеальною технікою для створення послідовності операцій, для досягнення оптимізованого стану.

У розгортанні NFV або в сегментах мережі VNFI, які знаходяться на віртуальних машинах або контейнерах, масштабуються відповідно до вимог служби. Однак продуктивність VNF чутлива до основного апаратного забезпечення [45, 46]. Наприклад, можливості обробки трафіку віртуальних ЦП на процесорі Intel Xeon відрізняються від процесора AMD Opteron [45]. Подібним чином час завантаження для віртуальних машин відрізняється для таких різних VIM, як OpenStack, Eucalyptus і OpenNebula [43]. Тим не менш, більшість досліджень передбачає однорідне обладнання, не звертаючи уваги на його вплив на продуктивність VNF. Це надмірне спрощення, яке може призвести до неякісних моделей ML і неточних рішень щодо масштабування на практиці. Таким чином, дуже важливо розробити профілі продуктивності [46], які охоплюють чутливість продуктивності VNF на різному обладнанні. У разі горизонтального масштабування ці профілі

можна використовувати для точного вимірювання впливу на продуктивність нових VNF на різних фізичних серверах. Дійсно, задіяння цих профілів збільшить розмірність проблеми масштабування. Наївним варіантом є врахування апаратної чутливості як вартості. Однак створення профілів продуктивності VNF для різного обладнання є громіздким завданням. Залишається оцінити, як ці апаратні профілі продуктивності вплинуть на точність моделей ML і рішень щодо масштабування VNF.

Очевидно, відбувся сплеск застосування ML для керування віртуалізованими мережами, починаючи від розміщення та масштабування VNF до контролю доступу в сегментах мережі. Однак численні дослідження припускають, що сам ML є непереможним. Це нереалістичне припущення, оскільки зловмисники можуть спотворити навчальні дані або скомпрометувати агента RL, маніпулюючи станами та політиками системи, що призведе до неповноцінних дій [2]. Наприклад, перешкодження фактичному споживанню ресурсів мережею підкладки може призвести до неоптимального розміщення SFC, що призведе до втрати ресурсів та/або порушень SLA. За своєю суттю моделям ML не вистачає стійкості до спроб суперництва. Змагальне навчання вирішує це занепокоєння, використовуючи ретельно розроблені змагальні (тобто тестові) зразки з незначними порушеннями регулярних вхідних даних [3, 4]. Вони можуть бути використані для впровадження надійності в моделі ML проти атак із спотворенням даних. GAN широко використовувалися для створення таких змагальних вибірок. GAN – це клас методів глибокого навчання, які використовують дві нейронні мережі, дискримінатор і генератор, щоб конкурувати одна з одною за навчання моделі. Однак GAN можуть страждати від нестабільності навчання через подроблені навчальні дані, які погіршують продуктивність моделі [5]. Тому забезпечення конвергенції GAN є відкритою дослідницькою проблемою. Крім того, використання GAN для захисту агентів RL від складних векторів загрози є досить невивченим. Змагальний глибокий RL з кількома агентами [6, 7], навчений у розподілених віртуалізованих середовищах, також може допомогти зменшити вплив змагальних спроб.

2.2 Самокеровані мережі 5G

Окрім покращення продуктивності (тобто меншої затримки, вищої пропускної спроможності, підвищеної надійності, серед іншого) та нових досягнень у радіотехнологіях (тобто нового спектра завдяки впровадженню суб-6 ГГц і мм хвиль) 5-е покоління (5G) мобільного зв'язку спрямоване на розширення дизайну підключення загального призначення попередніх поколінь для підтримки широкого спектра варіантів використання з різними вимогами та можливостями. Традиційний універсальний підхід до мережної інфраструктури більше не підходить для підтримки такого бачення; потрібен перехід до індивідуального підходу, який підтримує різні

віртуальні та ізольовані мережі в межах однієї спільної мережної інфраструктури.

Таким чином, дизайн мереж 5G рухається до високомодульної, високопрограмованої архітектури, інтегруючи інноваційні та революційні механізми з інших технологічних сфер, наприклад, архітектури на основі послуг, орієнтовані на інформаційні технології (IT) хмарні сервіси та штучний інтелект/машинне навчання. (AI/ML).

Це дозволяє створювати нові послуги та сценарії використання, які не були можливими з попередніми поколіннями мобільного зв'язку. У [1] визначено широкий спектр випадків використання, які потребують підтримки мереж 5G: відеорозваги, засоби масової інформації, автомобільна промисловість, індустрія 4.0 та послуги, пов'язані з електронною охороною здоров'я. Кожен із цих варіантів використання має різні вимоги не лише до затримки, пропускну здатності, надійності та мобільності, а й до пріоритету служби. Наприклад, трафік послуги віддаленої хірургії має бути пріоритетизованим із постійним дотриманням вимог зв'язку, на що не впливають будь-які послуги з нижчим пріоритетом, такі як послуга телевізійної трансляції 4K.

Ця різноманітність послуг, які підтримуються мережами 5G, змінює спосіб керування та експлуатації мобільної мережі. Потрібно забезпечити можливість співіснування та надання різних послуг через одну мережну інфраструктуру, одночасно задовольняючи вимоги кожної з цих послуг. В результаті, потрібно розставляти пріоритети трафіку послуг, щоб відповідати ключовим показникам ефективності послуг (KPI), ефективно розподіляти бездротові ресурси, довжини хвиль оптоволокна та обчислювальні ресурси; це завдання управління, які мають бути автоматизовані в системах 5G.

Наприклад, у разі стихійного лиха може знадобитися послуга електронної охорони здоров'я для віддаленої операції в постраждалому районі. За таких обставин мережа 5G має автоматично визначати пріоритети та розподіляти ресурси для послуг eHealth, одночасно підтримуючи існуючі послуги. Потрібне безкорисне керування мережею та послугами (ZSM), оскільки додавання людини в цикл збільшує час відповіді служби, що безпосередньо впливає на життя людей. Крім того, використання статичного або попередньо визначеного набору правил і політик може бути недостатнім для неочікуваних подій, що призведе до небажаних і/або неоптимальних рішень. Таким чином, потрібні нові та автоматизовані механізми керування мережею та послугами. 5G дає змогу підтримувати різні віртуальні та ізольовані мережі (надалі – сегменти мережі) у межах однієї спільної мережної інфраструктури. Важливою властивістю є те, що кожен зріз мережі ізольований один від одного, що означає, що продуктивність кожного зрізу не має впливати на продуктивність співіснуючих зрізів. Дотримуючись попереднього прикладу, сегмент мережі може бути призначений для послуг електронної охорони здоров'я,

маючи найвищий пріоритет серед усіх інших сегментів і навіть більші слоти в інтервалах радіопередачі; в той час як послуги відеорозваг працюватимуть поверх фрагмента з менш пріоритетною смугою пропускання для забезпечення потокового відео 4К.

У [2] організація проєкту партнерства третього покоління (3GPP) надає детальну специфікацію того, як внести нарізку мережі до стандартної специфікації архітектури 5G. У специфікації зазначено, що сегменти мережі можуть відрізнятися, «оскільки вони надають іншу закріплену послугу», як показано на рисунку 2.4. На рисунку 2.4 показано, як масштабуються інфраструктурні ресурси, призначені для сегмента електронної охорони здоров'я, після виникнення надзвичайної ситуації. Крім того, на рисунку 2.4 зображений інтелектуальний агент, який ухвалює дії щодо масштабування на основі перевірки отриманого трафіку.

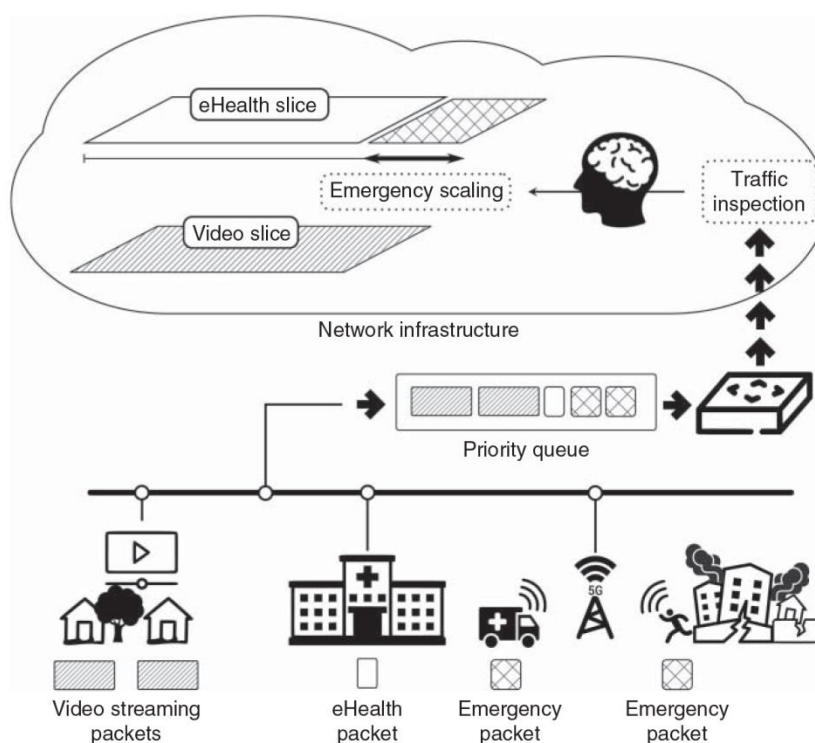


Рисунок 2.4 – Нарізка та пріоритезація мережного трафіку 5G. Зріз мережі eHealth збільшується після виникнення надзвичайної ситуації

Інтелектуальний агент – це інструмент керування на основі програмного забезпечення, який може базуватися на штучному інтелекті/ML, як розглядатиметься далі в цьому підрозділі, і він покращує мережу за допомогою самокерування, що, у зображеному сценарії, збільшує ресурси сегмента eHealth для розміщення вхідного трафіку надзвичайної події.

Віртуалізація мережних функцій (NFV) стала ключовим засобом для реалізації нарізки мережі, дозволяючи мережним функціям і службам віртуалізувати на стандартному обладнанні замість того, щоб працювати на

спеціальному обладнанні. Таким чином, NFV додає підвищену гнучкість для запуску віртуальних мережних функцій (VNF) на серверах у різних місцях. Незалежно від використовуваної технології віртуалізації [3], NFV пропонує можливість створення нових мережних служб (NS), виведення з експлуатації NS, коли вони більше не потрібні, або масштабування та міграції NS, щоб впоратися зі змінами вимог. Крім того, сегмент мережі може бути розгорнутий через визначений NFV NS і складатися з одного або кількох таких NS.

Як наводилося в даному розділі, агент може автоматизувати мережне забезпечення вищезгаданих сегментів.

Виконання угод про рівень обслуговування (SLA) є ключовим компонентом контрактних угод між вертикалями та постачальниками послуг. Враховуючи, що попит на мобільний трафік до 2023 року зросте до 43,9 Мбіт/с, ризик вузьких місць і порушення SLA є високим, якщо не буде змінено парадигму керування мережею. Безпосереднім наслідком зростання попиту на трафік є дефіцит обчислювальних і мережних ресурсів, що належать постачальникам послуг. Якщо в інфраструктурі постачальника послуг закінчуються ресурси, він може покладатися на парадигму федерації, підхід, який передбачає розподіл послуг або ресурсів між власниками інфраструктури або постачальниками послуг. Наприклад, два власники інфраструктури нададуть інтерфейси для оренди своїх ресурсів один одному, тому обидва вони зможуть негайно орендувати дисковий простір, радіопокриття в певній сфері або службу кешування місцезнаходження. Оренда відбувається під егідою узгоджених угод SLA з використанням інтерфейсів пірингової інфраструктури/постачальника послуг. Об'єднання в мережі 5G не тільки розширить постачальників послуг і власників інфраструктури за допомогою більш широкого пулу ресурсів і послуг, але також дозволить постачальникам послуг задовольнити вимоги до послуг, делегуючи розгортання одноранговим доменам.

Нарізка мережі, віртуалізація та об'єднання – три ключові концепції, які дозволять здійснювати самокерування в мережах 5G. Щоб отримати максимальну віддачу від трьох із них, потрібно зрозуміти, який стан поточної мережної технології. Таким чином, підрозділ 2.2 наводить огляд технології мережі радіодоступу (RAN), NFV, програмування площини даних і оптичних комутаторів, а також керування мережними даними. Цей підрозділ охоплює всі ці аспекти та збирає дані про поточний стан техніки (SoA); про те, як використовувати методи AI/ML для оцінювання керування ресурсами RAN, оркестровки NS, нарізки в площині даних, розподілу довжин хвиль волоконної оптики.

У цьому підрозділі наведено короткий огляд базових технологій, які є фундаментальними для наскрізного створення ресурсів і керування ними. За допомогою таких технологій мережі 5G можуть автоматизувати обробку ресурсів і ізоляцію між логічними та фізичними ресурсами. У цьому сенсі розглянуто останні підходи в домені RAN з відкритою мережею

радіодоступу (O-RAN) як яскравий приклад. Також зосереджується увага на відповідних контролерах, заснованих на стеку Європейського інституту телекомунікаційних стандартів керування та оркестровки (ETSI MANO), і, зокрема, на тому, як обробляються обчислювальні ресурси (наприклад, через Openstack). Після чого розглядаються фундаментальні ідеї та останні тенденції програмування площини даних на рівні пакетів і на рівні оптичного перемикачів, зосереджуючись на транспорті керування мережними ресурсами. Також описано створення конвеєра даних для забезпечення стека керування мережним моніторингом і аналітикою.

Віртуалізація радіодоступу (також відома як мережа віртуального радіодоступу (vRAN)), яка обіцяє значну економію операційних/капітальних витрат (OPEX/CAPEX), високу гнучкість і відкритість для сприяння інноваціям і конкуренції, є останньою віхою в революції NFV і буде ключовою технологією для систем поза 5G (B5G). Проте використання сильних сторін NFV на арені RAN тягне за собою багато проблем, які є об'єктами дослідження аких багатьох ініціатив, як розгортання Rakuten із нуля в Японії, Cisco Open vRAN Ecosystem, телекомунікаційний інфрапроєкт (TIP) TIP vRAN Fronthaul Project Group та O-RAN Alliance. Можливо, O-RAN є найбільш перспективним серед цих проєктів.

На рисунку 2.5 зображено високорівневий вигляд архітектури O-RAN. Безсумнівно, найважливішими функціональними компонентами, запровадженими O-RAN, є інтелектуальний радіоконтролер (RIC) не в реальному часі (Non-RT) і RIC, що працює майже в режимі реального часу.

У той час як перший розміщений у структурі керування послугами та оркестровки (SMO) системи (наприклад, інтегрований у платформу автоматизації відкритої мережі [ONAP]), другий може бути розміщений разом із функціями 3GPP gNB, а саме: O-RAN-сумісний хмарний блок (O-CU) та/або O-RAN-сумісний розподілений блок (O-DU), або повністю відокремлений від них, якщо дотримуються обмеження затримки. Крім того, O-CU може бути роз'єднаний у двох VNF для керування (O-CU-CP) і площиною даних (O-CU-DP). На рисунку 2.5 також зображено O-Cloud, хмарну платформу, сумісну з O-RAN, яка, за потреби, використовує надбудови апаратного прискорювача (наприклад, для прискорення швидкого перетворення Фур'є (ШПФ) або робочих процесів декодування), а також програмний стек, відокремлений від апаратного забезпечення для розгортання вдосконаленого NodeB/NodeB наступного покоління (eNB/gNB) як VNF у сценаріях vRAN (Open RAN NodeB (O-gNB)).

Керування послугами та оркестровка (SMO): SMO об'єднують кілька служб оркестровки та керування, які можуть виходити за межі чистого керування RAN, наприклад керування ядром наступного покоління (NG-) 3GPP або наскрізне керування сегментами мережі. У контексті O-RAN основними обов'язками SMO є: (1) інтерфейс несправностей, конфігурації, обліку, продуктивності та безпеки (FCAPS) для функцій мережі O-RAN; (2)

оптимізація RAN у великому часовому масштабі і (3) керування та оркестрування O-Cloud через інтерфейс O2, охоплюючи виявлення ресурсів, масштабування, FCAPS, керування програмним забезпеченням, створення, читання, оновлення та видалення (CRUD) ресурсів O-Cloud.

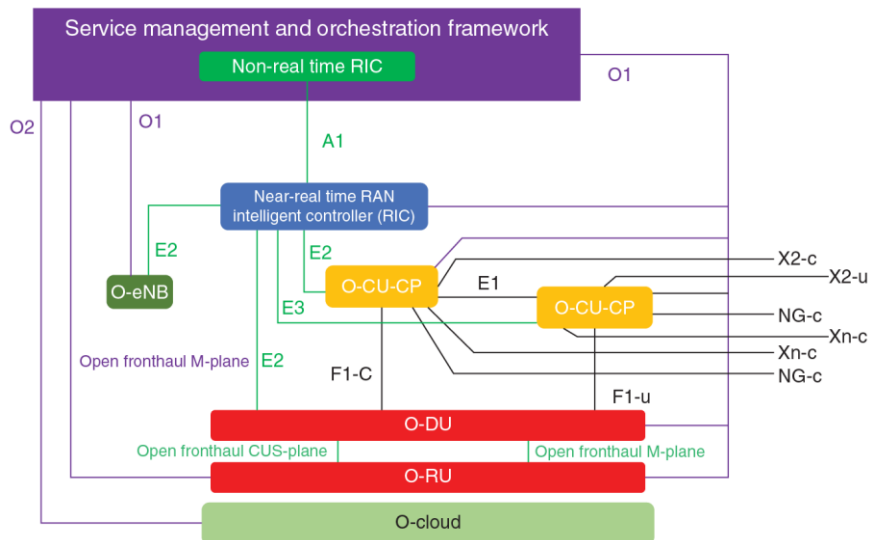


Рисунок 2.5 – Архітектура високого рівня O-RAN

Інтелектуальний контролер Non-RT RAN (Non-RT RIC), як згадувалося раніше, ця логічна функція знаходиться в SMO та забезпечує інтерфейс A1 для RIC, що ближче до RT. Його основна мета полягає в підтримці оптимізації RAN у великому часовому масштабі (секунди або хвилини), також обчислення політики, керування моделлю ML (наприклад, навчання) та інші функції керування радіоресурсами (RRM) у межах цього масштабу часу.

Завдання керування даними, запитувані не-RT RIC, мають бути перетворені в інтерфейс O1/O2; і контекстна/збагачена інформація може бути надана RIC у ближньому RT через інтерфейс A1.

Інтелектуальний контролер Near-RT RAN (Near-RT RIC): Near-RT RIC – це логічна функція, яка забезпечує оптимізацію та контроль у реальному часі, а також моніторинг даних вузлів O-CU та O-DU у часових масштабах, наближених до реального часу. (між 10 мс і 1 секундою). З цією метою керування RIC у режимі ближнього RT керується політиками та підтримується моделями, обчисленими/навченими Non-RT RIC. Однією з основних операцій, призначених для RIC near-RT, є RRM, але RIC near-RT також підтримує програми сторонніх розробників (так звані xApps). Оркестровка vRAN є складною дослідницькою проблемою, оскільки продуктивність vBS залежить від таких факторів, як умови каналу та попит користувачів.

Крім того, на бітрейт впливають SNR, конфігурація RAP і об'єднання обчислювальних ресурсів [6]. Крім того, рішення щодо розмірів процесора

та оркестровки радіозв'язку (такі як модуляція та вибір схеми кодування) впливають не лише на продуктивність, але й на вартість обчислювальних навантажень. Для цього ми подаємо рішення оркестровки vRAN за допомогою ML для спільного визначення політики розподілу радіозв'язку та ЦП у різних контекстах і мережних умовах.

Концепція NFV, презентована ETSI, дозволяє повністю розгорнути конкретні мережні функції в програмному забезпеченні, що використовує методи віртуалізації [7] в середовищі хмарних обчислень. Завдяки цьому телекомунікаційні мережі досягають вищої гнучкості мереж, а також інших привабливих переваг, таких як зниження операційних і капітальних витрат порівняно з традиційними виділеними апаратними вирішеннями, що підтримують такі функції мережі. У цьому контексті наскрізна послуга (називається NS) формується послідовністю VNF, розміщених у хмарі (наприклад, центри обробки даних (ЦОД)) або периферійні об'єкти, з'єднані між собою через різні сегменти/домени, що забезпечують потрібну функціональність NS.

У [8] ETSI запропонував архітектурну структуру для програмного забезпечення мережних функцій, вказуючи ключові елементи контролю та керування, функції та інтерфейси, що підтримують розгортання будь-якої NS, що складається з різних VNF. Зокрема, ця архітектура охоплює такі функції:

- підтримка роботи VNF на різних гіпервізорах і обчислювальних ресурсах для надання доступу до спільного сховища, обчислень і фізичної/віртуальної мережі;

- побудова графів пересилання VNF;
- NS з різними вимогами задіяних методів віртуалізації;
- використання технології постійного струму.

У цій структурі архітектури NFV основним елементом є MANO, який керує оркестровкою та життєвим циклом NS/VNF. Іншими словами, він бере на себе всі завдання керування віртуалізацією. MANO поділяється на набір функціональних модулів, які взаємодіють між собою через чітко визначені інтерфейси.

- Менеджер віртуальної інфраструктури (VIM): цей елемент забезпечує контроль і керування віртуалізованими ресурсами, доступними в інфраструктурі віртуалізації мережних функцій (NFVI), як-от зберігання, мережа та обчислення.

Це означає розподіл, модифікацію, звільнення ресурсів NFVI (тобто об'єднання віртуальних і фізичних ресурсів) тощо.

- Диспетчер віртуальних мережних функцій (VNFM), ця функція керує життєвим циклом розгорнутих VNF для NS.

- Оркестровник NFV (NFVO): він координує (тобто керує) вибором і розподілом ресурсів (як обчислювальних, так і мережних) між кількома VIM. Крім того, він відповідає за керування життєвим циклом розгорнутих NS.

Серед різних вирішень VIM, доступних у світі відкритого коду, OpenStack [9] є одним із найпопулярніших програм для керування NFVI та розгортання VNF у формі віртуальних машин (VM). Він пропонує зрілу платформу хмарних обчислень для оркестрування ресурсів, використовуючи переваги підтримки великої спільноти розробників і представників галузі. Інші VIM, орієнтовані на контейнери, останнім часом привертають все більше уваги (наприклад, Kubernetes). З іншого боку, для доступних розв'язань MANO існує набір підходів, керованих спільнотою, які набувають помітної актуальності в останні роки. У цьому контексті існує проєкт Open Source MANO (OSM) [10], ONAP [11], фреймворк Open Baton [12] і відкрита платформа для віртуалізації мережних функцій (OPNFV) [13].

Усі вони подають власну стадію розробки, але загалом вони займаються, головним чином, функціями NFVO та VNFM, а також підтримують, як було сказано раніше, найпопулярнішу реалізацію VIM, тобто OpenStack.

Спочатку задумана для сценаріїв внутрішнього центру обробки даних, парадигма SDN переживає швидкий розвиток завдяки надзвичайній гнучкості використання та наявності стандартних протоколів керування, які зробили їх реалізованими на постійно зростаючій кількості комутаційних пристроїв і в різних сценаріях, починаючи від 5G до оптичного транспорту. Однак правила SDN не мають статусу та застосовуються до чітко визначеного потоку трафіку. Це означає, що більш інтелектуальна мережна поведінка, заснована на контексті або інформації про стан, делегується центральному контролеру з урахуванням значних обмежень масштабованості.

Крім того, наразі виробники комутаторів SDN реалізували закриті та власні рішення конвеєра комутаторів (тобто функціональну структуру, реалізовану в ASIC), вибираючи, через архітектурні, деякі функції безпосередньо в апаратному забезпеченні, а інші – в програмному забезпеченні, таким чином впливаючи на перемикання продуктивності жорстким способом.

Нова фаза SDN відкриває шлях до визначення та програмування конвеєра комутатора для реалізації робочих процесів із збереженням стану без необхідності запитувати контролер.

Мова пакетних процесорів, незалежних від протоколу програмування (P4), була розроблена консорціумом P4.org [14] як мова високого рівня, незалежна від платформи, яка використовується для програмування керування, дій, кодування та декодування заголовків протоколів і таблиць, в програмованому комутаторі SDN, що працює на комутаторах загального призначення з голим металом, програмованою вентиляльною матрицею (FPGA) і інтелектуальними мережними адаптерами. Зокрема, P4 дозволяє визначати та керувати метаданими, тобто додатковими даними, які можна асоціювати з пакетом, разом із об'єктами зі збереженням стану (наприклад,

регiстрами, та лічильниками). Такі функції дозволяють реалізувати автомати стану всередині самого конвеєра, здатні ідентифікувати та керувати складними подіями, а також приймати динамічні рішення для кожного пакета. Крім того, комутатор P4 дозволяє визначати та реалізовувати власні заголовки.

У комутаторі P4 вхідні пакети спочатку передаються аналізаторам, визначеним P4 (стандартним або пропрієтарним). Потім застосовуються дві програмовані секції конвеєрів, перед вибором вихідного порту (тобто, вхідного конвеєра) і після вибору вихідного порту (тобто, вихідного конвеєра). Кожен конвеєр програмується різними таблицями потоку та діями. Крім того, секції керування визначають порядок і умовне виконання вибраних таблиць потоків, що дозволяє реалізувати прості кінцеві автомати.

Приклади застосування варіюються від розширених багаторівневих політик інженерного трафіку (наприклад, динамічне перемикання та агрегація трафіку на основі статистичних функцій трафіку) [15] до реалізації спеціальних мережних протоколів, прозорих для кінцевого користувача, активних внутрішньосмугових програм телеметрії [16], роздільної оптичної мережі на основі мережних адаптерів [17], багатокористувацького доступу [18] до контекстно-залежних функцій кібербезпеки [19], виділення функцій для двигунів AI [20] і потенційно прямої обробки прикладного рівня. У контексті 5G деякі функції, делеговані певним системним VNF, були реалізовані безпосередньо в інфраструктурних комутаторах «голого металу», наприклад, завершення GRPS-тунелювання периферійних обчислень з множинним доступом (MEC) і широкосмугові мережні шлюзи (BNG) [21] з чіткими переваги щодо продуктивності та затримки.

Мультиплексор з реконфігурацією Add Drop (ROADM, тобто оптичний комутатор) – це пристрій, який дозволяє комутувати оптичні сигнали без будь-якого електричного перетворення (тобто повністю оптичний). Він подає порти додавання/виведення, підключені до регульованих лінійних інтерфейсів оптичного мультиплексування з щільним розподілом хвиль (DWDM) транспондера, а також низку (відповідно до вузлового ступеня) лінійних інтерфейсів, підключених до інших ROADM. Загалом, керування ROADM здійснюється через власний GUI/LCT. Парадигма SDN була для оптичних мереж основним фактором розвитку нових функцій. Багато постачальників відкривають «чорну скриньку» своїх «рішень» для передачі, відкриваючи новий інтерфейс прикладного програмування (API) для контролера SDN. Різні робочі групи (наприклад, OpenConfig [22], OpenROADM [23] і TIP [24]), очолювані основними мережними операторами, докладають зусиль для визначення наборів даних нового покоління (YANG), незалежно від постачальника; моделей, що дозволяють відображати та керувати оптичними пристроями. Протокол конфігурації мережі (NETCONF) використовується для зв'язку між реконфігурованим оптичним додатковим мультиплексором (ROADM) і екземпляром(ами)

оптичного контролера SDN, спираючись на моделі YANG і дозволяючи як керувати пристроєм, так і контролювати основні параметри передачі.

Розглядаючи моделі OpenROADM YANG, пристрій ROADM подано в ієрархічній структурі, що складається з шести основних розділів: (1) загальна інформація, (2) список полиць, (3) список друкованих плат, (4) список інтерфейсів і списки як (5) внутрішніх, так і (6) зовнішніх посилань. Кожна «полиця» містить список слотів, куди встановлюються різні компоненти. Кожна «друкована плата» складається зі списку фізичних портів. Віртуальні інтерфейси на різних рівнях оптичної транспортної мережі (OTN) (тобто оптична транспортна секція (OTS), оптична мультиплексна секція (OMS)) зазначаються в «списку інтерфейсів». Кожен віртуальний інтерфейс визначається поверх фізичного інтерфейсу або поверх іншого віртуального інтерфейсу (за стеком OTN). «Список внутрішніх посилань» містить усі з'єднання між встановленими компонентами, тоді як усі зовнішні з'єднання з іншими оптичними комутаторами внесено до «списку зовнішніх посилань», повідомляючи інформацію про топологію мережі.

Нині моделі YANG були розширені функціями телеметрії, що дозволяє здійснювати моніторинг оптичних комутаторів у реальному часі. Ця функція полягає в потоковій передачі значень ключових параметрів передачі до певного збирача даних (тобто платформи моніторингу), що дозволяє виявляти можливі збої (тобто як м'які, так і жорсткі збої) та аномалії.

В архітектурі мережного керування даними високого рівня існує п'ять основних компонентів/модулів, які взаємопов'язані один з одним у різних формах: (1) з'єднання даних, (2) прийом даних, (3) аналіз даних, (4) зберігання даних і (5) візуалізація даних. Підключення даних використовується як тригер для підключення до джерела даних, яке може бути в Інтернеті, на мобільному пристрої/пристрої Інтернету речей або в сховищі даних. Джерела даних можуть бути в різних формах, наприклад джерело статичних даних із файлів, баз даних (MySQL, MongoDB тощо) або джерело поточних даних із сторонніх API чи фреймворків. Цей етап можна налаштувати для передачі даних у модуль прийому даних.

Модуль прийому даних діє як проміжний етап для вхідних даних від модуля підключення. Передача даних зазвичай використовується для передачі даних між зовнішніми системами та кластером великих даних (наприклад, на основі Hadoop). Поточні дані можуть надходити в кластер у реальному або майже реальному часі. Коли дані завантажуються, їх можна використовувати для подальшої обробки (наприклад, за допомогою Apache Spark) або для зберігання (наприклад, за допомогою розподіленої файлової системи hadoop (HDFS) [25]). Під час прийому даних також можна виконувати збагачення даних, агрегацію, перетворення тощо. Деякі приклади в цій категорії: Apache Kafka [26], Apache Spark Streaming [27] і Apache Flink (з використанням DataStream API) [28].

Після прийому даних модуль обробки/аналітики даних забезпечує виконання таких завдань, як перетворення даних, очищення, розміщення, перевірка цілісності та об'єднання як потокових, так і пакетних даних. Існує багато фреймворків, які можуть працювати як у пакетному, так і в потоковому режимах аналітики (Spark, Flink, Drill [29] тощо). Наприклад, Apache Spark можна використовувати для виконання стандартних процесів вилучення, перетворення та завантаження (ETL) великих даних. На етапі зберігання даних різні бази даних SQL/NoSQL, такі як PostgreSQL, MongoDB, Cassandra, CouchDB, Hbase і CosmosDB, можна використовувати для короткострокового або довгострокового зберігання великого обсягу. Етап візуалізації даних звітує про продуктивність, щоб пояснити та подати результати аналізу даних або самі дані у вичерпній формі.

Він також може слугувати інтерфейсом для користувачів, щоб виконувати або складати аналітику в рамках обробки даних і аналітики та візуалізувати результати. Деякі приклади інструментів містять Elasticsearch Kibana [30], Tableau [31], Grafana [32] і Apache Superset [33]. Простий приклад того, як такий конвеєр може бути інтегрований у структуру керування мережею 5G і оркеструванням, можна знайти в [34].

Для оркестрування існують різні розподілені фреймворки з відкритим вихідним кодом, доступні для оцінювання управління ресурсами та планування кластерів, центрів обробки даних або хмарних середовищ, доступні різні варіанти розподілених фреймворків з відкритим кодом. Ці фреймворки в основному використовуються з програмами (наприклад, Kafka, Hadoop, Elasticsearch, Spark тощо) і надають API для керування ресурсами, оркестрування та планування. Одними з найвідоміших програмних засобів для програмного визначення та планування завдань є Apache Mesos [35] і Apache YARN (для керування кластером і моніторингу виконаних завдань), Apache Zookeeper, Apache AirFlow [36] і Apache Oozie (для розподіленої координації, планування робочого процесу в кластері), Apache TEZ, Apache Ambari (надає інтерфейс керування), Kubernetes [37] і Docker Swarm.

У цьому підрозділі ми подамо основні доступні в літературі підходи до управління різними архітектурними компонентами мережі 5G. У ході розгляду ми зазначаємо, що різноманітність проблем на кожному архітектурному рівні, звичайно, потребує різних підходів.

Також розглянемо керування ресурсами RAN за допомогою виявлення моделей поведінки користувачів і розподілу ресурсів RAN за допомогою таких методів навчання з підкріпленням (RL), як підходи Q-навчання. На завершення проаналізуємо традиційні методи оптимізації та підходи на основі штучного інтелекту для вирішення проблем оркестрування мережі, що містить будь-які обчислювальні ресурси. Пізніше будуть наведені методи SoA для досягнення нарізки площини даних через ізоляцію співіснуючого трафіку. Розглядаючи ширше різні сегменти мережі 5G,

розглянемо різні програмні методи та алгоритми для керування оптичними ресурсами.

Також розглянемо надзвичайно важливе питання об'єднання ресурсів і методи ШІ, які використовуються для сприяння реалізації процедур об'єднання.

Існує безліч запропонованих досліджень, які намагаються проаналізувати, використати та керувати моделями поведінки користувачів і мережі, а також запровадити інноваційні механізми, засновані як на контрольованих, так і на неконтрольованих підходах [38–42] до ефективного та проактивного керування ресурсами RAN. У [38] наведено структуру виявлення знань у базах даних (KDD), яка обслуговує вилучення та використання шаблонів поведінки користувачів з мережної та сервісної інформації за допомогою K-середніх і спектральної кластеризації, наївного Байєса та алгоритмів навчання дерева рішень. Автори в [39] знайомлять зі структурою для кластеризації трафіку та прогнозування з використанням K-Means, нейронних мереж (NN) і стохастичних процесів, щоб керувати поведінкою трафіку для величезної кількості базових станцій. Валторта та ін. [40] пропонують методологію профілювання пристроїв IoT з урахуванням поведінки радіо та мережі за допомогою кластеризації K-Means. Інша робота [41] застосовує K-середні та методи ієрархічної кластеризації даних мобільної мережі для виявлення аномалій у мобільних бездротових мережах. У [42] автори пропонують ієрархічну кластеризацію з урахуванням розташування та трафіку, а також покращену версію поширення спорідненості для веж стільникового зв'язку та вводять три алгоритми з урахуванням розташування, покращені мобільністю та передачею для упакування модулів базової смуги (BBU). Крім добре встановлених і вдосконалених методів машинного навчання, існують численні дослідження, які впроваджують нові та інноваційні алгоритми кластеризації для використання агрегованих даних трафіку з мережі в [43–45]. У [43] автори пропонують як статичний, так і динамічний алгоритми кластеризації вибору мережі для розподілу потоків трафіку в різні мережні інтерфейси на основі характеристик потоку. У [44] автори пропонують спеціальну та адаптивну кластеризації з урахуванням мобільності комірок із неузгодженою координованою багатоточковою спільною передачею (CoMP-JT), спрямовані на налаштування розміру кластера комірок окремо для кожного користувача.

З появою алгоритмів і методів машинного навчання фокус проблем розподілу ресурсів у багатьох випадках зміщувався до використання мережних мереж, а їх адаптивність – до покриття широкої цільової сфери та різних сценаріїв або архітектур зв'язку (від пристрою до пристрою (D2D) зв'язку, Cloud-RAN та ін.); деякі орієнтовні приклади наведені в [6–8], в яких автори намагаються оптимізувати конкретні KPI розподілу ресурсів RAN, беручи до уваги такі різні параметри або обмеження, як доступність

каналу, завади, адаптивні підходи до керування потужністю, покриття послуг, кількість орендарів та ін.

У поєднанні з мережними мережами прийняття підходу Q-навчання, тобто безмодельного алгоритму навчання з підкріпленням, є багатообіцяючим, його можна адаптувати до подолання складності та обчислювальних вимог задач оптимізації [9]. У [4] завади використовуються як ключовий індикатор для розподілу доступних радіоресурсів і – у поєднанні з кешуванням і обчислювальними ресурсами – цілей для мінімізації наскрізної затримки.

Дотримуючись шляху завад, можна мінімізувати їх, контролюючи потужність передачі базових станцій [5]. Крім того, автори в [2], беручи до уваги інтерференцію між стільниками та розширення діапазону стільникового зв'язку, мають на меті збільшити пропускну спроможність користувача та зменшити збій хендовера, керуючи передачею потужності базових станцій гетерогенної мережі. Інший підхід прийнято Do and Koo [3], де автори мінімізують енергоспоживання за допомогою базових станцій, що їх живлять відновлювані джерела енергії. Абіко та ін. [4] пропонують більш динамічний підхід, вводячи блоки ресурсів для кількісного визначення змінної кількості необхідних сегментів і використовуючи поділ часу та частоти для розподілу мінімальної кількості блоків ресурсів. У [5] автори використовують Deep-Q-Learning (DQL) для допомоги у розподілі ресурсів LoRaWAN, де обсяг пам'яті для обчислення розподілу каналів обмежений.

У цьому підрозділі наведено приклад застосування AI/ML для керування ресурсами RAN, зокрема оптимізації динамічного та спільного вибору політик оркестрування радіо та обчислень. Цю проблему оптимізації можна сформулювати як проблему контекстного бандита (CB), яка є окремим випадком навчання з підкріпленням (RL). Там спостерігають вектор контексту, вибирають дію та отримують сигнал винагороди як зворотний зв'язок, послідовно на різних етапах часу. Мета полягає в тому, щоб знайти політику, яка відображає контексти введення в політику керування або дії, які максимізують очікувану винагороду. Приклад цього підходу наведено у [6], де визначено вказані нижче аспекти моделі CB.

- Простір стану або контексту: враховуючи інформацію користувачького обладнання (UE), vRAN створює великий вектор стану з розміром буфера та інформацією SNR, яка пізніше ущільнюється у вектор контексту за допомогою розрідженого автоматичного кодувальника.

- Простір дій: виконуються дії як з комп'ютерного керування, так і з радіоуправління. Для першого [6] виділяє частку процесорного часу c_i для кожної vBS i . Для останнього агент встановлює верхню межу прийнятного індексу MCS m_i .

- Винагороди: конфігурація функції винагород залежить від мети системи. Тут розглядається подвійна мета: (1) мінімізація операційних витрат через резервування процесора та (2) максимізація продуктивності шляхом зменшення частоти помилок декодування та затримки. Функція

винагороди розроблена відповідно, оскільки: (1) тривале заповнення буфера (яке є основним джерелом затримки) штрафує винагороду, (2) ненульові помилки декодування через дефіцит ЦП (що спричиняє втрати пропускної спроможності) штрафує винагороду і (3) менші призначення ЦП збільшують винагороду.

Рисунок 2.6 ілюструє замкнутий цикл прийняття рішень із застосуванням формулювання RL, наведеного вище. Кожен період або етап оркестрування розділений на T слотів моніторингу. Наприкінці кожного слота t $vrAI$ агрегує такі вибірки радіозв'язку, як середнє SNR і нові дані, які очікують на передачу, зібрані під час останнього слота для всіх користувачів у кожній vBS . У результаті на початку кожного етапу $vrAI$ кодує вектор контексту, що містить T зразків радіоінформації, такої як SNR, і завантаження даних. Потім $vrAI$ відображає закодоване подання такого вектора контексту в пару керувальних дій, як визначено вище, щоб застосувати їх як політики планування на наступному етапі.

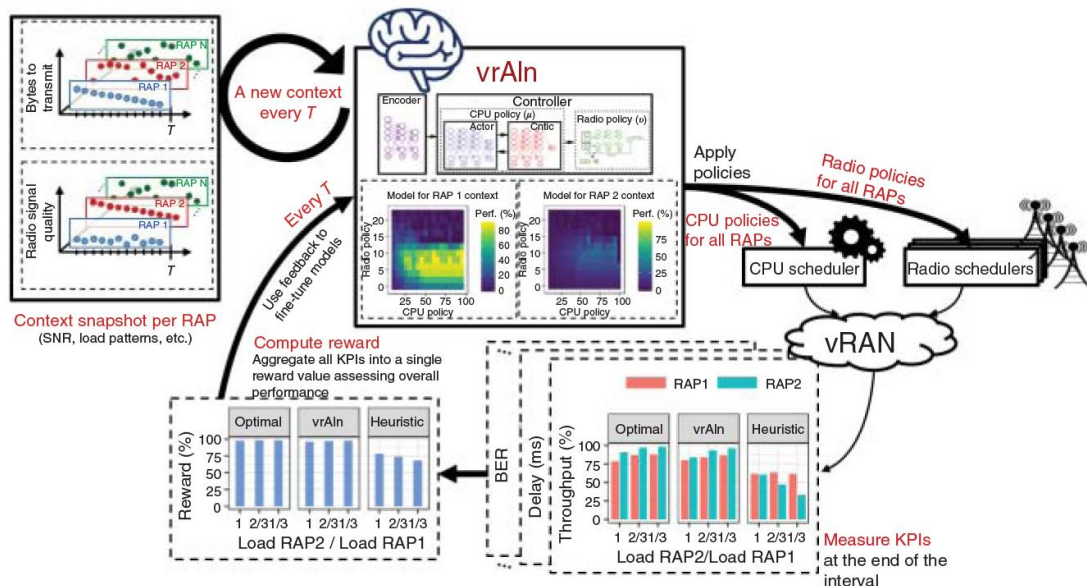


Рисунок 2.6 – Схема керування $vrAI$

Наприкінці цього етапу показники ефективності збираються та кодується в сигнал винагороди, який повертається у $vrAI$ для оптимізації внутрішніх мереж $vrAI$.

Було запропоновано кілька дослідницьких робіт у напрямку застосування концепції NFV для мереж 5G. Вертикальні служби дійсно можна визначити як набір підключених VNF, часто поданих у вигляді графу, які можуть бути розгорнуті на обчислювальних ресурсах, сховищах і мережних ресурсах в інфраструктурі MNO. Для ефективної підтримки різних вертикальних служб також з'явилася парадигма нарізки мережі, завдяки чому оператори мобільного зв'язку можуть розгортати різні служби, забезпечуючи для кожної з них свої вимоги до ізоляції, а також

необхідні КРІ, незважаючи на обмежені доступні ресурси. Зрізання мережі також підтримує складені послуги, тобто такі, де набір VNF містить підмножини, кожна з яких відповідає службі підзрізу.

Щоб створити зріз, MNO мають вирішити, де розмістити кожен VNF і виділити необхідні ресурси (наприклад, віртуальні машини або контейнери та віртуальні посилання, що їх з'єднують). Динамічне розміщення VNF поблизу межі мережі, а також через хмару, край і туман було ретельно вивчено. Оптимальний вибір для розміщення VNF, як відомо, є недетермінованим поліноміальним часом (NP). Тому стає все більш важливим використовувати наближені лінійні моделі для ефективної підтримки прийняття вирішень у процесі оркестрування послуг.

Типовий сценарій, який розглядається для мереж 5G на основі NFV, передбачає, що ухвали про розміщення VNF приймаються NFV Orchestrator (NFVO), визначеним у структурі ETSI MANO [7]. Однак ETSI [5] визначає чотири рівні деталізації щодо розміщення: окремий хост, зона (тобто набір хостів із загальними характеристиками), група зон і точка присутності (PoP) (наприклад, центр обробки даних). Реалізація мобільних мереж у реальному світі, також [6], передбачає, що NFVO або подібні організації ухвалюють свої рекомендації на рівні PoP. Натомість остаточні вирішення розміщення та спільного використання в окремих точках доступу можуть прийматися іншими організаціями під іншими назвами та з невеликими відмінностями між вирішеннями IETF, альянсу NGMN та 5G PPP. Останній, зокрема, містить програмно-визначений координатор мобільної мережі (SDM-X), як показано на рисунку 2.7. SDM-X працює на нижчому рівні абстракції, ніж NFVO, і вирішує як розміщувати VNF всередині PoP. Зокрема, для кожної послуги, щойно запитана вертикаллю, SDM-X вирішує: (1) чи може будь-яка підмножина VNF служби повторно використовувати існуючий підзріз; (2) якщо це неможливо, який PoP має розмістити віртуальну машину (VM), що реалізує VNF; (3) як розподілити (враховуючи збільшення/зменшення масштабу) обчислювальну спроможність віртуальних машин у межах PoP.

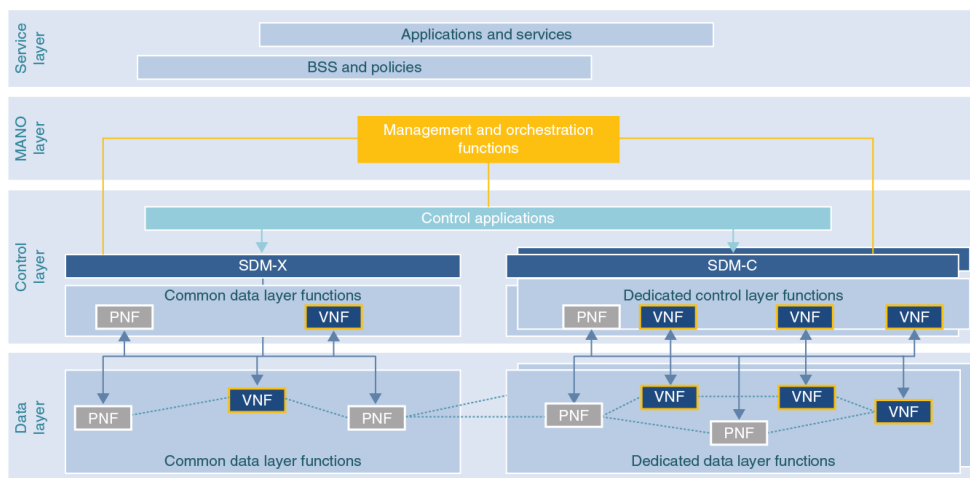


Рисунок 2.7 – Архітектура мережі 5G-PPP

Що стосується існуючих розв'язань проблеми розміщення VNF, існує безліч наукових робіт, які формулюють цю проблему як цілочислове лінійне програмування (ILP), враховуючи різноманітні KPI та накладаючи обмеження, пов'язані з ресурсами, щоб досягти цільової продуктивності [8–16]. Зокрема, в [9] Sun та ін. запропонували економічний евристичний офлайн-алгоритм, який розширено, щоб передбачити майбутні потреби VNF і зменшити затримку налаштування ланцюжків функцій обслуговування (SFC). Арутюнян та ін. [15] пропонують евристичний підхід, що демонструє компроміс між оптимальністю та масштабованістю розміщення VNF у великомасштабних середовищах. У [6] Richard Cziva та ін. використовували принципи теорії оптимальної зупинки, щоб визначити, коли потрібно повторно оцінити рекомендацію щодо розміщення, враховуючи зміни в значеннях затримки в реальному сценарії та, якщо необхідно, перенести VNF. У [1] Luizelli та ін. задіюють змінний пошук сусідства (VNS), орієнтований на мінімізацію кількості екземплярів VNF, відображених на фізичних вузлах (PN), і операційних витрат відповідно.

Оскільки ILP стає обчислювально нерозбірливим зі збільшенням розміру мобільної мережі, у кількох роботах пропонуються такі різні вирішення, як нові евристики з урахуванням складності, генетичні алгоритми та навчання дерева рішень [2–5]. Важливо, що обидва [5, 6] також вирішують проблему спільного використання підзрізів різними екземплярами служби, щоб мінімізувати вартість розгортання, одночасно задовольняючи можливі вимоги ізоляції.

Крім того, багато останніх дослідницьких робіт розглядають проблему розміщення VNF з використанням моделей глибокого навчання з підкріпленням (DRL) завдяки їх ефективності та застосовності в обставинах, коли реакція середовища на дію апріорі невідома, але її потрібно вивчити [6–7].

Іншою важливою проблемою дослідження NFV є коливання навантаження трафіку, яке потрібно ефективно обробляти NS, щоб відповідати необхідним SLA. Це призводить до дій динамічного масштабування для адаптації виділених ресурсів для певних екземплярів VNF, що належать до NS, щоб забезпечити потрібну продуктивність. У літературі методи автоматичного масштабування поділяються, в основному, на дві основні групи, а саме: реактивний і проактивний підходи. Реактивний підхід базується на застосуванні порогових правил і наразі розглядається як вбудований механізм у поточні платформи керування та оркестрування MANO з відкритим кодом (MANO), як-от OSM. Тим не менш, вибір порогів є складним і може бути неефективним у разі раптових стрибків трафіку. Для вирішення цієї проблеми кілька дослідницьких робіт пропонують нові евристичні та адаптивні рішення [1–4], а також моделі нейронних мереж [5, 7], які допомагають приймати проактивні рішення щодо автоматичного масштабування шляхом періодичного оцінювання необхідної кількості екземплярів VNF, що працюють на кожному хості PN.

Централізація інтелекту мережі в SDN є перевагою для додатків, які не мають строгих вимог до реального часу та залежать від стану глобальної мережі. Однак, коли служба використовує інформацію про місцевий стан, наприклад щоб підтримувати QoS, той же рівень гнучкості має підтримуватися на площині даних.

У сучасних програмованих мережних пристроях логіка керування трафіком не програмується. У попередніх дослідженнях щодо розширення можливостей програмування на площину даних [7] підкреслювалося важливість налаштування таких алгоритмів керування трафіком, як стратегії постановки в чергу та методи планування, до вимог додатків. Зокрема, Sivaraman et al. [7] стверджують, аналізуючи різні підходи до керування активними чергами (AQM), що не існує універсального алгоритму.

Крім того, вони забезпечують можливість програмування на площині даних, додаючи FPGA до швидкого шляху апаратного комутатора з простим інтерфейсом до черг пакетів, реалізуючи керування активними чергами з контрольованою затримкою (CODEL) і випадкове раннє виявлення (RED) як доказ концепції. У тому ж напрямку Kundel et al. [8] продемонстрували, що можна реалізувати такі алгоритми для програмованих площин даних P4, ілюструючи можливості та обмеження P4. У [9] надається реалізація P12 [80] у P4, яка також демонструє шляхи подолання обмежень P4 для розробки алгоритмів AQM. Нарешті, в [1] обидві схеми розширеного пропорційного інтегрального контролера (PIE) і RED AQM реалізовані та перевірені в контексті P4. Ці підходи покращують використання черги в загальних мережних інфраструктурах (з'єднаннях), але вони не можуть повністю забезпечити пропускну спроможність кожного клієнта (зрізу) і гарантії затримки.

Попередні зусилля щодо підтримки кількох мережних контекстів в одній площині даних використовують підходи, подібні до віртуалізації на основі гіпервізора, наприклад NuPer4 [2] і HyperV [3] для базового програмного забезпечення VMV2 [4]. NuPer4 був першою спробою використати можливості реконфігурації P4, щоб досягти віртуалізації площини даних. Програма, схожа на гіпервізор, заснована мовою P4, використовується для забезпечення часткової віртуалізації площини даних, що дозволяє виконувати кілька програм P4 ізольовано на одному пристрої обробки пакетів. Таблиця використовується для розподілу мережного трафіку клієнта між програмами P4 орендарів.

Оновлюючи певні записи таблиці, гіпервізор може (де)активувати програми під час їх виконання. Будова NuPer4 для підтримки складних функціональних можливостей і оперативної реконфігурації програми значно знижує продуктивність. HyperV розширив NuPer4 і запропонував нові методи для забезпечення повної віртуалізації, такі як послідовність потоку керування для інтерпретації віртуалізованих даних у єдиний лінійний шаблон і динамічне відображення довільних послідовностей

етапів у віртуалізованому конвеєрі на апаратну площину даних з обмеженими ресурсами. Архітектура HyperV представлена більш детально в [8], охоплюючи реалізацію для DPDK-формату [16]. В платформі P4Visor [7] ще більше зменшено ресурси, необхідні для підтримки віртуалізованих програмованих площин даних порівняно з [2, 3], використовуючи полегшений примітив віртуалізації для програм P4 за допомогою злиття коду. Платформа P4Visor, що працює між програмами P4 і програмованою площиною даних, використовує оптимізацію компілятора та аналіз програм для досягнення ефективного злиття вихідного коду; насправді, платформа приймає як вхідні дані кілька програм P4 і створює одну програму P4, зберігаючи їх функціональні можливості. Порівняльний опис усіх існуючих на сьогоднішній день підходів на основі гіпервізора чи компілятора можна знайти в [18].

Мережі 5G охоплюють численні сегменти мережі (тобто доступ, метро та ядро), а також такі технології, як мобільні, фіксовані та оптичні, щоб забезпечити різні типи послуг, необхідні вертикалям 5G, разом із виконанням їхніх KPI.

У цьому контексті оптичні технології вважаються необхідними для фактичного задоволення деяких із цих вимог і потреб у послугах 5G, використовуючи величезну транспортну ємність для з'єднання розподілених хмарних обчислень і центрів зберігання (тобто периферійних і основних DC). Зокрема, оптичні технології забезпечують провідне рішення для досягнення ефективної мережної інфраструктури, що задовольняє очікувані вимоги до послуг 5G з точки зору високошвидкісного підключення з низькою затримкою, енергоефективності тощо [9].

Потрібні транспортні оптичні ресурси (тобто оптичний спектр, передавач, приймачі тощо), що підтримують послуги 5G (наприклад, між віддаленими DC), вибираються та програмуються за допомогою спеціального оптичного контролера SDN. Загалом, технологічні контролери SDN централізовано керуються об'єктом вищого рівня, який називається оркестровником мережі (ресурсів). Прикладом цього є IETF Application Based Network Orchestrator (ABNO) [10]. Як правило, він спирається на ієрархічну архітектуру керування, де батьківський контролер забезпечує всеохоплювальний контроль над пулом багатотехнологічних доменів, охоплюючи пакетну та оптичну комутації [11]. Крім того, для розгортання послуг і програм 5G, які потребують пам'яті, обчислювальних і мережних ресурсів, оркестровник мережі (наприклад, ABNO) поводить як контролер VIM, який координується екземпляром оркестровника служби (тобто, MANO). У цій спільній оркестрації IT/хмари та мережі спеціальний елемент для керування мережним з'єднанням називається диспетчером інформації глобальної мережі (WIM).

У рамках оптичних мереж добре відомо, що традиційні мережі DWDM з фіксованою сіткою еволюціонувати до так званих гнучких мереж DWDM. Це робиться для досягнення більш ефективного використання оптичного

спектра [2], сприяючи спеціальному розподілу оптичного спектра, достатнього для потреб послуги. З цією метою гнучкі мережі DWDM використовують гнучкість, яку забезпечують трансивери зі змінною смугою пропускання (SBVT). Впровадження гнучких мереж DWDM із пристроями SBVT забезпечує ефективну транспортну інфраструктуру (наприклад, з'єднання віддалених об'єктів постійного струму) для дотримання суворих вимог до послуг 5G щодо високої пропускної спроможності та низької затримки. З цією метою оптичний SDN, що керує транспортною інфраструктурою DWDM, бере на себе вибір і налаштування всіх залучених елементів мережі, таких як оптичні комутатори, оптичний спектр каналів (тобто центральна частота та ширина слота) і параметри SBVT кінцевої точки (наприклад, формат модуляції) під час налаштування цільових міжпостійних оптичних потоків. Контролер SDN зобов'язаний: (1) гарантувати швидкість передачі даних послуги 5G; (2) мати справу з будь-якими обмеженнями фізичної передачі даних, що накладаються оптичною технологією (наприклад, максимально досяжна швидкість передачі даних), і (3) виконувати такі внутрішні технологічні обмеження, як неперервність спектра [3]. Проблема вибору просторових і спектральних шляхів, ресурсів і параметрів SBVT зазвичай вирішується за допомогою так званих алгоритмів призначення спектра маршрутизації та модуляції (RS(M)A). В останні роки було зроблено значний внесок у алгоритми маршрутизації та призначення спектра (RSA). Повний огляд, що стосується багатьох аспектів алгоритму RSA, можна знайти в [4].

Отже, оптичний контролер SDN отримує вимоги до з'єднання від оркестровника мережі. Ці вимоги визначають пари вихідних і цільових DC, які розміщують VNF, відповідно до підсумкового VNF Forwarding Graph для цільової служби 5G, пропускної спроможності, максимальної наскрізної затримки тощо [5]. Потім ці обмеження використовуються як вхідні дані для алгоритму RSA контролера SDN. Вихідні дані алгоритму RSA визначають набір оптичних ресурсів (тобто, пройдених вузлів, каналів, спектра, параметрів SBVT), щоб задовольнити вимоги підключення. Потім ці ресурси розподіляються згідно з відповідним інтерфейсом керування між контролером SDN та агентами, що обробляють кожен мережний елемент у базовій інфраструктурі оптичної мережі.

Очікується, що мобільні мережі наступного покоління працюватимуть у дуже неоднорідних середовищах. Керування мережею таких сценаріїв із залученням різних технологій і мережних сегментів потребує багатодоменого оркестрування.

Однак у мережах мобільного зв'язку наступного покоління термін «домен» має мати додаткове й суттєве значення, головним чином накладене вимогою задоволення різноманітних потреб різних типів користувачів. Деякі потрібні функції для складання NS або розподілу ресурсів інфраструктури можуть надаватися різними організаціями, відомими як адміністративні домени (AD), підкоряючись таким різним критеріям, як

дефіцит ресурсів, проста доступність служби або можливість розгортання служби, що задовольняє різні вимоги. Залежно від того, як цей процес виконується між AD, можна розрізнити об'єднання ресурсів і служб.

Отже, об'єднання ресурсів можна визначити як процес, за допомогою якого AD-споживач потребує управління ресурсами інфраструктури AD-постачальника для розгортання NS (або його частини). З іншого боку, об'єднання послуг – це процес, у якому споживацька AD потребує розгортання NS у домені постачальника, тоді як домен постачальника зберігає повне керування ресурсами своєї інфраструктури.

У літературі вирішення проблеми федерації можна розділити на дві основні групи. По-перше, деякі роботи стосуються процедур та інтерфейсів для ефективного процесу об'єднання (в обох формах), щоб дозволити реальну інстанціацію NS серед різних AD. По-друге, інші роботи стосуються проблеми розподілу різних складових частин NS серед кількох AD.

Щодо визначення процедур та інтерфейсів для виконання операцій об'єднання, більшість робіт надходить від організацій розробки стандартів (SDO), таких як Open Networking Foundation (ONF) [6], Metro Ethernet Forum (MEF) [7] і робочі групи ETSI NFV [8, 9].

Що стосується проблеми розподілу NS між кількома доменами, в літературі пропонуються, в основному, два типи підходів, а саме: централізований і розподілений. У централізованому підході [10, 11] центральна стороння організація має повні знання про доступні ресурси і розділяє ланцюг NS на «підланцюги», засновані переважно на моделях ILP. У розподіленому підході [12, 13] AD встановлюють однорангові відносини. Відсутність інформації про топологію різних AD та доступність ресурсів ускладнює вирішення проблеми розповсюдження в кількох доменах, і для обміну між суміжними AD потрібно більше повідомлень. Однак саме цей брак важливої інформації вважається привабливим для мережних провайдерів.

AI/ML можна застосовувати різними способами для сприяння реалізації процедур об'єднання. ETSI ZSM [14, 15] зосереджується на підході централізованої федерації та застосуванні AI/ML як частини процесів самопланування, самооптимізації, самовідновлення та самозахисту. На противагу застосуванню моделей AI/ML, для сприяння оркеструванню процедур об'єднання, він застосовує об'єднання для навчання моделей AI/ML через Federated ML [16]. У цьому випадку навчання моделей AI/ML відбувається в різних об'єднаних AD, але дані зберігаються конфіденційними та не передаються об'єднаним AD. У деяких випадках для підвищення продуктивності об'єданого ML застосовується технологія розподіленої книги (DLT), така як Blockchain [17].

Ми навели короткий огляд проблем, з якими стикаються розробники систем 5G, що відповідають KPI, необхідних для мобільних послуг наступного покоління. Зокрема, ми зосередилися на таких аспектах, як

мережі радіодоступу, оптичні мережі, керування площиною даних, нарізка мережі та оркестрування послуг, і ми також навели найвідоміші рішення, існуючі в науковій літературі, а також ті, що пропонуються відповідними організаціями з розробки стандартів. При цьому ми виділили два основні підходи, які знадобляться для розробки систем 5G і не тільки: автономне керування мережею на основі даних і об'єднання різних AD.

З огляду на те, що системи 5G використовують технології, які все ще перебувають на ранній стадії, багато аспектів потребують подальшого вивчення. Зокрема, в рамках проєкту 5Growth [18], нові інтелектуальні алгоритми оркестрування ресурсів будуть розроблені та реалізовані таким чином, щоб відповідати вимогам щодо надійності, пропускнуєї спроможності та затримки. У цьому контексті надзвичайно важливо розробити підходи AI/ML, які можуть привести не лише до ефективного вирішення виконання KPI та використання ресурсів, але й до масштабованих і гнучких механізмів, які можуть ефективно працювати з різними службами та значно скоротити розгортання послуг.

З цією метою та спираючись на роботу, виконану в попередніх проєктах 5G PPP, таких як 5G-TRANSFORMER [6], 5Growth [18] спрямований на розробку архітектури на основі AI/ML, де розглядаються різні аспекти оркестрування послуг і ресурсів на різних архітектурних шарах.

По-перше, отримавши запит на вертикальну послугу, суб'єкт, який називається вертикальним зрізом, подбає про: (1) оцінювання того, чи можна задовольнити послугу, враховуючи кількість ресурсів, які має право використовувати вертикаль, (2) який тип зрізу має бути створений, (3) який (за наявності) підрозділ(и) може(уть) повторно використовувати для розгортання служби.

Для виконання цих завдань вертикальний зріз використовуватиме дані, зібрані за допомогою платформи моніторингу, запитуватиме оркестровник основного сервісу щодо інформації про доступність ресурсів, для подачі моделі AI/ML для прийняття розумних рішень.

Отримавши наведені вище рекомендації, оркестровник служби може виконати фактичне розміщення VNF і належним чином масштабувати ресурси, які будуть розподілені на існуючі підрозділи, щоб задовольнити потреби трафіку та робочого навантаження. Важливо, що оркестровник служби має забезпечувати безперебійну роботу розгорнутих сегментів, незважаючи на навантаження трафіку та робочі умови, що змінюються в часі. Знову ж таки, використовуючи дані моніторингу, може знадобитися подальше масштабування ресурсів, виділених на зріз, щоб досягти цільових KPI. Це може призвести до зміни обсягу ресурсів, призначених віртуальним машинам і розподілених на віртуальних з'єднаннях для роботи з трафіком служби (збільшення/зменшення) або до створення/видалення репліки VNF. Тому важливо розробити рішення AI/ML, які можуть вирішити ці проблеми шляхом прийняття майже оптимальних рішень. З одного боку, підходи до

навчання з підкріпленням будуть сприяти прийняттю рішень у реальному часі шляхом визначення функцій винагороди, які відображають вимоги до послуг. З іншого боку, буде реалізовано гібридний алгоритм, який поєднує генетичні рішення з мережними мережами, націлений на мінімізацію наскрізної затримки, гарантуючи відсутність порушень KPI. Точніше, генетичні алгоритми досліджуються та вибираються як рішення проблеми розміщення VNF через їх швидку збіжність до майже оптимального рішення. У цьому напрямку були обрані NN, щоб зіставити трафік і метрики, що стосуються додатків, та рішення щодо масштабування VNF.

Крім того, на рівні ресурсів платформа 5Growth запровадить програмовані алгоритми керування трафіком із урахуванням SLA на основі ML на площині даних (наприклад, планування, активне керування чергою тощо), зосереджуючись на потенційно вузьких місцях (наприклад, RAN), забезпечуючи продуктивність кожного сегмента через взаємодію замкнутого циклу. З цією метою потрібно розробити відповідні абстракції зрізу для програмованих площин даних, уможливаючи також програмоване керування трафіком.

Поєднання керування на основі ML із повністю програмованими площинами даних забезпечить додаткові ступені свободи для конфігурації та налаштування сегментів згідно з вимогами до продуктивності.

Для реалізації вищезазначеного бачення зрозуміло, що на різних архітектурних рівнях знадобляться попередньо підготовлені моделі AI/ML. Тому надзвичайно важливо розробити спеціальну платформу для того, щоб зробити доступними офлайн-навчені моделі, а також забезпечити Vertical Slicer, Service Orchestrator і Resource Layer моделями, які навчаються на місці, використовуючи дані, зібрані за допомогою моніторингу системи. Це справді ядро замкнутої системи, як це також передбачено ETSI ZSM [19]. Потрібно буде розробити відкриті інтерфейси, які дозволять визначати характеристики послуг і цільові KPI для вертикалі, передавати їх як гіперпараметри до вертикального зрізу, а потім до інших рівнів мережної архітектури. Таким чином, механізми штучного інтелекту/ML можна адаптувати до конкретних вимог до послуг і додатків, що приведе до повністю автоматизованої системи 5G.

Однак ефективного керування обчислювальними ресурсами та ресурсами зберігання недостатньо для досягнення цільових KPI, необхідних для вертикалей: ресурси RAN також потрібно розумно розподіляти. Зокрема, в надщільних екосистемах RAN, передбачених для 5G (і не тільки), підходи AI/ML потребуватимуть спільних стратегій, які враховуватимуть усі типи ресурсів. Розподілене навчання, а також методи розподіленого висновку вже пропонуються в літературі, наведеній раніше, щоб використовувати збільшені можливості обробки різноманітних елементів RAN, навіть використовуючи можливості пристрою кінцевого користувача, коли це можливо. Як можна зробити висновок з попередніх парадигм, концепції AI/ML поступово стають структурними компонентами мережі,

запроваджуючи нові можливості для інтелектуального керування ресурсами RAN, профілювання користувачів і мережних пристроїв, а також методи розподілу спектра.

Окрім спільного керування ресурсами, однією з найважливіших проблем є те, що досі алгоритми AI/ML, мережні та комунікаційні протоколи розроблялися окремо. Спільне керування ресурсами RAN і розробка алгоритмів AI/ML мають мати одні з високих пріоритетів на шляху до справжньої мережної парадигми, де підходи до кодування та обробки сигналів інтегровані з інфраструктурою AI. Це буде реалізовано шляхом визначення загальних вимог та обмежень, які є результатом двох доменів; наприклад, такі спільні підходи можуть містити розгляд різних методів зменшення розмірності або кодування даних для пристроїв з обмеженими обчислювальними можливостями (таких як вузли Інтернету речей), адаптивної градієнтної агрегації для покращення стійкості або методів спільного кодування каналів і ущільнення зображень у заводових бездротових середовищах.

У цьому сенсі Альянс O-RAN визначає серію варіантів використання, які поєднують моделі ML з відкритими та віртуалізованими RAN. Приклади таких варіантів використання:

- динамічні радіоресурси безпілотних літальних апаратів (БПЛА) на основі траєкторії польоту, де управління БПЛА та розподіл радіоресурсів спільно оптимізовано;
- спільне використання RAN, де віртуалізовані (можливо, спеціально створені) екземпляри точок радіодоступу від різних операторів спільно використовують загальну обчислювальну інфраструктуру на межі хмари та/або радіочастот;
- контекстне динамічне управління хэндовером для V2X, де моделі машинного навчання допомагають у прогнозуванні та класифікації завдань для налаштування послідовностей хэндовера з деталізацією UE.

3GPP вже презентував нову функцію аналізу мережних даних (NWDAF) у Rel.16 [11], яка вказує на поступове впровадження концепцій AI/ML в архітектуру основної мережі, також з точки зору стандартизації.

Наразі ця функція має обмежену функціональність і розгортається лише як частина базової мережі (5GC), щоб полегшити маніпулювання політиками операторів.

Крім того, наразі аналітика даних обмежена лише інформацією, орієнтованою на 3GPP. Взятися вищезазначену мережну парадигму з підтримкою штучного інтелекту як орієнтир для проектування, еволюція такої NF до розподіленого, багатодоменого, об'єднаного підходу, заснованого на навчанні, використовуючи також інформацію про ресурси з мереж, що не належать до 3GPP, може потенційно підвищити можливості ШІ мережі в напрямку безперебійного та більш гнучкого керування ресурсами RAN на межі мережі. Крім того, виділення моделей поведінки користувачів і мережі та їх використання для прогнозованого розподілу

ресурсів RAN забезпечить значні додаткові переваги для поточних підходів до керування ресурсами RAN. Однак такий механізм потребуватиме радикальних удосконалень поточної архітектури та роботи NWDAF, уможлиблюючи підхід до розподіленого вилучення профілів, використовуючи обчислювальну потужність крайових вузлів, охоплюючи навіть UE.

Нарешті, важливим аспектом, що охоплює різні домени, є федерація.

Як зазначалося раніше, у проєктах 5G-TRANSFORMER і 5Growth були виявлені важливі прогалини та можливі рішення. Зокрема, робота в ньому розширює оркестрування служби кількох AD, щоб охопити операції оркестрування та ефективно з'єднати різні NS, що працюють у різних AD. У цьому контексті цікавий напрямок досліджень полягає в розробці архітектурних рішень, які використовують DLT для встановлення адміністративних зв'язків між доменами та підходами Q-навчання для прийняття кращих рішень щодо розподілу послуг між одноранговими доменами, у яких можуть бути задіяні технічні та бізнес-параметри, як запропоновано в [11, 12].

3 МОДЕЛІ ТА АЛГОРИТМИ ПРОГРАМНОГО КЕРУВАННЯ

3.1 Штучний інтелект в мережах 5G

Парадигми мережного програмного забезпечення, такі як програмно-визначена мережа (SDN) і віртуалізація мережних функцій (NFV), а також методи нарізки мережі прокладають шлях для гнучких і програмованих мереж 5G [1]. У той же час неоднорідність послуг і пристроїв, які потребують підтримки, потребує високого рівня автоматизації, щоб забезпечити швидко адаптацію до мінливих умов мережі та підтримувати ефективність використання ресурсів і якість обслуговування.

Завдяки численним досягненням у сферах машинного навчання (ML) і штучного інтелекту (AI), а також їх успішному застосуванню в широкому діапазоні галузей, у поточній дослідницькій роботі обговорюється використання методів ML/AI для керування комунікаційними мережами та послугами [2]. Щоб забезпечити бездоганну інтеграцію та максимізувати переваги автоматизації, механізми на основі штучного інтелекту мають бути вбудовані в архітектуру 5G і підключені до систем керування та оркестрування (MANO), а також до функцій, які надають дані моніторингу, пов'язані з мережею та послугами.

На рисунку 3.1 показано ключові компоненти інтегрованої мережної архітектури 5G, яка має наскрізне (E2E) нарізання мережі. Використовуючи парадигму нарізки, оператори можуть розділити свої фізичні мережі на декілька віртуальних та ізольованих логічних мереж, які адаптовані до конкретних вимог, і використовувати їх для досягнення диференційованої обробки мережного трафіку. Далі ми шар за шаром описуємо вміст фігури, починаючи з фрагментів E2E на найвищому шарі. Ці фрагменти E2E є віртуальними мережами з вимогами до цільового рівня обслуговування (SLO) і використовуються для досягнення диференційованої обробки трафіку на основі запитуваного типу послуги та типу підписки пристрою, який надсилає запит. Наприклад, різні зрізи E2E можна використовувати для надання послуг потокового відео з інтенсивним використанням пропускної спроможності та медичних послуг із критичними затримками. Ці фрагменти E2E охоплюють кілька таких доменів, як доступ, транспорт і ядро, що мають свої окремі характерні компоненти, стандарти [4, 5] і проблеми. Таким чином, зрізи E2E відображаються на зрізи домену (показані в середньому шарі на рисунку 3.1), які охоплюють відповідну частину мережі та мають власні компоненти MANO. Диференційоване коригування на цьому рівні ґрунтується на угодах про рівень обслуговування (SLA) щодо показників якості обслуговування (QoS).

Щоб відповідати вимогам SLO для сегментів E2E, потрібно вибрати відповідні фрагменти домену для обробки відповідного трафіку. Це зіставлення між сегментами E2E і доменом також дає операторам

можливість отримати вигоду від економії масштабу, оскільки один сегмент домену може обслуговувати кілька сегментів E2E. Наприклад, два сегменти E2E можуть спільно використовувати транспортний і основний сегменти, маючи при цьому окремі сегменти доступу, оскільки останні, як правило, є вузьким місцем. Нарешті, для забезпечення дотримання гарантій SLA потрібне зіставлення між сегментами домену та ресурсами рівня інфраструктури. Рівень інфраструктури складається з фізичних каналів, вузлів, а також фізичних і віртуальних мережних функцій (NF). На цьому рівні оператори знову можуть отримати вигоду від економії на масштабі завдяки спільному використанню інфраструктурних ресурсів між сегментами домену, уникаючи при цьому порушень SLA.

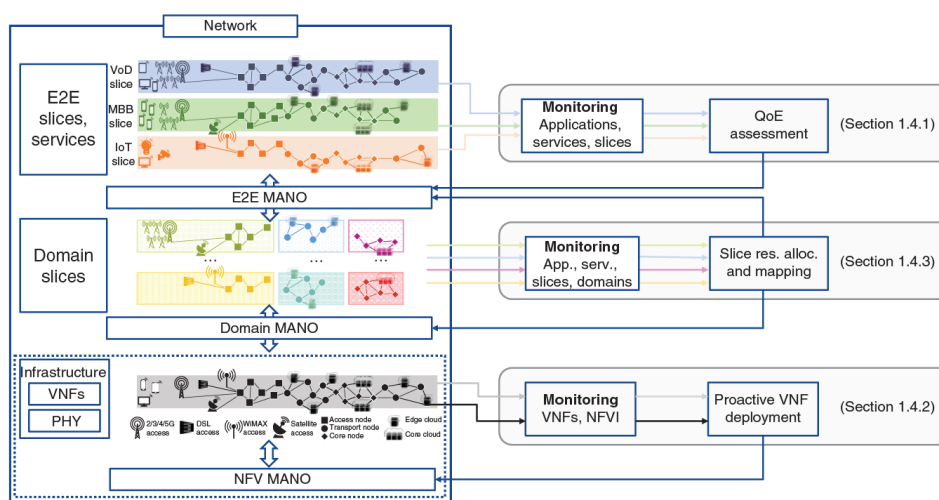


Рисунок 3.1 – Огляд варіантів використання AI/ML в інтегрованій мережі 5G з підтримкою послуг

Для зручності ми опускаємо деякі деталі на рисунку, але хотіли б відзначити два додаткові аспекти: домени можуть бути розбиті рекурсивним способом перед відображенням на фізичні ресурси [6]. Крім того, різні зрізи можуть підтримуватися різними операторами, що створює додаткові проблеми щодо зв'язку та управління.

Методи на основі штучного інтелекту можуть допомогти досягти автономного управління в контексті таких складних систем, які додатково піддаються часовій динаміці.

Зразкові варіанти використання охоплюють моніторинг і контроль додатків на основі якості досвіду (QoE), автоматичне масштабування функцій віртуалізованої мережі (VNF) і автоматичний розподіл ресурсів сегментів.

Перш ніж застосовувати такі механізми на основі штучного інтелекту, потрібно взяти до уваги кілька проблем. До них належать загальні проблеми, як-от збирання і підготовка даних, а також вибір алгоритму та функції. Інші проблеми залежать більше від контексту. Наприклад, потрібно враховувати вплив на саму мережну екосистему, зокрема надійну роботу.

У цьому розділі ми розглянемо три сфери застосування ШІ в екосистемі 5G, наведемо результати тематичних досліджень, щоб висвітлити як загальні, так і конкретні проблеми, які виникають у цьому контексті, і запропонуємо життєздатні шляхи їх подолання.

В подальшому розглянемо роботу щодо загального застосування методів машинного навчання в мережній платформі, а також керування програмними мережами 5G на основі ML та оцінювання QoE. Після цього, наведемо тематичні дослідження, які охоплюють керування QoE, розгортання VNF і керування зрізами. Ми визначимо основні виклики, проілюструємо, як їх можна вирішити, і запропонуємо вказівки для операторів і практиків та завершимо коротким викладом ключових висновків.

У цьому підрозділі ми висвітлимо спільну роботу щодо трьох основних напрямків. По-перше, обговоримо застосування методів ML до мережних проблем загалом. Крім того, вкажемо програми ML у конкретному контексті віртуалізованих мереж і управління QoE.

Поточні та майбутні комунікаційні мережі мають мати справу з неоднорідністю щодо таких численних аспектів, як варіанти використання, програми, пристрої, комунікаційні парадигми та варіанти розгортання рішень на основі NFV [7]. Усі ці фактори сприяють постійному зростанню простору параметрів, у якому потрібно проводити оптимізацію, і, отже, обмежують або перешкоджають застосуванню традиційних методів, таких як ті, що базуються на цілочисловому лінійному програмуванні (ILP) або симуляції. Для досягнення цілей w.r.t. гнучкості, адаптивності та автоматизації [8] останні публікації стосовно мереж пропонують підходи на основі машинного навчання [2, 9].

Такі підходи охоплюють не лише такі загальні мережні проблеми, як виявлення значних збитків [10], але також стосуються проблем, які є специфічними для програмних мереж [11], як-от проблема розміщення контролера SDN [12]. Крім того, інтерес до мереж 5G зумовив прогнозування трафіку [13], прогнозування кількості користувачів [14] і когнітивних мереж [15]. Залежно від таких факторів, як конкретний варіант використання та середовище розгортання, застосовуються механізми, засновані на всіх чотирьох основних парадигмах машинного навчання, тобто контрольоване навчання (SL), напів-SL, неконтрольоване навчання та навчання з підкріпленням.

У віртуалізованих мережах підходи ML були запропоновані у формі контролю допуску на основі нейронних мереж у сценаріях вбудовування віртуальної мережі (VNE) [16], а також призначення ресурсів [17] на основі навчання з підкріпленням.

Крім того, методи ML були застосовані до кількох задач прогнозування попиту, ресурсів і продуктивності в мережах на основі NFV. У той час як приклади прогнозованих метрик практикують використання ЦП екземплярами VNF [18–20] і показники продуктивності в хмарних

середовищах [21], розглянуті механізми ML застосовують нейронні мережі, регресію випадкового лісу та регресію вектора підтримки. Окрім безпосереднього використання, результати моделей прогнозування також можуть бути інтегровані як нові функції для покращення продуктивності механізмів прийняття рішень на основі ML для таких завдань, як автомасштабування VNF [22, 23].

ML широко застосовується для оцінювання QoE за ключовими показниками продуктивності (KPI) на рівні мережі. У цьому контексті багато робіт зосереджені на потоковому відео, оскільки це одне з найпоширеніших програм у сучасних мережах. У зв'язку з постійною тенденцією до шифрування трафіку ML часто застосовується до метаданих трафіку для оцінювання таких QoE-релевантних показників потокового відео, як роздільна спроможність або бітрейт [24–28], а також для прогнозування [29] або класифікації [30, 31] QoE в термінах середньої оцінки параметра (MOS).

Оскільки потокове передавання мобільного відео стає все більш популярним і завдяки таким додатковим ключовим показникам ефективності, як якість каналу, і таким факторам, як характеристики руху клієнтів, знаходження оцінки QoE у мобільних середовищах потребує спеціального оцінювання. Тому Lin et al. [32] зосереджуються на мобільних мережах під час вивчення продуктивності різних класифікаторів і впливу використовуваних функцій, отриманих із даних, пов'язаних з мережею та додатками. Однак бажано оцінювати QoE виключно за допомогою функцій, пов'язаних з мережею, оскільки такі дані зазвичай доступні оператору мобільної мережі (MNO) і їх легше отримати в масштабі, ніж детальну інформацію на рівні програми. Таке рішення наведено в [33], базуючись на мережах довгострокового розвитку (LTE). З впровадженням 5G виникають нові можливості та проблеми для оцінювання QoE на основі ML.

Хоча [34] зосереджено на проблемах і пропонує керовану даними архітектуру для управління QoE, Schwarzmann та ін. [35] обговорюють, як нещодавно введена функція застосування NFs (AF) і функція аналізу мережних даних (NWDAF) можуть підтримувати процес оцінювання QoE. Автори [36] використовують оператор найменшого абсолютного скорочення та вибору (LASSO) для більш глибокого дослідження відповідної статистики в NWDAF з метою досягнення певної точності оцінки QoE. Робота в [37] йде на крок далі за оцінювання QoE, подаючи розподіл ресурсів на основі ML для мереж 5G. Запропонована система визначає рівень продуктивності мережі за допомогою кластеризації, прогнозує KPI мережі за допомогою регресії та динамічно надає ресурси в проактивний спосіб. Одна з нових ключових функцій 5G, нарізка мережі, використовується в поєднанні з глибоким навчанням [38] і навчанням з підкріпленням [39] для досягнення оптимізованого використання ресурсів.

У цьому підрозділі ми розглянемо три варіанти використання в контексті 5G, які більш виграшні порівняно з технологіями на основі 3G. На

додаток до надання огляду основних технічних проблем і пов'язаних із цим труднощів як для традиційних підходів, так і підходів на основі машинного навчання, ми наводимо методологію вирішення цих проблем разом із результатами оцінювання та вказівками до вирішення подібних завдань.

По-перше, ми розглядаємо тему отримання оцінок QoE та дій керування в середовищах, які містять кілька програм і надають доступ лише до показників QoS. По-друге, ми демонструємо, як ML може допомогти в проактивному розгортанні екземплярів VNF для оптимізації як ефективності використання ресурсів, так і часу прийому запитів у контексті Service Function Chaining (SFC). Нарешті, ми обговорюємо потенціал використання інформації з обох сфер для розробки стратегій для розподілу та управління ресурсами сегментів з урахуванням QoE.

Таблиця 3.1 ілюструє, як наші конкретні випадки використання, пов'язані з 5G, вписуються в ландшафт машинного навчання, класифікуючи їх за категоріями, тип проблеми, застосовані алгоритми та процес від збирання даних до попередньої обробки. Ми моделюємо перші два випадки використання як завдання SL, мета яких полягає у вивченні зв'язку між даними моніторингу та неперервними рівнями QoE або набором окремих дій керування відповідно. В обох випадках ми ретельно розробляємо симуляції, які об'єднують відповідні аспекти, пов'язані зі стандартами, комунікаційними інтерфейсами та часовою динамікою.

Згодом ми використовуємо встановлені моделі, щоб збагатити дані базовими мітками істинності для навчання моделей. Щоб оцінити загальну здійсненність підходів на основі машинного навчання для цих завдань, ми застосовуємо існуючі алгоритми до навчальних даних і оцінюємо їх ефективність з точки зору точності.

Таблиця 3.1 – Категоризація охоплених випадків використання

Use case	QoE estimation (Section 5.3.1)	VNF deployment (Section 5.3.2)	Slice mgmt. (Section 5.3.3)
Problem type	Regression-based prediction	Classification-based decision-making	Estimation, prediction, decision making
Algorithm choice	SVM, LASSO	XGBoost, GBM, neural networks	Unsupervised learning (k-means clustering), reinforcement learning
Data collection, analysis, and preparation	Omnet simulation, ITU-T model(s) for ground truth	Simulation, ground truth labeling based on ILP solutions	Simulation, testbed

У випадку керування зрізами ми визначаємо два різні завдання, які можна вирішити за допомогою різних класів алгоритмів ML. Щоб визначити, які служби можна об'єднати в один зріз, підходи до кластеризації можна застосувати до репрезентативних характеристик трафіку. З іншого боку, стратегії, засновані на навчанні з підкріпленням, можна використовувати для динамічного відображення та розподілу

ресурсів між рівнями, наприклад між сегментами домену та фізичними ресурсами.

Керуючись бізнес-стимулами, забезпечення хорошого QoE для клієнтів є важливим для постачальників мереж і постачальників додатків або контенту (CP), таких як YouTube або Netflix. Забезпечення хорошого QoE для кожного користувача залежить від двох ключових факторів: (i) цілісного уявлення про поточний QoE в системі, напр. усі користувачі, підключені до однієї базової станції в мобільній мережі, і (ii) можливість відповідно адаптувати систему, напр. певні конфігурації базової станції. Однак невід’ємна дилема полягає в тому, що лише MNO здатний здійснювати керування мережею з урахуванням QoE, тоді як інформація щодо поточного QoE системи доступна лише для CP. Цю проблему можна подолати за допомогою нещодавно презентованих NF у 5G. Зокрема, AF є фундаментальним для обміну інформацією між MNO та CP. Це стандартизований інтерфейс, який дозволяє стороннім орендарям, напр. CP, передавати таку інформацію, як показники якості додатків або QoE, до MNO. Як наслідок, це дає змогу MNO виконувати контроль ресурсів з урахуванням QoE.

Але, незважаючи на наявність AF, оператор стикається з двома проблемами: (i) MNO залежить від інформації, наданої через сторонній AF, тобто оператор MNO не контролює частоту та кількість інформації, яка передається.

Інформація передається через площину управління і, отже, збільшує кількість дорогого трафіку площини управління. Щоб усунути ці проблеми, другий 5G NF може фундаментально змінити мережний менеджмент NWDAF [40]. Він збирає та надає аналітику з інших NF в архітектурі 5G і, отже, потенційно може ввімкнути моніторинг і керування QoE на основі даних: QoE, отриманий через сторонній AF, можна співвіднести зі статистикою моніторингу мережі, тобто пропускнуою спроможністю (TP), пакетами втрати або індикатором якості каналу (CQI).

Моделі ML, які використовують ці кореляції, здатні оцінити QoE на основі статистики на рівні мережі, яка доступна оператору оператора. Це дозволяє збирати інформацію про QoE системи навіть за відсутності інформації QoE, наданої стороннім AF.

Реалізація такого керування ресурсами керованого ML, з урахуванням QoE, створює широкий спектр проблем, які розглянуті нижче.

- *Ідентифікація відповідних функцій*: алгоритми ML оцінюють QoE на основі різних статистичних даних, так званих функцій. Хоча деякі з цих статистичних даних дуже важливі для процесу оцінювання, інші лише незначно впливають на результат моделі. Завдання розробки функцій, тобто виділення найбільш релевантних функцій і перетворення їх у формат, сумісний з вибраним підходом ML, створює значні труднощі, оскільки потребує як знань предметної області, так і знань про підхід ML. Приклади

аспектів, які потрібно враховувати, містять застосовану техніку ML, середовище (мобільний чи дротовий доступ) і тип послуги.

- *Ідентифікація відповідних моделей*: з широкого спектра доступних методів, заснованих на ML, нам потрібно визначити, який підходить для даної проблеми.

Застосовність конкретної методики залежить не тільки від її ефективності з точки зору точності оцінки, але також від таких факторів, як складність навчання та застосування моделі, а також її адаптивність і зрозумілість. Наприклад, хоча поведінка базових моделей регресії може бути зрозумілою для людей, рішення складних нейронних мереж може бути важко або неможливо зрозуміти, оскільки вони діють як чорний ящик.

- *Обмеження щодо якості та деталізації даних*: можливості AnMNO для моніторингу мережі мають практичні обмеження, що означає, що важливі показники можна зібрати лише для частини активного обладнання користувача (UE).

Крім того, часові інтервали між двома вимірюваннями, напр. TR на базовій станції не може бути довільно коротким. Це означає, що зібрані дані агреговані в часі чи просторі, або доступні лише дані моментального знімка.

- *Кількісне оцінювання витрат на розгортання ML*: розгортання оцінки QoE на основі ML в архітектурі 5G коштує дорого. Сюди входять, серед іншого, витрати на збирання, передачу, обробку та зберігання всіх відповідних даних. Потрібно кількісно визначити такі витрати та дослідити, як такі різні фактори, як деталізація моніторингу чи відповідні функції, впливають на ці витрати.

- *Компроміс між точністю та вартістю*: використовувані функції, алгоритми ML і моніторинг впливають не лише на витрати, але й на точність, якої можна досягти.

Загалом дрібніша деталізація моніторингу та більша кількість функцій забезпечують кращу точність оцінювання. Однак вони збільшують витрати. Аналіз компромісу між вартістю та точністю є критичним завданням, яке потрібно вирішити, коли мова йде про ML для 5G.

- *Інтеграція в архітектуру 5G*: необхідно визначити, як можна інтегрувати оцінку QoE на основі ML відповідно до специфікацій мережної архітектури 5G. Це містить, наприклад, завдання різних зацікавлених сторін і залучених НФ. Потрібно вказати, навчає NWDAF саму модель, чи він оснащений остаточною моделлю, навченою ззовні; передає сторонній AF QoE чи лише відповідні показники, щоб NWDAF отримував QoE за допомогою стандартизованої моделі QoE?

- *Впровадження циклів керування зворотним зв'язком*: доступну QoE тепер можна використовувати для ініціювання автоматизованих дій керування мережею, напр. якщо QoE нижче попередньо визначеного порогу протягом певного часу. Щоб оптимізувати систему для автономного управління QoE, ефекти керувальних дій потрібно відстежувати та адаптувати, якщо це необхідно.

Запропонована нами інтеграція оцінки QoE на основі ML в архітектуру 5G проілюстрована на рисунку 3.2. Розглядаються три фази.

Фаза 1 – збирання даних: сторонній AF передає дані про продуктивність додатків у NWDAF, збагачуючи їх базовою істинністю QoE.

Етап 2 – вибір функцій і навчання моделі: цей етап охоплює типовий конвеєр ML із навчання та тестування відповідної моделі. Зокрема, велика кількість мережних характеристик генерується шляхом статистичної обробки даних моніторингу мережі. Ці функції ранжуються відповідно до їхньої важливості з точки зору оцінки QoE 2. Навчання моделі на основі ML здійснюється з використанням підмножини важливих функцій 3 і базових значень QoE, що зумовлює можливість отримання оцінки QoE 4. Далі оцінюється продуктивність моделі 5. Цей процес можна повторювати для різних наборів функцій і різних моделей 6, доки не буде отримано бажану точність оцінки.

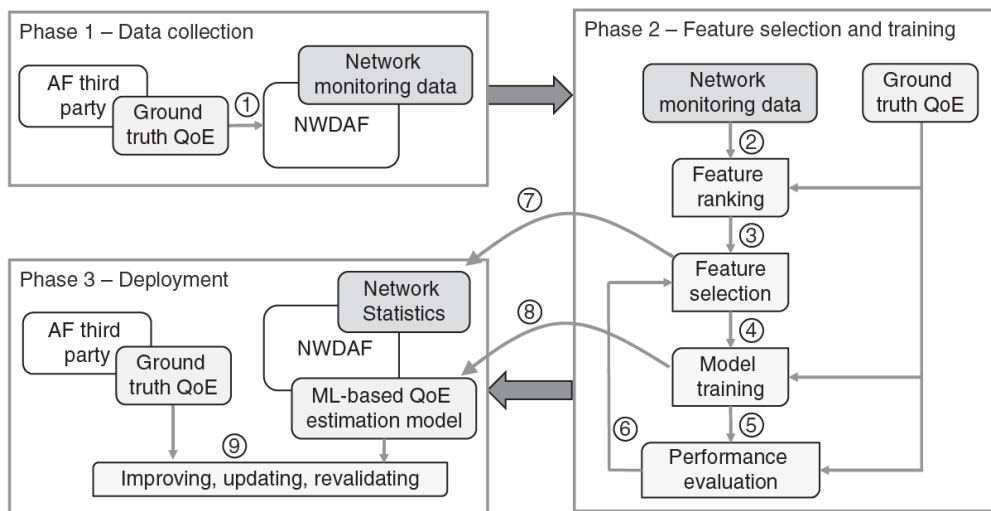


Рисунок 3.2 – Структура для інтеграції машинного навчання в 5G

Виявлений набір функцій визначає необхідну мережну статистику, яку NWDAF має надати 7 для надійного оцінювання QoE на етапі розгортання на основі навченої моделі 8.

Фаза 3 – розгортання: хоча MNO тепер може оцінити QoE без потреби в метриках додатків, наданих AF, CP все ще може передавати таку інформацію для полегшення оновлень, перевірки та вдосконалення навченої моделі 9.

За допомогою цієї процедури ми вирішуємо проблему щодо того, як інтегрувати оцінку QoE на основі ML в архітектуру 5G. Щоб оцінити відповідні функції та врахувати компроміс між точністю та вартістю з точки зору використовуваних функцій, ми застосували підхід ML LASSO до даних, отриманих із моделювання за допомогою симулятора мережі дискретних подій OMNeT++ [41]. LASSO – це метод регресійного аналізу, який виконує вибір ознак і навчає модель передбачати результат на основі вибраних ознак [42]. Його параметр регуляризації дозволяє налаштувати

кількість коефіцієнтів регресії, встановлених на нуль, що зменшує кількість ознак. Для будь-якого розміру набору функцій він вибирає найбільш відповідні функції з усіх доступних. Таким чином, LASSO є відповідним способом дослідження впливу збільшення кількості ознак на точність оцінювання.

Ми дослідили сценарій використання потокового відео та змоделювали клієнтів у мобільній комірці з різними характеристиками руху. Релевантна для мережі інформація відстежується як часові ряди в симуляторі OMNeT++: TP UE, TP базової станції, CQI та час проходження в обидві сторони (RTT). З цих часових рядів ми отримуємо характеристики, застосовуючи такі типові статистичні дані, як середнє значення, стандартне відхилення, мінімум і максимум. Крім того, ми збираємо всі пов'язані з QoE такі показники потокового відео, як зупинки відео та якість, щоб обчислити QoE користувача за допомогою стандартизованої моделі ITU-T P.1203 [43]. Ми застосовуємо LASSO до цих даних і використовуємо різні значення параметра регуляризації, щоб отримати набори ознак різного розміру [36].

Оцінюючи QoE за шкалою MOS на основі 33 різних пов'язаних з мережею функцій, LASSO здатний досягти середньоквадратичної помилки (MSE) приблизно 0,15.

Враховуючи той факт, що MOS коливається від 1 до 5, LASSO може досягти високої точності, незважаючи на те, що це базовий метод регресії. Загалом можна спостерігати тенденцію підвищення точності зі збільшенням розміру набору функцій. Ці висновки можуть бути корисними для оператора мережі, щоб визначити, які точки моніторингу є вирішальними та що має бути пріоритетним у плані оброблення даних. У таблиці 3.2 показано MSE, якого можна досягти, розглядаючи додаткові точки моніторингу для генерування характеристик. Наприклад, ми отримуємо MSE приблизно 0,32, якщо використовуються лише функції, згенеровані з TP низхідної лінії зв'язку (DL) UE. Маючи додаткові функції, які генеруються з CQI висхідної лінії зв'язку (UL), MSE падає нижче 0,25. Якщо також взяти до уваги функції, пов'язані з DL CQI, MSE можна зменшити приблизно до 0,21. Однак можна досягти лише незначного підвищення продуктивності шляхом додаткового задіяння функцій, пов'язаних із TP або RTT вузла доступу (AN).

Таблиця 3.2 – Вплив використання різних типів моніторингу на точність QoE.

Monitoring type	UE DL TP	CQI UL	CQI DL	UE UL TP	AN UL TP	RTT	AN DL TP
MSE	0.318	0.236	0.215	0.193	0.189	0.153	0.151

Ми не тільки вивчали релевантність самої функції, але й додатково оцінювали, чи впливають характеристики руху клієнтів на процес оцінювання QoE.

Дійсно, ми могли помітити, що мобільність має значний вплив на QoE і, як наслідок, на функції, вибрані LASSO. Функції для статичних клієнтів здебільшого генеруються на основі таких статистичних даних, як середнє значення, медіана або різні проценти. Однак для клієнтів, які переміщуються, більшість функцій генерується на основі статистики, яка виражає дисперсію. Наприклад, стандартне відхилення, коваріація або асиметрія. Як наслідок, оператор MNO має проводити моніторинг із більшою деталізацією, якщо він націлений на надійне оцінювання QoE для мобільних клієнтів. В іншому випадку варіації часових рядів, напр. у CQI або DL TP, не може бути зафіксовано достатньо точно. Як наслідок, витрати на оцінювання QoE вищі в мобільних сценаріях.

Парадигми програмного забезпечення, такі як SDN і NFV, надають мережним операторам переваги щодо гнучкості, масштабованості та ефективності витрат. Крім того, вони є ключовими факторами для концепції SFC, яка дозволяє пов'язувати різні NF для формування ланцюжків послуг, а також динамічно змінювати їх структуру та розмір для адаптації до мережних подій. Оскільки сучасні комунікаційні мережі мають підтримувати численні різноманітні служби, які працюють на одному фізичному субстраті та вимоги до яких динамічно змінюються протягом усього терміну служби, ефективне управління та робота цих мереж потребує високого рівня автоматизації в поєднанні з проактивним прийняттям рішень. Це гарантує підтримку ефективності використання ресурсів без впливу на якість обслуговування.

Особливо важливим кроком є визначення оптимальної кількості екземплярів VNF для задоволення поточних і майбутніх запитів на обслуговування. Це безпосередньо впливає на ефективність використання ресурсів і є основою для наступних розв'язань, таких як розміщення та з'єднання VNF. У роботі [23, 44] розроблено схему швидкого та проактивного прийняття розв'язань на основі ML, яка використовує дані моніторингу, щоб передбачити, чи потрібно адаптувати поточну кількість екземплярів VNF. Приймавши це завчасно, запити, що надійшли, можуть бути прийняті безпосередньо після надходження, не чекаючи створення нових VNF. Щоб вирішити проблему відсутності реалістичних наборів даних, ми додатково задіємо методологію генерації позначених навчальних даних, яка об'єднує часову динаміку та різноманітні вимоги реальних мереж. Використовуючи дві різні топології мережі, які презентують сценарій глобальної мережі (WAN) і периферійних обчислень із множинним доступом (MEC), ми демонструємо застосовність запропонованої методології та надаємо розуміння спроможності моделей до узагальнення. Крім того, ми даємо вказівки для мережних операторів щодо

релевантності функцій, кількості навчальних даних і компромісів із точністю, що є результатом довгострокових прогнозів.

Ми використовуємо модель системи, подібну до моделі розміщення VNF [45, 46]. Базова фізична мережа подана неорієнтованим графом, вузли та ребра якого мають такі ресурси, як ядра процесора та пропускна спроможність, і створюють затримки для трафіку, що їх пропускає. Характеристики VNF зберігаються в каталозі, який містить вимоги до ресурсів і потужність обробки пакетів. Запити ланцюга функцій обслуговування характеризуються часом надходження, тривалістю, вузлами, між якими відбувається обмін трафіком, а також обмеженнями щодо мінімальної пропускної спроможності та максимальної затримки відповідного потоку. Кожен запит SFC також містить упорядкований список VNF, які потрібно пройти кожному пакету. Нарешті, ми припускаємо, що система моніторингу постійно збирає різноманітну мережну статистику, наприклад кількість активних запитів у мережі, швидкість надходження запитів SFC, пропускна спроможність, необхідна для кожного запиту, а також кількість активних екземплярів на VNF. типу.

У цій системній моделі завдання прогнозування для дій з розгортання VNF визначається таким чином: у поточний час t_{cur} нам надаються дані моніторингу з вікна шириною a , а також горизонту прогнозування p . Мета завдання передбачення полягає в тому, щоб визначити, потрібно нам збільшити, зменшити чи зберегти поточну кількість VNF, щоб мати можливість врахувати запити, які будуть активними після часу p . Ми розглядаємо цю проблему класифікації, коли вектор ознак витягується з даних моніторингу та зіставляється з рішенням, яке відповідає одній із трьох дій розгортання. Графічне подання процесу передбачення наведено на рисунку 3.3.

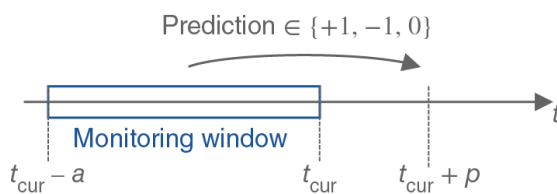


Рисунок 3.3 – Огляд процесу прогнозування

Для виконання окресленого прогнозного завдання потрібно вирішити декілька проблем.

По-перше, бракує реалістичних і загальнодоступних наборів даних, які можна використовувати для навчання та оцінювання алгоритмів. Проблеми, пов'язані з якістю даних, мають високий ступінь однорідності з точки зору VNF і послуг, обмежену кількість часової динаміки та використання синтетичних мережних топологій, які не обов'язково є репрезентативними для мереж реального світу. Крім того, визначення оптимальної конфігурації, а разом з тим і кількості екземплярів VNF у кожен момент часу, є

недетермінованою поліноміальною проблемою (NP) через її зв'язок із проблемою розміщення VNF [45]. Хоча така оптимізація може бути виконана за допомогою рішень на основі ILP, вона дає результати лише для певної конфігурації та не може використовуватися в контексті прогнозів, які містять невизначеності щодо майбутніх подій прибуття та відправлення запитів на обслуговування.

Оператори мереж, які хотіли б застосувати алгоритми прогнозування на основі ML до завдання прогнозування розгортання, також стикаються з операційними питаннями. До них належать: вибір, релевантність і доступність показників, які можна відстежувати за розумну ціну, кількість навчальних даних для навчання моделі ML, і ступінь узагальнення моделі щодо змін у топології мережі та шаблонах трафіку.

Щоб подолати вищезазначені проблеми та дозволити іншим дослідникам відтворити та розширити нашу роботу, ми пропонуємо робочий процес, який охоплює весь процес, зокрема генерацію трасування запитів, генерацію позначених навчальних даних, а також навчання моделі та оцінювання.

1. Загальна конфігурація таких параметрів, як топологія мережі, матриця трафіку, яка визначає шаблони зв'язку між вузлами, і каталог VNF з їх характеристиками. Додаткові параметри для налаштування загального навантаження системи охоплюють максимальну швидкість надходження запитів, а також їх необхідну пропускну спроможність і тривалість.

2. Генерація запитів SFC інтегрує реалістичну часову динаміку, використовуючи дані з матриць реального трафіку для модулювання процесу надходження запиту. Таким чином, як міжденні, так і внутрішньоденні явища щодо кількості активних запитів відображаються в отриманій трасі.

3. Розрахунок оптимального розміщення виконується для конфігурації системи при кожній події прибуття та використовує алгоритм на основі ILP [45]. Зокрема, вихідні дані містять оптимальну кількість екземплярів VNF, які служать слугують основою для навчання моделі.

4. Генерація навчальних даних охоплює побудову позначених векторів ознак. З цією метою різні функції витягуються з вікна моніторингу, яке створюється навколо кожної події надходження, а мітки дії розгортання) визначаються шляхом порівняння виходів ILP для поточного та майбутнього моментів часу.

Навчання та оцінювання моделі виконується шляхом спочатку розбиття набору даних на навчальний і тестовий набори та передачі перших у декілька найсучасніших алгоритмів SL, напр. XGBoost, Gradient Boosting Machine (GBM) і нейронні мережі. На додаток до конкурентоспроможності в різних завданнях SL, алгоритми, засновані на посиленні, дозволяють просто оцінювати важливість функцій, що є важливою можливістю для отримання оперативних вказівок. Нарешті, навчені моделі оцінюються за допомогою тестового набору.

Ми використовуємо запропоновану методологію для оцінювання продуктивності підходів на основі ML для вирішення завдання прогнозування розгортання VNF у контексті двох різних сценаріїв: сценарій WAN охоплює перспективу постачальника послуг, який може створювати екземпляри VNF у різних місцях регіонального або глобального хмарного постачальника, тоді як сценарій MEC передбачає оператора, який може вибрати розміщення VNF в центральних офісах з обмеженою пропускнуою спроможністю, які знаходяться поблизу кінцевих користувачів, або в центральному центрі обробки даних, який має компроміс із затримкою. Далі ми підсумовуємо наші основні результати та переводимо їх у рекомендації для операторів мереж, які планують розгортати такі механізми.

- **Важливість функції.** Ми підкреслюємо, що точні передбачення потребують функцій двох категорій. Перша категорія містить таку інформацію про VNF, як загальна залишена ємність активних екземплярів і їх використання. Друга категорія містить такі контекстуальні глобальні характеристики, як час з моменту надходження останнього запиту або швидкість надходження протягом інтервалу моніторингу. Крім того, кількісно оцінюючи відносний внесок функцій у точність передбачення, оператори можуть приймати більш обґрунтовані рішення щодо того, чи варто інвестувати витрати на моніторинг і зв'язок для збирання відповідних даних.

- **Розмір навчального набору:** надаючи різні обсяги навчальних даних для алгоритмів ML і застосовуючи отримані моделі до тестового набору, ми аналізуємо вплив розміру навчального набору на продуктивність. Крім того, ми виконуємо цей експеримент із процесами надходження запитів, які мають різний ступінь варіативності: один із нормально розподіленим часом між надходженнями та коефіцієнтом варіації, що дорівнює 0,25; і інший, який використовує від'ємний експоненціальний розподіл із коефіцієнтом варіації, що дорівнює 1. Рисунок 3.4 показує числові результати з [44]. Можна спостерігати два ефекти. По-перше, конвергенція є швидшою, і тому потрібно менше даних у випадку менш змінного i , отже, більш передбачуваного процесу надходження.

По-друге, навіть за допомогою всього лише 1000 маркованих прикладів можна досягти рівня точності близько 70%, тоді як точність припиняє суттєве покращення для розмірів навчального набору понад 20 000. Отже, оператори мають знати про особливості трафіку в своїх мережах.

- **Вплив горизонту прогнозування:** хоча довгострокові прогнози залишають більше часу для виконання дій з розгортання та пошуку оптимального місця для розміщення нових VNF, вони також мають компроміси щодо точності. Ми кількісно оцінюємо цей компроміс у дослідженні параметрів щодо горизонту прогнозування p і знаходимо майже лінійне зниження точності прогнозування при зміні p у діапазоні від 10 до 100 секунд. Знову ж таки, параметри моделі ML мають бути узгоджені з характеристиками технічної системи.

- Можливість узагальнення: нарешті, сценарії WAN і MEC разом із різними мережними топологіями служать для аналізу можливості узагальнення моделей, навчаючи їх за одним сценарієм і тестуючи за іншим.

Ми показуємо, що моделі збирають достатньо незалежних від сценарію знань, щоб добре узагальнювати сценарії, але напрямок має значення. Зокрема, особливі властивості сценарію MEC не вивчаються під час навчання на сценарії WAN і, отже, призводять до вищого штрафу за точність. Подібна поведінка очікується при використанні моделей, навчених на основі даних симуляції, щоб робити прогнози у фізичному розгортанні, що вносить додаткову складність.

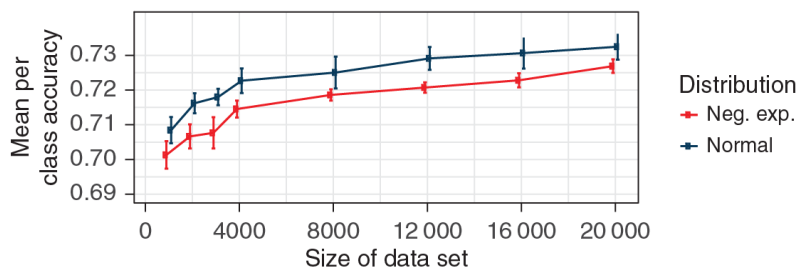


Рисунок 3.4 – Середні значення точності для класу в наборі для тестування при використанні різних обсягів навчальних даних, а також різних розподілів для часу між надходженнями запиту

Мережні служби мають різні вимоги щодо ресурсів, ТР, затримки та тремтіння. Хоча застосований наразі найкращий робочий режим задовольняє базовий доступ до даних, він може не відповідати SLA для послуг із суворими вимогами. Наприклад, потокове відео в реальному часі потребує високого ТР і мінімальної затримки, яка часто недоступна в режимі найкращих зусиль. Одне з розв'язань полягає в тому, щоб зробити більше мережних ресурсів доступними, але це небажано для операторів мереж, оскільки надмірне надання коштує дорого. Натомість оператори мереж прагнуть до розв'язань, які є гарним компромісом між цими двома сценаріями, що означає надання послуг із належним рівнем якості при одночасному зменшенні надмірного надання.

Щоб залишити клієнтів задоволеними та уникнути порушень SLA й надмірного надання, оператори мереж можуть трансформувати свої мережі на основі QoS у мережі на основі QoE, які працюють із користувачами та програмами. Щоб досягти цього, оцінка QoE на основі параметрів QoS стає вирішальною. ML є можливим способом інструментування відображення. Мережі на основі QoE потребують складної структури управління QoE для керування різнорідними та співіснуючими послугами.

У найбільш гнучкому випадку структура обробляє кожен потік пакетів і призначає йому необхідні ресурси. Цей підхід наведено в [7], він демонструє, що управління ресурсами з урахуванням QoE перевершує підходи на основі QoS.

Однак, беручи до уваги зростаючу кількість користувачів і служб із суворими вимогами, цей підхід обробки потоків не можна масштабувати. Новим вирішенням є класифікація послуг у певну кількість класів на основі їхніх вимог.

Наприклад, як запропоновано в [8], таку класифікацію можна зробити таким чином: базова якість (найкраще зусилля), фоновая якість (найменше зусилля), покращена якість і гарантована якість. Інший підхід до класифікації може базуватися на таких типах послуг, як потокове відео, перегляд веб-сторінок, передача голосу через IP, електронна пошта та завантаження файлів.

Класи можна далі диференціювати в різні підкласи якості, наприклад, відео преміум якості та потокове відео базової якості.

Різні мережні послуги призначаються для цих класів на основі їхніх вимог до таких ідентифікованих ключових функцій послуги, як пропускна спроможність, затримка та допуск до фазових флуктуацій. Угоди SLA для кожного класу базуються на вимогах до послуг E2E. Кількість класів і процес кластеризації мають бути ретельно підібрані та розроблені.

Це проблема оптимізації для компромісу між масштабованістю (накладними витратами на площину керування) та підвищенням продуктивності. Наприклад, наявність п'яти класів є масштабованою з точки зору планів управління та контролю, але деталізація диференціації послуг низька, і деякі служби можуть бути недостатньо або надто забезпечені.

Кількість класів, кластеризація та критерії прийнятності для класів є проблемою оптимізації, яка підходить для вирішення на основі ML. Неконтрольоване навчання, а точніше, алгоритм кластеризації k -середніх можна використовувати для кластеризації послуг у визначену кількість k класів. Однак кластери послуг мають бути оцінені з точки зору їх продуктивності та вартості для кожного значення k . Після оцінювання (можливо, за допомогою моделювання або впровадження на фізичному випробувальному стенді) кластер із найкращою продуктивністю та показником вартості вибирається як оптимальний.

Як згадувалося раніше в цьому розділі, існує потреба в складній структурі, що керує класами обслуговування та розподілом ресурсів. Компоненти фреймворку показано на рисунку 3.5. На рисунку показано співіснування різних програм, які спільно використовують ті самі фізичні ресурси. У E2E MANO модельєр QoE приймає дані QoS з мережі як вхідні дані та прогнозує рівень QoE для кожного зрізу. Такий підхід можна використовувати для прогнозування QoE. Після того, як рівень QoE оцінено, картограф/розподільник перевіряє цю інформацію, порівнює її з даними підписки для кожного користувача та виконує такі дії, як перерозподіл фрагментів домену або зміна маршруту, щоб переконатися, що SLA задовольняються. Домен MANO реалізує аналогічну рекурсивну логіку для розподілу фізичних ресурсів між сегментами домену.

На рівні програми кожен клас обслуговування відповідає одному сегменту E2E. Як згадувалося раніше, такі зрізи охоплюють кілька зрізів домену, зокрема доступ, транспорт і ядро. Оператори можуть отримати вигоду від економії масштабу, оскільки співіснуючі програми можуть спільно використовувати ті самі сегменти домену. Однак фрагменти мають надаватися таким чином, щоб не порушити гарантовані SLA для кожної програми.

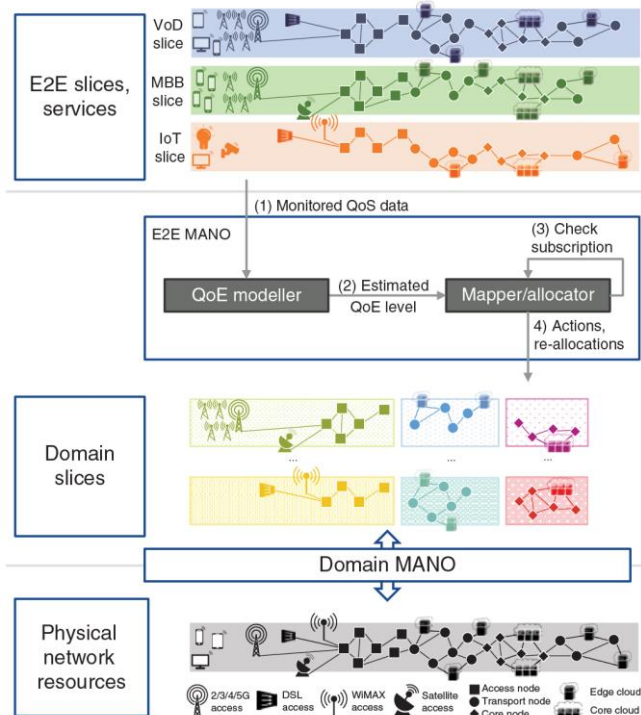


Рисунок 3.5 – Компоненти структури керування QoE

Оптимізація зіставлення між фрагментами домену та сегментами E2E, а також зіставлення між сегментами домену та ресурсами фізичної інфраструктури є проблемою майбутньої роботи.

Оскільки оптимальний розподіл залежить від поточного моніторингу стану мережі, відображення з сегментів E2E на сегменти домену має бути динамічним і в режимі реального часу. Потрібно визначити ключові характеристики мережі, щоб визначити точки запуску, коли поточний розподіл більше не є оптимальним. Прикладами деяких ключових функцій є кількість активних користувачів, використання послань і розмір черги маршрутизатора. Перше завдання – визначити, як часто потрібно контролювати дані. Крім того, потрібно ретельно вибирати порогові значення точки запуску.

Перерозподіл сегментів домену накладає зміни на ресурси, виділені для кожної служби. Робити це занадто часто не бажано, оскільки це може призвести до зміни порядку пакетів, тремтіння та затримки. З іншого боку, нечасте виконання перерозподілу може призвести до непомічення деяких

значних змін стану мережі, що в подальшому призведе до неоптимального рівня QoE і, отже, до незадоволення користувачів.

Кількість користувачів і послуг настільки величезна, що ручне керування та перерозподіл неможливі. В ідеалі потрібен автоматизований агент, який відстежує функції та виконує оптимальну кількість перерозподілів. Агент є частиною QoE Manager (див. рис. 3.5). Оскільки це співвідношення витрат і вигод у майбутньому поточного стану, агент може скористатися перевагами навчання з підкріпленням. На основі набору ключових функцій агент може визначати поточний стан мережі, відстежувати зміни в стані та детально описувати, що було зроблено востаннє, коли система перебувала в такому ж або подібному стані. Він буде навчений у циклі зворотного зв'язку, де дії з позитивною винагородою покращать SLA, а негативні – погіршать або порушать SLA. У цьому контексті дії відносять до перегрупування сегментів домену, виділених сегменту E2E, що є одним класом обслуговування. Проблеми, визначені в цьому розділі, є майбутньою роботою в сфері застосування машинного навчання до автоматизованих мереж на основі QoE та орієнтованих на користувача мереж.

У контексті 5G такі парадигми, як SDN, NFV і нарізка мережі, сприяють створенню складної архітектури, яка спрямована на забезпечення високого рівня програмованості та автоматизації для розміщення широкого спектра послуг, які чутливі до різних характеристик мережі та мають різну часову динаміку. Крім того, оператори хочуть максимізувати ефективність використання ресурсів і обізнаність про послуги, щоб оптимізувати витрати, зберігаючи задоволеність користувачів.

Архітектура складається з кількох рівнів, які містять сутності MANO всередині та між собою. Ці компоненти надають операторам мережі та постачальникам послуг можливість оптимізувати весь стек, починаючи від оцінювання якості послуг через оцінювання QoE і закінчуючи максимізацією ефективності використання ресурсів за допомогою оптимізованого розподілу ресурсів і відображення ресурсів між рівнями.

У той час як високий ступінь складності часто робить традиційні механізми оптимізації незастосовними через швидко зростаючий простір станів, нещодавні досягнення щодо методів AI та ML продемонстрували їхню придатність для вирішення проблем, пов'язаних з мережею. У цьому розділі ми висвітлюємо, як штучний інтелект і машинне навчання можуть допомогти у виконанні вищезгаданих завдань, надаючи інформацію з наших останніх досліджень. По-перше, ми розглядаємо загальний робочий процес і вказуємо на важливі аспекти дизайну та проблеми під час вирішення вищезазначених завдань за допомогою ML. Ці виклики містять отримання реалістичних наборів даних, відповідних системних моделей і необхідного поєднання досвіду, пов'язаного з ML, і спеціалізованого досвіду.

Окрім демонстрації застосовності механізмів на основі ML, ми надаємо розуміння таких операційних аспектів, як важливість функції,

чутливість до різних параметрів і можливість узагальнення. Такі дослідження допомагають операторам оцінювати здійсненність запропонованих підходів у контексті їх конкретної мережі та умов трафіку. Наприклад, ми виявили, що невеликого набору функцій уже достатньо для створення надійних оцінок QoE, що дозволяє операторам мінімізувати накладні витрати на отримання і обробку даних моніторингу. Цей приклад також підкреслює важливу роль, яку відіграють знання домену та ретельна розробка функцій. Що ж стосується розгортання VNF ми демонструємо, що моделі добре узагальнюються в мережних топологіях, тобто їх можна використовувати для прийняття розв'язань у нових середовищах без явного перенавчання.

Нарешті, ми пропонуємо погляд на нові напрямки, які мають високий потенціал щодо підвищення ефективності й способи підходу до них за допомогою методів на основі машинного навчання, які інтегрують і розширюють існуючі механізми. Зокрема, ми зосереджуємося на визначенні відповідних рівнів деталізації для диференціації послуг і розробки стратегій управління зрізами з урахуванням сервісів, спрямованих на оптимальне відображення та спільне використання ресурсів.

3.2 Машинне навчання для розподілу ресурсів у мобільних широкосмугових мережах

Попит на надзвичайно високі швидкості передачі даних, наднадійність і низьку затримку в бездротових мережах зростає через поширення смартфонів, потокового відео, соціальних мереж (наприклад, Facebook), масових комунікацій машинного типу (mMTC), транспортних з'єднань і так далі. До цього часу розробники систем зв'язку поклалися, в основному, на звичайну оптимізацію, не керовану даними, для керування розподілом мережних ресурсів. Однак через складність поширення каналів і блокування у великомасштабних багатодіапазонних бездротових мережах традиційні вирішення оптимізації мережі та керування радіоресурсами (RRM) зазвичай не застосовуються [1]. Причина полягає в тому, що вони не є масштабованими і не враховують унікальні проблеми поширення каналу в різних діапазонах частот [2, 3]. Згодом інтеграція штучного інтелекту (AI) у звичайні алгоритми RRM стає необхідністю.

ШІ можна визначити як дослідження, яке дозволяє машині самостійно ефективно вирішувати проблему. Іншими словами, штучний інтелект можна описати як будь-який механізм, який спостерігає з середовища, що сприймається, і використовує результати спостереження для вирішення конкретної проблеми. Таким чином, штучний інтелект може допомогти створити розумну мережу, яка зможе навчатися, реалізовувати та проактивно розподіляти ресурси. Машинне навчання (ML) є одним із найпотужніших методів штучного інтелекту, які можуть автоматизувати бездротові мережі наступного покоління, забезпечуючи автономне

підключення до різноманітних спектральних діапазонів, таких як міліметрові та терагерцові частоти, автономне регулювання потужностей передачі за допомогою навчання енергії [4, 5], протоколи автономної передачі з навчанням якості обслуговування (QoS) тощо. ML може запропонувати розумне розв'язання шляхом автоматичного вивчення характеристик трафіку даних, керування, засобів контролю та інших характеристик і оволодіння знаннями з експлуатації та обслуговування мережі. Потенціал ML ще більше зростає завдяки масивній інформації та великим наборам даних, які можна зібрати з вузлів зв'язку та точок доступу (AP).

З цією метою в цьому підрозділі наведений вичерпний огляд існуючих методів машинного навчання, які донині застосовувалися до бездротових мереж, а також обговорено їхні переваги, недоліки та сценарії застосування. Потім ми надаємо поглиблений огляд існуючих методів машинного навчання в контексті бездротового каналу та розподілу потужності, планування користувачів і асоціації користувачів. Нарешті, ми перераховуємо ключові показники продуктивності нових бездротових мереж 6G і обговорюємо потенційні методи ML у 6G, такі як навчання передачі даних, імітаційне навчання, федеративне навчання та квантове ML.

У цьому підрозділі обговорюватимуться різні методи ML та їх застосування до проблем бездротової мережі. На рисунку 3.6 показано класифікацію методів машинного навчання.

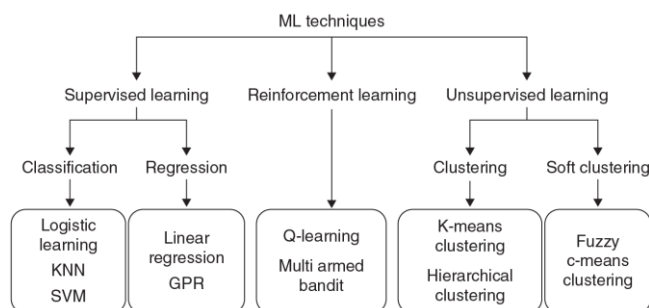


Рисунок 3.6 – Класифікація методів ML

Контрольоване навчання – це процес ML, який вивчає характеристики вихідних даних із вхідних даних, якщо набір даних позначено (тобто вирішення проблеми відомо заздалегідь). Контрольоване навчання передбачає особливості позначеного набору навчальних даних. Його використовували для виявлення помилок, каналного кодування та декодування в бездротових мережах [6]. Контрольоване навчання можна розділити на класифікацію та регресію на основі надійності результату.

Мета алгоритму класифікації полягає в тому, щоб передбачити, у якій мітці групи чи класу ідентифікується об'єкт. Алгоритми класифікації здебільшого дискретні й використовуються для групування даних у

категорії. Загальні алгоритми класифікатора містять логістичну регресію, K-найближчих сусідів (KNN) і машину опорних векторів (SVM).

- Логістична регресія – це метод, який може передбачити ймовірність двійкової класифікації (1/0), і її можна врегулювати, щоб запобігти надмірній «підгонці». Метод можна оновити відповідно до нових даних. Однак він не працює належним чином, коли подається з нелінійними даними. У [7] автори показали застосування логістичної регресії до бездротового зв'язку. Їхня модель складалася з техніки логістичної регресії, яка використовувалася на фізичному рівні для виявлення атак спуфінгу в мережі. Не знаючи моделі каналу, запропонований метод логістичної регресії продемонстрував високу точність виявлення та низьку обчислювальну складність.

- KNN – це техніка ML класифікатора, яка групує дані в кластери. Кластери створюються таким чином, що точки даних у кластері подібні, а коефіцієнт подібності визначається за допомогою евклідових відстаней для пошуку найближчого сусіда. Пропонується зважена модель KNN для прогнозування характеристик мережного трафіку та класифікації різних часових і просторових моделей запасів радіоресурсу [8]. Ця модель KNN може фіксувати динаміку навантажень трафіку, щоб забезпечити балансування навантаження та підвищення ефективності використання радіоресурсів.

- SVM може класифікувати дані, знаходячи лінійну межу, яка розділяє точки даних від одного класу до іншого. SVM підходить для нелінійно розділених даних; він простий для інтерпретації та має високу точність. Тим не менш, SVM потребує тривалого часу обробки та часу навчання. SVM використовується для класифікації базових станцій відповідно до їх трафікового навантаження [9]. Для класифікації базових станцій автори використовували три різні класи: клас-1 (завжди завантажений), клас-2 (ранковий пік завантаження) і клас-3 (вечірній пік завантаження). Записи даних викликів (CDR), які містять таку корисну інформацію, як ідентифікатор користувача, ідентифікатор стільника та географічне положення веж стільникового зв'язку, до яких був підключений користувач, використовувалися як навчальні дані для SVM з точністю передбачення 87,14%.

Алгоритми регресії можуть адаптувати математичні моделі для опису набору даних для апроксимації або інтерполяції. Загальні алгоритми регресії містять лінійну регресію та регресію процесу Гаусса (GPR). Ці методи регресії можуть оцінювати зв'язки між змінними і є потужними статистичними інструментами для передбачення та прогнозування неперервної змінної [10]. Регресійна модель була використана для побудови зв'язку між співвідношенням сигнал/завада плюс шум (SINR) і коефіцієнтом прийому пакетів [11]. За допомогою цієї моделі було створено аналітичну структуру для відображення компромісів між накладними витратами та точністю вимірювань завад.

- Лінійна регресія – це техніка статистичного моделювання, яка використовується для опису неперервної відповіді як функції однієї чи кількох змінних предиктора. Ця техніка проста для інтерпретації, але може не вловлювати складні моделі.

- Георадарні моделі можуть передбачити значення неперервної змінної відгуку, і воно є непараметричним. Цей алгоритм є точним для інтерполяції просторових даних за наявності невизначеності.

Неконтрольований ML – це процес, який досліджує приховану структуру немаркованих даних. Він не має контролера чи супервізора, які мають позначені дані, тому машина навчається сама на основі немаркованих вхідних даних. Крім того, неконтрольоване навчання використовує кластерний аналіз наборів даних, у якому дані розділені на категорії на основі подібних характеристик. Однак під час використання неконтрольованого навчання виникає багато проблем, оскільки дані, надані машині, не містять жодних міток; в результаті реалізатор може не знати, яким має бути правильний результат. Тому оцінювання алгоритмів неконтрольованого навчання потрібно аналізувати вручну. Існує два типи кластерних алгоритмів: жорстка кластеризація та м'яка кластеризація.

Алгоритми кластеризації групують точки даних в один кластер таким чином, щоб кластери не перетиналися. Деякі популярні алгоритми кластеризації містять К-середні та ієрархічну кластеризацію.

- К – означає, що кластеризація поділяє дані на К розділів взаємовиключних кластерів, звідки й назва. Кластеризація K-means – це простий алгоритм, який кластеризує точки даних таким чином, що евклідова відстань від однієї точки даних до іншої мінімізується. Алгоритм продовжує цей процес, щоб знайти центроїд, який є точкою даних, що має мінімальну відстань до сусідніх точок даних для створення кластерів. У [12] учасники запропонували алгоритм К-середніх для проблеми кластеризації гарячих точок для максимального використання спектра. Алгоритм K-means запропонував кластеризувати користувачів на основі розташування та відстані. Користувач із мінімальною віддаленістю від центроїда вибирається як гаряча точка для кластера.

- Ієрархічна кластеризація групує дані у бінарне ієрархічне дерево, де кластери формуються для визначення подібності між точками даних. Цей алгоритм забезпечує керівництво візуалізації для вибору кластера. Однак він має кубічну складність часу, через що працює повільніше, ніж інші алгоритми кластеризації. У [13] дослідники запропонували комбінацію ієрархічної кластеризації та кластеризації К-середніх для аналізу активності користувача та виявлення аномалій користувача. Вони змогли точно перевірити справжню особу користувачів за допомогою динамічної просторово-часової діяльності.

М'яке кластеризування без нагляду дозволяє точкам даних збігатися в різних кластерах/групах. Найпоширенішим алгоритмом м'якої кластеризації є нечіткі с-середні. Нечіткі с-середні дані розбивають на

кластери, де точки даних можуть належати до одного або кількох кластерів. Точки даних, розташовані ближче до центроїда, мають високе значення членства в цьому кластері, а дальні точки даних мають нижчий ступінь.

Нечіткі с-середні корисні для розпізнавання образів і для точок даних, які мають більше однієї характеристики. Для досягнення енергоефективності в бездротових сенсорних мережах запропоновано нечіткий алгоритм с-середніх [14]. Нечіткі с-середні використовуються для формування оптимальної кількості кластерів. Потім ці кластери використовуються для вибору голів кластера та усунення надлишкових генерації та передачі даних, щоб уникнути втрати енергії шляхом відключення непотрібних датчиків.

Навчання з підкріпленням (RL) є найпопулярнішим методом у контексті розподілу ресурсів у бездротових мережах. Навчання з підкріпленням спирається на ітераційне навчання та процес прийняття рішень. Ця методика була застосована для оптимізації охоплення та потужності й розподілу ресурсів [15]. Різниця між RL і навчанням під наглядом/неконтролем полягає в тому, що RL ґрунтується на системі зворотного зв'язку, де користь отримується, якщо все зроблено правильно, в іншому випадку дається попередження. Таким чином, метод RL постійно розвиватиметься, тоді як контрольоване/неконтрольоване навчання зазвичай має статичні розв'язання. Алгоритми навчання з підкріпленням зазвичай бувають двох типів: Q-навчання та методи багаторукового бандита.

- Q-навчання – це використовувана широко методика підкріплення, яка взаємодіє з середовищем/проблемою для вивчення значень «Q». Значення Q вважаються винагородою, яка є кортежами, що складаються зі стану та дії, виконаної машиною. Після того, як машина навчена, вона має можливість обчислювати дії, які максимізують значення Q, оскільки винагороду потрібно максимізувати. Q-навчання використовується для автономного розподілу ресурсів каналу [16]. Система Q-навчання складається з трьох основних компонентів: винагороди R, стану каналу S і дії каналу A. A вибирається так, щоб для розподілу вибиралися вільні канали. Однак усі дії мають бути вичерпані та оновлені. S вказує стан каналу, який також містить інформацію про якість і використання каналу, а також час простою кожного разу перед тим, як дія вибере канал. Потім машина навчається з цими обмеженнями для створення таблиці з діями, що максимізують винагороду. Недоліки цього методу охоплюють масштабованість у великі мережі з більшою кількістю каналів і часову складність виконання всіх дій.

- Використовується Q-навчання без збереження стану, інша програма Q-навчання, де автори застосовують варіацію Q-навчання без збереження стану для покращення просторового повторного використання в бездротовій мережі [17]. Автори зосередилися на дозволі мережі змінювати як потужність передачі, так і ресурси каналу на основі досвідченої пропускної спроможності. Q-навчання без збереження стану застосовується до децентралізованого сценарію, де доступна вся інформація про

бездротовий вузол (WN), і кожна WN вважається агентом, який виконує алгоритм Q-навчання без збереження стану. Використовується стратегія жадібного вибору дій, щоб дії відповідали всім комбінаціям, які можна вибрати для каналу та потужності передачі [17].

Цей метод Q-навчання дозволяє підвищити продуктивність за допомогою дій, що максимізують винагороду. Алгоритм здатний використовувати оптимальний канал і розподіл потужності.

- Багаторукий бандит стає на шлях дослідження та експлуатації. Цей алгоритм використовується, коли є кілька шляхів для вибору, і за кожен вибір надається винагорода. Подібно до Q-навчання, пріоритетом є максимізація винагороди, але існують компроміси між поточним шляхом і збиранням інформації для отримання більшої винагороди. Багаторукий бандит застосовувався для динамічного вибору каналів у чотириканальних бездротових локальних мережах (WLAN) [18].

Враховується якість зв'язку, наприклад пропускна спроможність; значення винагороди та багаторукий бандит використовуються для максимізації пропускної спроможності шляхом використання вжитих дій або дослідження нових. Однак поглиблене дослідження може призвести до втрат у максимізації винагороди, тоді як швидкі рішення про експлуатаційні дії можуть призвести до втрати найкращого вибору. Тому автори запропонували рішення надшвидкого багаторукого бандита, яке використовує хаотично коливальні форми сигналів для досягнення затримки 1 нс. Ця система була реалізована в чотириканальній WLAN, яка успішно продемонструвала адаптивний і автономний вибори каналу для підвищення пропускної спроможності.

Алгоритми глибокого навчання (DL) засновані на штучних нейронних мережах (ANN), які здатні моделювати та обробляти нелінійні зв'язки. DL залежить від великих обсягів даних і можливостей обробки, що, у свою чергу, забезпечує багаторівневі моделі, які вивчають ефективно подання даних із неструктурованих джерел. Крім того, алгоритми DL можуть обробляти великі обсяги даних, що може призвести до кращої точності передбачення. DL здатний вивчати функції на різних рівнях і дозволяти цільовій системі вивчати складні функції. Однак найбільшим недоліком DL є те, що він потребує великих обсягів даних для навчання та тестування, які можуть бути недоступними та важко згенерованими (особливо, коли дані стосуються «міток» під час навчання під керівництвом).

DL можна використовувати для пошуку оптимальних рішень для проблем розподілу бездротових ресурсів [19]. DL було поєднано з результатами методики навчання під наглядом для навчання глибокої нейронної мережі (DNN). Показано, що це створює оптимальні рішення з низькою обчислювальною складністю [20]. Автори використовували цю форму мережі DL (DNN) для вирішення проблеми розподілу потужності в каналі завад, причому продуктивність була дуже близькою до зваженої мінімальної середньоквадратичної помилки (WMMSE).

Інше застосування DL для бездротового розподілу ресурсів представлено в [21].

DL поєднується з підкріпленням навчанням для створення алгоритму розподілу спектра на основі глибокого Q-навчання (DQL), який застосовується до бездротової мережі IoT, що складається з основних і вторинних користувачів. Основним користувачем було надано фіксований контроль потужності, тоді як вторинні користувачі могли регулювати свою потужність, автономно вивчаючи спільний загальний спектр за допомогою запропонованого алгоритму DQL.

Короткий опис методів ML наведено в таблиці 3.3. Оскільки точність навчання значною мірою залежить від подання навчальних даних, певні методи навчання під наглядом можуть потребувати інтенсивних обчислень і не підлягати масштабуванню. Крім того, якість результатів навчання під контролем обмежена якістю етикеток. Підсилене Q-навчання є хорошим вибором для оптимізації розподілу ресурсів за допомогою методології дії, стану та винагороди. Однак навчання з підкріпленням стає обчислювально інтенсивним зі збільшенням кількості дій і станів. Таким чином, поєднання контрольованого навчання та інших методів ML (RL або DL) було б корисним, наприклад, алгоритм, запропонований у [21] для стільникових мереж IoT. Неконтрольований ML зазвичай корисний для класифікації вхідних і вихідних даних у різні групи на основі доставки даних і використовується для керування ресурсами, розподілу ресурсів і розумного кешування в бездротовому зв'язку [22].

Таблиця 3.3 – Опис методів ML

Objective	Methodology	Main conclusion	Scenario
Improve the maximization of spectrum utilization	K-Means clustering	K-Means proposed to cluster users based on locations and distances Provided optimal centroid selection for possible hot spot	Cellular network
Building relationship between SINR and packet reception ratio	Regression	Analytical framework created to display trade-offs between overhead and accuracy of interference measurements	Sensor networks
Predict characteristics of network traffic while categorizing different temporal and spatial patterns of radio resource margins	KNN	Proposed a weighted KNN that can capture the dynamics of traffic loads which enables load balancing	Wireless network
Improve spatial reuse in wireless networks using stateless Q-learning	Stateless Q-learning	The method addressed performance enhancements by utilizing reward maximizing actions. Method exploits optimal channel and power allocations	Wireless network
To dynamically select channels within a WLAN with low latency	Multi-armed bandit	ML technique achieved latency of 1 ns and adaptive channel selection to improve throughput	WLAN

Динамічний характер бездротових мереж потребує регулярного повторного виконання алгоритмів розподілу ресурсів. Однак якість алгоритмів RRM може погіршитися, коли вони розгортаються в реальних ситуаціях, оскільки вони зазвичай базуються на математичних моделях, а не на даних реального світу. Традиційне керування ресурсами можна покращити шляхом накопичення цінних знань, пов'язаних із користувачами та мережею, і внесення їх у алгоритми ML. У цьому підрозділі ми надаємо

огляд алгоритмів ML, які використовуються для керування потужністю, планування користувачів, асоціації користувачів і розподілу спектра.

Керування потужністю в бездротових мережах означає регулювання потужності передавача для досягнення покращеної передачі сигналу або мінімізації завад. Ефективне керування живленням може зменшити завади між користувачами по всій смузі пропускання, підвищити надійність мережі, зберігаючи обмеження QoS у налаштуваннях реального часу, збільшити пропускну спроможність стільникових мереж і збільшити пропускну спроможність пристрою. Контроль потужності є важливим у стільниковій мережі для керування завадами та енергією, оскільки ємність батареї пристроїв обмежена. Крім того, контроль живлення для керування підключенням через варіації бездротових каналів. Приймач має мати можливість підтримувати постійний рівень переданого сигналу, щоб він міг залишатися підключеним до передавача, та передбачати стан каналу. Таким чином, керування потужністю дозволяє зберегти логічну сумісність для передачі конкретного сигналу [33, 34].

Глибоке навчання з підсиленням використовувалося в бездротових мережах для керування потужністю завдяки його надійним функціям, ефективності та достатньому часу обробки. Автори в [23] розробили некооперативну гру розподілу потужності в комунікації між пристроями (D2D). Вони навели три окремих алгоритми навчання з глибоким підкріпленням. Ці алгоритми склалися з DQL, дуельного DQL і подвійного DQL для багатоагентного навчання. Інший підхід глибокого підкріплення до навчання був запропонований для вирішення динамічної проблеми розподілу ресурсів у рамках неортогонального доступу з кількома несучими [24].

Запропоновано безмодельний алгоритм розподілу потужності [35]. Була створена глибока система розподілу потужності на основі RL, яка може контролювати розвиток мереж і працювати з мінімальним розповсюдженням даних розподіленням способом. Підхід передбачав центральну систему навчання, де кожен передавач функціонує як навчальний агент. При цьому досліджувалось невідоме середовище з жадібною стратегією. Потім результат передається до центру керування для оптимізації функції зваженої сумарної ефективності, яку можна було визначити для досягнення повної сумарної швидкості. У [25] дослідники описали метод вирішення проблеми розподілу ресурсів у гетерогенних бездротових мережах. Він керує багатоагентною мережною інфраструктурою, щоб максимізувати сумарну пропускну спроможність мережі, таким чином зменшуючи складність і забезпечуючи QoS для користувачів. Техніка ML у поєднанні з алгоритмами SVM і глибокої нейронної мережі була реалізована для спільного керування потужністю та планування передачі в стільникових мережах [26].

Таблиця 3.4 підсумовує алгоритми керування потужністю на основі методів ML. Загалом, DQL визнано ефективною технікою ML, що

застосовується в сценаріях керування потужністю [23]. Тут Q -значення можуть бути виражені у форматі таблиці або мережею нейронів із певними накладними витратами на пам'ять і обчислення. Використання глибокого навчання з підкріпленням для оцінювання звичайних алгоритмів потужності високої складності є корисним підходом для виконання розподілу потужності в реальному часі та вирішення проблем безперервної дії. Нарешті, мультиагентні методи навчання для некооперативних проблем і передові методи DL є перспективними підходами для підвищення точності навчання, зменшення пробної помилки та зменшення складності алгоритму.

Планування є критично важливим аспектом бездротових мереж, який дає змогу вирішувати, які користувачі збираються передавати, використовуючи певний час/частоту/ресурс антени.

Планування радіоресурсів – це механізм, за допомогою якого кадри ресурсів виділяються користувачам. Численні дослідження алгоритмів бездротового планування були опубліковані в літературі. Крім того, враховуючи великі вимоги до підключення, бездротові мережі мають підтримувати не лише окремих користувачів, але й, можливо, велику кількість недорогих машин і датчиків. Ці машини мали б функції трафіку, відмінні від звичайних людських програм. Наприклад, датчик може прокинутися, передати свої вимірювання через бездротову мережу та повернутися в режим сну. У цьому контексті методи ШІ мають потенціал для досягнення кращої ефективності порівняно з традиційними методами.

У [36] автори навели опис планувальника на основі градієнта керування для автоматизації потоку трафіку до стільникової мережі. Вони явно розглядають проблему планування як процес прийняття рішень за Марковим (MDP) і вказують, що процес динамічно реагуватиме на різницю в трафіку, дозволяючи мобільним мережам приймати більше трафіку даних, що на 14,7% більше порівняно з існуючими планувальниками, перевершуючи при цьому на коефіцієнт 2. Так само дослідження, описане в [37], використовувало глибоке навчання Q для планування придорожньої мережі. Відносини між популяціями транспортних засобів структуровані як MDP, охоплюючи послідовність дій, знахідок і сигналів винагороди.

Апроксимуючи функцію значення Q , агент виявляє стратегію планування, яка приводить до меншої затримки порівняно зі звичайними підходами до планування.

У [38] учасники одночасно розглядали питання планування користувача та кешування вмісту. Зокрема, вони навчають агента глибокого підсилювального навчання (DRL), щоб визначати, яка базова станція обслуговуватиме такий вміст і чи потрібно вміст кешувати. Крім того, спостерігач призначається для вимірювання функції вартості та надання виконавцю вхідних даних. Моделювання на кластері базових станцій демонструє, що агент може забезпечити низьку затримку передачі. У [26]

дослідники вивчали планування зв'язку та керування живленням у бездротових мережах для оптимізації продуктивності. Вони розклали початкову проблему на лінійну та нелінійну системи та навели алгоритм ML у поєднанні з SVM та глибокою нейронною мережею для вирішення нелінійної підпроблеми.

Таблиця 3.4 – Керування потужністю на основі машинного навчання

Objective	Methodology	Main conclusion	Scenario
Improve network efficiency while maintaining the QoS limitations in real-time environments	DRL	Three separate deep reinforcement learning algorithms was introduced to maximize energy-efficient resource allocation	D2D
Optimizing the efficiency of the non-orthogonal multiple access multi-carrier framework	DRL	A deep reinforcement learning system with optimum power weights using an attention-based neural network is considered to resolve the channel assignment problem	non-orthogonal multiple access (NOMA) system
To optimize the weighted efficiency sum-rate function that could be defined to achieve the full sum-rate	Multi-agent DL	Proposed a model-free interference power allocation algorithm that achieves comparable efficiency by implementing enhanced learning techniques to different complex wireless networks	Wireless network
Manage the multi-agent network infrastructure and the total difficulty of the distribution of resources	Q-Learning	The method addressed the optimization problem in dense heterogeneous wireless networks as a distributed solution, thus reducing complexity	HetNets
Performance optimization power management in cellular networks	SVM	ML technique combined with both SVM and deep belief neural network algorithms to address a nonlinear problem	Wireless network
Femtocell trains itself to change its transmitting capacity to support its consumer service while preserving the macrocell consumer	Q-Learning	Solve the issue of power optimization in compact heterogeneous networks thus substantially reducing resource consumption	HetNets

Метод глибокого навчання Q для мінімізації споживання енергії в мережах реального часу був наведений у [28]. У їхній пропозиції для наближення Q-функції використовується автоматичний кодер (AE), а система проводить повторне відтворення для оптимізації циклу тестування та прискорення конвергенції. Було реалізовано два алгоритми на основі нейронної мережі, щоб довести, що вільні слоти в мережі множинного доступу з частотним поділом часу (MF-TDMA) можна точно передбачити за допомогою підходу наближення [30]. Алгоритми планування на основі ML наведені в таблиці 3.5.

Методи DL, такі як DQL, корисні для вирішення проблем планування. DQL можна використовувати для планування масштабування частоти [28].

Нещодавно ML було впроваджено для планування користувачів і показало значну ефективність порівняно з традиційними методами. Спільне планування, яке передає інформацію для автоматизованого планування користувача, також корисне для усунення завод. Тим не менш, деякі

неідеальні з'єднання можуть зіткнутися з дуже довгою затримкою, що є проблемою для скоординованого планування у великих мережах. Крім того, мобільність може створювати помилки передбачення каналу, що є ще однією проблемою для створення надійних алгоритмів планування.

Асоціація користувачів, тобто асоціація користувача з конкретною базовою станцією, що його обслуговує, впливає на ефективність мережі. Стільникові мережі стають неоднорідними в основному через додавання малих комірок. Через різну потужність передачі різних базових станцій вимірювання асоціації користувачів, наприклад SINR, можуть виникнути значні розбіжності у навантаженні та недостатньому використанні малих комірок [3].

Таблиця 3.5 – Планування на основі машинного навчання.

Objective	Methodology	Main conclusion	Scenario
Enables cellular networks to accommodate more data traffic, although outperforming better	DRL	Develop a regulation gradient-based scheduler to maximize traffic flow to the mobile network. They placed the scheduling concern for forecasting system performance	HetNets
Reduces the energy consumption	Q-Learning	Discovers a scheduling strategy that results in reduced latency and busy power consumption relative to conventional scheduling approaches	V2V
The operator can produce low delay in transmission	DRL	The RL method is introduced to improve the predictive policy with a view to reducing the average latency in transmission	Cellular networks
Minimizing ultra-reliable time of low latency contact	RL	A machine-based learning algorithm introduced combined with SVM and deep-belief network approaches to tackle the nonlinear sub-problem	Wireless networks
Reduce the use of power in real time networks	Deep Q-learning	Adopt a deep Q learning scheduling method for frequency scaling, to reduce power efficiency in real time applications	Real time systems
Reduce the overall delay costs and energy usage for all user	Deep Q-learning	Uses deep Q learning to simultaneously automate offloading decisions and the utilization of computing resources	Mobile edge computing
The number of collisions with other networks are reduced by half	Imitation learning	Presenting two neural network-based algorithms to show that an approximation method can reliably predict free slots in the MF-TDMA network	HetNets

Ключовим відсутнім компонентом у поточних показниках асоціації є навантаження, яке дає бачення розподілу ресурсів, таким чином впливаючи на довгострокові рівні. Загалом, через невідомі двійкові змінні ідентифікація відповідного рішення асоціації користувачів, з урахуванням навантаження, є проблемою, пов'язаною з експоненціальною складністю нелінійної оптимізації. Таким чином, ШІ може служити практичним і менш складним методом для отримання оптимальних глобальних рішень. Алгоритми асоціації користувачів на основі ML підсумовано в таблиці 3.6.

Таблиця 3.6 – Асоціації користувачів на основі машинного навчання

Objective	Methodology	Main conclusion	Scenario
Maximize the long-term downlink potential while maintaining the QoS specifications of the UEs	Multi-agent Q-learning	The multi-agent distributed DRL was proposed to achieve a mutually optimized user association for the HetNets	HetNets
To support multi-connectivity in millimetre wave (mmWave) networks	Multi-label classification	Proposed a machine-based computing method to address user-association problems in mmWave communication networks	mmWave networks
Identify user associations which can reduce the occurrence of breaks in presence	Federated learning	Develop federated learning algorithm based on echo state networks to identify the association of users	Wireless virtual reality networks
Conduct load balancing between the base stations	RL	Online RL algorithm considered to deliver reasonable service level for vehicles	Vehicular networks
Enhance association between users and base stations taking into account various factors that influence QoS	Collaborative filtering	Collaborative filtering based network was adopted that indeed balance the traffic load	HetNets

Алгоритм на основі RL був наведений для низхідної лінії зв'язку гетерогенних стільникових мереж для досягнення оптимальної довгострокової загальної ефективності мережі при одночасному забезпеченні споживачів специфікаціями QoS [31]. При цьому розробляється метод розподіленої оптимізації на основі багатоагентного RL для спільної оптимізації асоціації користувачів і розподілу ресурсів. Існують численні рішення проблеми асоціації користувачів у мережах міліметрового діапазону для оптимізації використання стільникових послуг [42–45].

У [39] автори описали розроблену нову техніку ML, орієнтовану на користувача, щоб увімкнути багатозв'язність у мережах mmwave, де завдання асоціації користувачів було визначено як завдання класифікації з кількома мітками. Шляхом реалізації ефективних алгоритмів для різних класифікаторів вихідна задача трансформувалася в набір задач класифікації з однією міткою. Використання просторово-часового потоку трафіку в транспортних мережах допомагає створити структуру онлайн-асоціацій, яка фіксує зміни навантаження різних комірок [40].

Традиційні методи розв'язання проблеми асоціації користувачів містять: (1) максимальний SINR, який дозволяє користувачеві вибрати базову станцію, яка максимізує його SINR, і (2) градієнтний спад і методи подвійного розкладання. Лі та ін. [40] показує, що навчання з підкріпленням забезпечує рішення, які можуть більш адаптивно реагувати на динаміку бездротових мереж порівняно з традиційними рішеннями.

Багатообіцяючим рішенням є перетворення проблеми асоціації користувачів у послідовну проблему прийняття рішень, яку можна вирішити за допомогою навчання послідовності. Іншим рішенням є застосування стратегії RL, яка усуне залежність від навчальних зразків, отже, заощаджуючи час і ресурси. Алгоритми неконтрольованого навчання

для асоціації користувачів також передбачають отримання швидкого результату.

Розподіл спектра – це процес розподілу ресурсів між клієнтами таким чином, щоб досягти певної мети (наприклад, загальної швидкості передачі даних). Цей процес виконується в мобільних мережах шляхом збирання інформації про стан спектра користувачів і визначення оптимальної відповіді. Вплив спектра буде надзвичайно важливою проблемою для майбутніх бездротових мереж. Модельно-залежні підходи не можуть точно адаптувати реальні системи. Таким чином, підходи до динамічного доступу до спектра на основі ML можна застосовувати централізовано для підтримки великої кількості користувачів, які споживають спектр [46]. Розподіл спектра на основі ML відображено в таблиці 3.7.

Таблиця 3.7 – Розподіл спектра на основі машинного навчання

Objective	Methodology	Main conclusion	Scenario
Helps to ensure economic and social justice in CRAVENET	RL	Methodology for controlling spectrum was built for QoS in CRAVENET system with small average latency	CRAVENET
DRL is used to learn about the availability of spectrum resources for test the performance	DRL	Deep Q-network approach is shown to converge faster with better performance comparing Q-learning	Cognitive radio networks
Maximize secondary network efficiency when fulfilling the main link interruption limit	Deep Q-networks	Using deep Q-network resulted in greater integration and improved the adaptability of cognitive radio networks	Cognitive radio networks
Estimate spectrum allocation with low signal latency using DRL	DRL	Using unlicensed spectrum for small long term evolution (LTE) cells thus maintaining compatibility for existing Wi-Fi networks and other LTE providers	HetNets
Boost the number of V2I connections and the packet transmission rate for V2V connections	Multi-agent RL	Enhanced spectrum utilization by upgrading Q-networks using the feedback obtained	Vehicular networks

У [7] автори дослідження навели централізований, самоорганізований підхід до ефективного контролю спектра в системі Cognitive Radio Vehicular Adhoc Network (CRAVENET). Метод складається з п'яти масштабів вимог користувача до підключення. Кожен користувач надає інформацію для прийняття рішень самоорганізовувальному директору. Потім структура автоматично організовує розподіл спектра відповідно до потреб клієнта, використовуючи методику навчання з підкріпленням. Це допомагає налагодити ефективне самоорганізоване керування спектром разом із покращеним QoS у системі CRAVENET для значно меншої затримки. В іншому дослідженні розглянуто складну проблему спектрального контролю в когнітивних радіомережах з використанням методів на основі DRL [48]. Автори запропонували підхід DRL та обчислення резервуарів для спільного доступу до спектра. Додатковий користувач здатний відстежувати всі потоки в режимі реального часу та враховувати збій у розпізнаванні та конфліктну взаємодію з іншими додатковими користувачами. Вони показали, що DRL може забезпечити стан каналу та надійні прогнози для складного розподілу спектра.

Глибокі Q-мережі подані для оптимізації продуктивності вторинної мережі [9]. Крім того, було показано, що використання глибокої Q-мережі забезпечує швидше розгортання та більшу адаптивність до складних архітектур бездротових мереж.

У [5] дослідники описали реалізацію проактивного алгоритму розподілу ресурсів.

Вони застосували комірки довготривалої короткочасної пам'яті на рівні безпілотних літальних апаратів, дозволяючи ефективно розподіляти радіоресурси та дозволяючи невеликим бездротовим мережам швидко оцінювати канал та граничний розподіл спектра.

Оскільки базовим станціям дозволено контролювати ресурси спектра, ML може бути життєздатним методом для базових станцій для ефективного розподілу ресурсів спектра. Крім того, такі алгоритми, як Q-навчання, можуть бути можливим кандидатом, оскільки вони спроможні навчатися в своєму середовищі. Це дозволить базовим станціям прогнозувати надійний розподіл ресурсів спектра. Спільне використання спектра є важливою технікою для збільшення спектра ресурсів у майбутніх бездротових мережах [2].

У цьому підрозділі наведено вичерпний огляд існуючих методів ML, які мають вирішальне значення для розподілу бездротових ресурсів, особливо в контексті бездротового каналу та розподілу потужності, планування користувачів і асоціації користувачів. Ми висвітлюємо сценарії застосування, де можна застосувати різні методи ML. Ми робимо висновок, що алгоритми на основі ML дозволять розв'язувати проблеми розподілу жорстких ресурсів NP масштабним і економічно ефективним способом із низькою обчислювальною складністю. Спільне використання спектра на основі блокової мережі є цікавим для підвищення ефективності та безпеки стандартного спільного використання спектра. Ще один багатообіцяючий підхід – це квантовий зв'язок, який може підвищити ефективність і безпеку обчислень. Значна ефективність навчання та обробки знадобиться для забезпечення співпраці між високоінтегрованими мережами 6G із периферійними обчисленнями з множинним доступом, автомобільним зв'язком, масивним Інтернетом речей, безпілотними літальними апаратами, інтегрованими повітряно-наземними та супутниковими мережами, бездротовими датчиками тіла в тілі людини.

Методи ML можуть повністю автоматизувати архітектуру фізичного рівня, прийняття рішень, керування ресурсами та використання ресурсів у 6G. Передові методи машинного навчання, такі як навчання за допомогою трансферу, імітаційне навчання, федеративне навчання по краях і квантове навчання, стануть ключовими моментами для розподілу ресурсів у 6G.

Нижче ми обговорюємо такі потенційні методи ML для 6G, як навчання з перенесенням, імітаційне навчання, навчання з об'єднаними краями та квантове ML.

Розподіл ресурсів має вирішальне значення в бездротових мережах для оптимізації продуктивності.

Тим не менш, через складність цих проблем оптимізації вони, як правило, стають проблемами змішаного цілочислового нелінійного програмування (MINLP), особливо розподілу спектра, розподілу підканалів, асоціації користувачів і проблем планування. Трансферне навчання за допомогою самоімітації може бути застосоване до цих проблем, що значно покращує продуктивність, але страждає від проблем неузгодженості завдань, які виникають при зміні параметрів мережі [3]. Трансферне навчання – це техніка ML, у якій модель, створена на основі попереднього завдання, повторно використовується як відправна точка для моделі для інших завдань. Тому застосування трансферного навчання усуває завдання створення моделі з нуля. Загалом, це унікальний метод оптимізації розподілу ресурсів, оскільки він здатний адаптуватися до різних параметрів мережі за допомогою мінімальної кількості навчальних наборів, тим самим скорочуючи час навчання.

Розподіл ресурсів у бездротових мережах зазвичай формується за допомогою MINLP, а розгалуження та зв'язок (BB) – це загальноживаний алгоритм для вирішення таких проблем розподілу ресурсів. Однак він збігається повільно і має дуже високі часові складності. Використання імітаційного навчання покращує продуктивність BB за допомогою політики скорочення, яка автономно відкидає неоптимальні вузли [4]. Основна мета алгоритму – знайти оптимальне розв'язання шляхом порівняння всіх рішень між собою. Цей метод імітаційного навчання забезпечує хорошу оптимальність і скорочує час обчислень, але він покладається на результат алгоритму BB, який може бути як добрим, так і поганим. Крім того, запропонований алгоритм імітаційного навчання прискорює алгоритм BB, але інколи швидкість прискорення недостатньо висока, щоб побачити значні зміни [4].

Методи імітаційного навчання мають на меті копіювати людську поведінку в заданому завданні.

Машина (агент) навчена виконувати завдання з попередніх демонстрацій, вивчаючи відображення між спостереженнями та діями. Імітаційне навчання також є поєднанням навчання під наглядом і підкріпленого навчання без явних ярликів. Проте наводяться приклади того, як дістатися до об'єкта, якщо це необхідно. Імітаційне навчання більше підходить для таких програм, як підключені та автономні транспортні засоби.

Інтегроване навчання – це розподілена техніка ML, яка дозволяє навчатися на великому обсязі даних, що зберігаються на таких пристроях, як мобільні телефони. Він складається з двох основних компонентів: власників даних (учасників) і власника моделі (сервер інтегрованого навчання). Програми федеративного машинного навчання містять периферійні обчислення та кешування.

Основна роль кешу полягає в тому, щоб покращити час обробки та час обчислення, як правило, на комп'ютерах і електронних пристроях. Дані, які часто використовуються, зберігаються в кеш-пам'яті, і до них можна отримати миттєвий доступ. Основним обмеженням кеша є його обмеження розміру. З іншого боку, периферійні обчислення – це технологія, яка дозволяє обробляти інформацію поблизу межі мережі, де об'єкти та люди виробляють або споживають інформацію.

Вважається, що об'єднане навчання зменшує транзитний трафік у бездротових мережах із кешуванням і периферійними обчисленнями [55]. Інтегроване навчання тренує глобальну модель із локальними даними користувача, щоб визначити «популярну інформацію», яка буде зберігатися в кеші. Об'єднана модель, пройшовши навчання, може робити майбутні прогнози щодо того, що користувачі, як правило, переглядають, і забезпечувати частий доступ до програм. Крім того, це дозволяє програмам, які мають сувору затримку та пропускну спроможність, відповідати цим вимогам.

Найбільший недолік у впровадженні розподіленого навчання у великій популяції полягає в занепокоєнні конфіденційністю обміну взаємодіями користувача з мережею, як описано Нікнамом та ін. [5]. Проблеми з безпекою та конфіденційністю можна пом'якшити, модифікувавши федеративне навчання, задіявши безпечне агрегування, що його надає користувачам приватний простір, який не розкривається іншим користувачам. Це забезпечує конфіденційність користувача (учня), а не окремий пакет даних. Однак із додаванням алгоритмів продуктивність і обчислювальні ресурси приносяться в жертву.

Загалом, інтегроване навчання – це техніка, що її можна застосувати для розподілу ресурсів бездротової мережі, яка може вирішити проблеми з енергією, пропускнуною спроможністю, затримкою та конфіденційністю. Моделювання об'єданого навчання проводилося в мережі з підтримкою кешу з додатками доповненої реальності [5]. Буде цікаво побачити застосування федеративного навчання в бездротових мережах з різними типами користувачів, оскільки існує багато обмежень, які потрібно враховувати. Ці обмеження охоплюють питання конфіденційності окремих даних, які взаємодіють із мережею, загальний час конвергенції, оскільки він залежатиме від швидкості навчання окремих пристроїв користувача та якості каналу. Крім того, оскільки федеративне навчання залежить від децентралізованих даних, кількість користувачів, які бажають навчати глобальну модель, також впливатиме на продуктивність.

Бачення 6G – це тісно пов'язана багатообіцяюча парадигма, яка, як очікується, швидко реагує на запити користувачів через вивчення мережного контексту в реальному часі, як це визначено межею мережі (наприклад, діапазони базової станції [BS] і функції кешу), радіоконтролер (наприклад, зв'язок), спектр і канал передачі) і можливості пристрою (наприклад, ємність акумулятора та пункти призначення). Багатовимірну

складність усієї мережі, яка потребує інформації в реальному часі, можна подати як проблему квантової невизначеності. Розвиток концепцій ML, квантових обчислень (QC) і методів Q-навчання та їх стратегії з комунікаційними системами можна визнати необхідними в 6G. Quantum ML відіграватиме значну роль у використанні доступних ресурсів і великомасштабних даних для забезпечення розумної взаємодії в бездротовому зв'язку 6G. Це допоможе в усіх процесах, починаючи від проактивного кешування і закінчуючи оцінюванням величезної кількості каналів у безстільниковому режимі масового введення та виведення (MIMO).

3.3 Навчання з підкріпленням для розподілу функціональних сервісів в туманних обчисленнях

З появою Інтернету речей (IoT) розподілена хмарна архітектура стала потенційною можливістю для бізнесу, для більшості хмарних провайдерів [1].

Низька затримка та високі обмеження мобільності є одними з найсуворіших вимог до послуг IoT, що робить централізовані хмарні рішення непрактичними. У відповідь хмарні обчислення розвинулися до нової парадигми під назвою Fog Computing (FC) [2], де розподілена хмарна інфраструктура створена для надання послуг, наближених до кінцевих користувачів. Крім того, мікросервіси зараз революціонізують спосіб створення розробниками програмного забезпечення [3]. Програма розкладається на набір самодостатніх контейнерів, розгорнутих на великій кількості серверів, замість традиційної єдиної монолітної програми. Насправді контейнери є найбільш багатообіцяючою альтернативою звичайним віртуальним машинам (VM) через їх низькі накладні витрати та високу переносимість. Тим не менш, кілька проблем з точки зору надання ресурсів і планування послуг зберігаються, що заважає постачальникам послуг і кінцевим користувачам повністю скористатися шаблонами мікросервісів.

Однією з ключових проблем, яка залишається невирішеною, є ланцюжок функцій служби (SFC) [4], де служби з'єднуються в певному порядку, утворюючи ланцюжок послуг, який має пройти кожен запит для доступу до певної мережної служби (NS). Наприклад, ланцюжок послуг може складатися з інтерфейсу прикладного програмування (API), бази даних і служби машинного навчання (ML). Датчики отримують доступ до API, щоб надсилати свої дані в інфраструктуру, тоді як користувачі отримують доступ до служби бази даних, щоб отримати зібрані датчиком дані. Можливо, ці дані вже були відфільтровані та змінені службою ML. У середовищах туман-хмара взаємодія між місцями туману та хмарою має вирішальне значення для забезпечення належної роботи сервісів завдяки ієрархічній архітектурі. Наприклад, службу бази даних потрібно

розташувати поблизу користувачів у туманному місці, але службу ML можна створити в хмарі, де доступно більше обчислювальних ресурсів.

Нам потрібні належні стратегії надання, щоб переконатися, що обидві служби розміщені достатньо близько, щоб користувачі не відчували затримки під час доступу до прогнозованих результатів. Ці вимоги до ланцюга (наприклад, розташування служби, низька затримка, мінімальна доступна пропускна спроможність) мають бути гарантовані під час розподілу SFC у FC, які наразі не вивчаються, оскільки концепції SFC ще здебільшого не вивчені в середовищах FC.

Незважаючи на те, що теоретичні основи FC вже створені, прийняття його концепцій все ще знаходиться на ранніх стадіях, а практичне впровадження все ще є незначним. Крім того, поточні дослідження розподілу ресурсів в основному зосереджені на теоретичному моделюванні та евристичних рішеннях, які в більшості випадків не можуть впоратися з динамічною поведінкою мережі та призводять до поганого використання ресурсів і проблем масштабованості. Насправді розподіл ресурсів є складною онлайн-проблемою прийняття рішень, де належні дії залежать від повного розуміння мережного середовища. Таким чином, у цьому підрозділі ми описуємо дослідження підмножини ML під назвою Reinforcement Learning (RL) [5], щоб забезпечити відповідне рішення для розподілу SFC у FC. Проблему розподілу SFC було перетворено на проблему RL, де найкращі рішення щодо розподілу ресурсів (тобто дії) вивчаються залежно від поточного стану мережної інфраструктури (тобто середовища).

На основі наведеного раніше формулювання змішаного цілочислового лінійного програмування (MILP) було розроблено середовище, де агенти вчаться розподіляти ланцюжки послуг у FC безпосередньо під час взаємодії з середовищем без будь-яких знань чи інформації на початку. Наші результати показують, що методи RL працюють порівняно з сучасними реалізаціями на основі цілочислового лінійного програмування (ILP), але забезпечують більш масштабовані рішення.

Підсумовуючи, FC є однією з найскладніших тем у сучасних хмарних обчисленнях разом із концепціями розподілу ресурсів і ланцюжка послуг. Решта підрозділу організована таким чином: короткий огляд технічної бази; обговорення сучасного стану розподілу ресурсів для FC; подання підходу RL для розподілу SFC у FC, за яким слідує налаштування оцінювання. Далі показані результати. Нарешті, майбутні напрямки досліджень і відкриті проблеми обговорюються на завершення.

У цьому підрозділі подано короткий огляд парадигми FC. Потім обговорюються фундаментальні концепції, пов'язані з розподілом ресурсів і SFC. Нарешті розтлумачено основні поняття RL.

Парадигма FC є розширенням хмарних обчислень для надання ресурсів на межах мережі для роботи з експоненціальним зростанням підключених пристроїв [7]. На рисунку 3.7 показано високорівневий вигляд середовища

FC. На відміну від централізованої хмари, вузли туману розподілені по мережі, щоб діяти як проміжний рівень між кінцевими пристроями та хмарою. Ці так звані туманні вузли, крайові розташування або навіть Cloudlets [8], по суті, є невеликими хмарними об'єктами, які наближають обчислювальну потужність, процедури зберігання та ємність пам'яті до пристроїв і кінцевих користувачів, щоб забезпечити локальні операції. Хмарні вузли – це традиційні хмарні сервери, де доступна велика кількість ресурсів.

Надання ресурсів що також відоме як розподіл ресурсів вивчалось роками в сфері керування мережею [9–11]. Надання ресурсів пов'язане з розподілом обчислювальних, мережних ресурсів і ресурсів зберігання, необхідних для створення екземплярів послуг, запитуваних користувачами та пристроями через Інтернет. Останніми роками постачальники та користувачі хмарних технологій працюють разом над ефективним способом роботи з обчислювальними ресурсами. З одного боку, користувачі хочуть отримати найкращу якість обслуговування (QoS) за мінімальну вартість, тоді як хмарні провайдери хочуть збільшити свій дохід. Користувачі хочуть максимізувати свій план обслуговування без збільшення своїх витрат, тоді як хмарні постачальники хочуть дотримуватися узгодженого рівня QoS, використовуючи мінімальний відсоток своєї інфраструктури. Таким чином, енергоефективність є важливою для хмарних провайдерів, тоді як низька затримка має вирішальне значення для користувачів. Зниження витрат за рахунок використання мінімальної кількості апаратного забезпечення, гарантуючи користувачам рівень QoS, або збільшення кількості активних вузлів, щоб зменшити затримку між розгорнутою службою та користувачем до мінімуму. Ефективні стратегії розподілу мають вирішальне значення як для постачальників хмарних послуг, так і для користувачів. Залежно від стану мережної інфраструктури або поточного попиту користувача можуть застосовуватися різні політики забезпечення.

Крім того, з появою FC розподіл ресурсів став ще більш важливою темою дослідження. FC був створений як відповідь на проблеми, пов'язані з наданням послуг, пов'язаними з послугами IoT. Наприклад, послуги IoT є дуже складними з точки зору затримки. Сервіси, чутливі до затримки (наприклад, підключені транспортні засоби, інтерактивні відеопрограми), потребують затримки порядку мілісекунд. Якщо затримка збільшується, перевищуючи поріг зв'язку, з'єднання користувача може стати нестабільним, і користувач може втратити контроль над службою. Крім того, оскільки транспортні засоби та користувачі безперервно рухаються в зоні мережі, мобільність є ще одним важливим фактором, який потрібно враховувати. Стратегії розподілу мають враховувати перерозподіл послуг у разі втрати підключення користувача, щоб забезпечити належну роботу служби в будь-який час. Централізовані інфраструктури не можуть повністю задовольнити динамічні потреби цих видів послуг. Таким чином,

FC має важливе значення для швидкої зміни розподілу послуг відповідно до дуже мінливих моделей попиту.

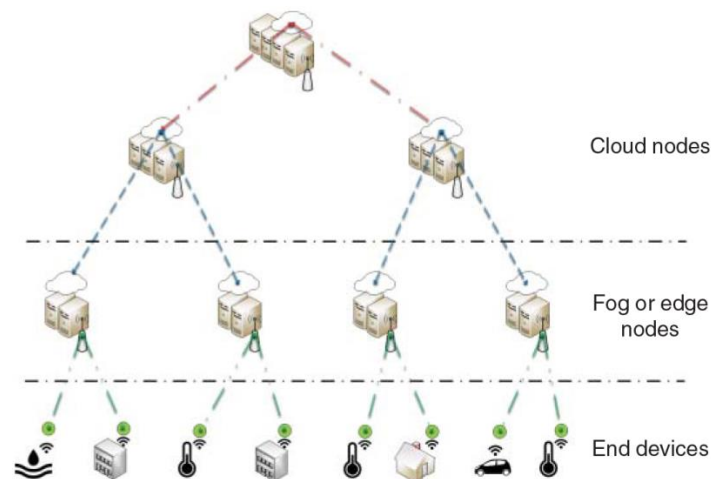


Рисунок 3.7 – Архітектура багаторівневого обчислювального туманного середовища

Розміщення SFC [12, 13] вивчалось у сфері управління мережею протягом останніх кількох років. SFC пов'язаний із належним упорядкуванням служб, гарантуючи, що кожен користувач має пройти заданий ланцюжок служб для доступу до конкретного NS, як показано на рисунку 3.8. Кола показують різні сервісні функції, тоді як стрілки показують, як спрямовується трафік у мережі. Запити користувачів направляються через сервісний ланцюжок відповідно до сервісного графіка, який спрямований на оптимізацію розподілу ресурсів для подальшого підвищення продуктивності програми. SFC дозволяє хмарним провайдерам динамічно змінювати конфігурацію програмних NS без необхідності вносити зміни на апаратному рівні. SFC надає гнучку та надійну альтернативу сучасному статичному мережному середовищу.

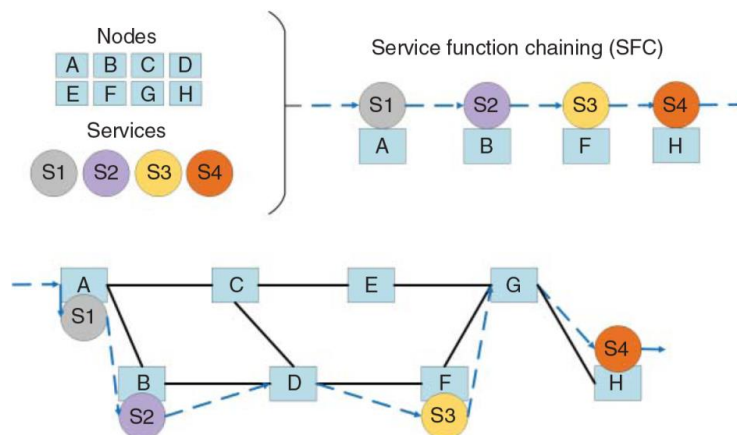


Рисунок 3.8 – Схема розгортання службового сервісного функціонала

Останнім часом надзвичайну увагу привернули шаблони мікросервісів [15]. Програма розкладається на набір слабо пов'язаних служб, які можна розробляти, розгортати та підтримувати незалежно. Кожна служба відповідає за одне завдання та спілкується з іншими службами за допомогою спрощених протоколів. Потім ці служби можна розробляти різними мовами програмування та навіть з використанням різних технологій. Контейнери є найбільш багатообіцяючою альтернативою традиційній монолітній парадигмі додатків, де майже все централізоване та навантажене кодом.

В останні роки методи RL стали важливою сферою досліджень ML [16–18]. Типовий сценарій в RL наведений на рисунку 3.9. У більшості випадків методи RL використовуються для вирішення завдань послідовного прийняття рішень. Агент RL вчиться приймати кращі рішення безпосередньо з досвіду взаємодії з навколишнім середовищем. Середовище подає проблему, яку потрібно вирішити. На початку агент нічого не знає про наявну проблему і вчиться, виконуючи дії в середовищі. За кожну виконану дію агент отримує винагороду та нове спостереження, яке описує новий стан середовища. Залежно від мети та того, наскільки добре агент виконує поставлене завдання, винагорода може бути позитивною або негативною. Агент вчиться досягати успіху, постійно взаємодіючи з середовищем, визначаючи притаманну синергію між станами, діями та подальшими винагородами. Зрештою, алгоритми RL намагаються максимізувати загальну винагороду, яку агент міг би отримати, використовуючи кілька проблемних раундів. Наприклад, розглянемо агента, який розподіляє ресурси в хмарній інфраструктурі.

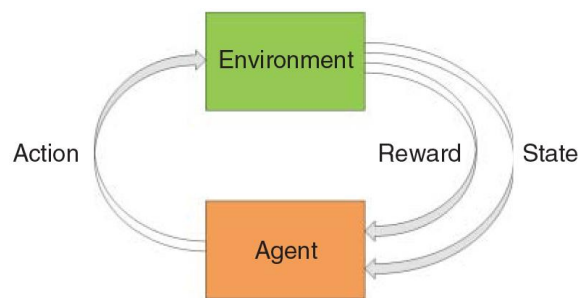


Рисунок 3.9 – Схема подання більшості сценаріїв RL

Агент отримувати бинагороду за кожну дію, застосовану в системі. Якщо дія перетворюється на відповідну схему розподілу, агент отримає позитивну винагороду. В іншому випадку, якщо агент виконує неправильну дію (наприклад, припиняє послугу, необхідну в мережі), що створює невідповідну схему розподілу, винагорода буде негативною. Щоб максимізувати винагороду, агент має постійно застосовувати дії, які перетворюються на належні схеми розподілу.

Кінцевою метою цього сценарію було б навчити агента, здатного навчитися правильному розподілу, щоб максимізувати продуктивність і мінімізувати витрати.

З появою концепцій FC у сучасних хмарних інфраструктурах потрібне ефективне забезпечення ресурсами. У цьому підрозділі наведено короткий виклад відповідних попередніх досліджень, що стосуються специфікацій і реалізації стратегій розподілу ресурсів для інфраструктур туманно-хмарних технологій. По-перше, описано дослідження розподілу ресурсів для FC, за яким слідує нещодавні досягнення в забезпеченні ресурсами за допомогою методів ML. Нарешті, висвітлено останні роботи з RL для розподілу ресурсів.

Було проведено декілька досліджень у контексті надання ресурсів у середовищах FC. У [19] автори запропонували схему розподілу для підтримки програм краудсенсингу в контексті IoT.

Їхній підхід був сформульований як модель MILP, яка враховує економічно ефективно забезпечення та розподіл завдань. Результати підтвердили, що їхня пропозиція може перевершити традиційні хмарні інфраструктури. У [20] було запропоновано формулювання оптимізації для розгортання додатків IoT з урахуванням якості обслуговування на інфраструктурах туману та реалізовано як прототип під назвою FogTorch. Їхній підхід зосереджувався не лише на вимогах до апаратного та програмного забезпечення, але й на таких вимогах до якості обслуговування, як затримка мережі та пропускна спроможність. Результати показали, що їхній алгоритм забезпечує оптимальне розгортання сервісу, одночасно зменшуючи потужність обладнання та збільшуючи вимоги до ресурсів. Крім того, в [21] проблема розподілу ресурсів IoT у FC була змодельована як формулювання ILP, де розглядалися метрики QoS і кінцеві терміни для розгортання кожної програми. Результати довели, що їх формулювання може запобігти порушенням QoS і знизити витрати порівняно з традиційним хмарним підходом. Крім того, в [22] був запропонований алгоритм оптимізації роїв частинок для проблеми розподілу ресурсів в інфраструктурах туману та хмар, спеціально зосереджених на розумних будівлях. Результати показали, що їх підхід може зменшити час відгуку та вартість розподілу віртуальних машин. У [23] була модель ILP для проблеми забезпечення ресурсами туману, а потім евристичний алгоритм, здатний знаходити неоптимальні рішення, хоча й досягає кращої ефективності часу. У своїй роботі автори вивчали компроміс між максимізацією надійності та мінімізацією загальної вартості системи. Більш того, у [24] були презентовані стратегії розміщення послуг для FC на основі алгоритмів зіставлення гри. З одного боку, перший підхід базується на концепціях SFC, оскільки розглядається впорядкована послідовність послуг, запитуваних кожною програмою. З іншого боку, другий формулює проблему, не звертаючи уваги на структуру ланцюга, щоб знизити складність обчислень без шкоди для продуктивності. Крім того, в [25] був

описаний оркестровник крайового контейнера для малопотужних пристроїв під назвою FLEDGE. Результати показали, що FLEDGE мінімізує витрати на ресурси порівняно з іншими платформами. Нещодавно ця робота була розширена в [26], де вивчалася масштабованість і нестабільність інфраструктури туман-хмара. Автори запропонували алгоритм планування для розподілу послуг у великомасштабному розгортанні туману, здатний адаптуватися до змін мережі.

Незважаючи на те, що більшість цитованих досліджень стосувалися питань розподілу в FC, жодне з вищезгаданих досліджень не розглядало реалістичні вимоги до QoS або будь-які обмеження, що впливають з високих вимог, що висуваються IoT (наприклад, порогові значення затримки, розташування служби, послуги на основі контейнерів). Крім того, більшість досліджень зосереджено на теоретичному моделюванні та імітаційних дослідженнях, що обмежує їх практичне впровадження.

Завдяки останнім досягненням у ML були проведені дослідження щодо застосування методів ML до проблем розподілу ресурсів. У [27] автори запропонували методи навчання під наглядом для прогнозування майбутніх запитів віртуалізації мережних функцій (NFV).

Їхня мета полягає в активному розподілі ресурсів на основі раніше спостережених закономірностей. Результати показали, що пропозиція може проактивно задовольняти запити NFV.

У [28] були запропоновані моделі нейронних мереж для вирішення проблеми автоматичного масштабування функції віртуальної мережі (VNF) у мережах 5G. Їх мета полягає в тому, щоб передбачити необхідну кількість VNF в даний момент на основі попередніх запитів на трафік. Крім того, було наведено формулювання ILP для вирішення проблеми розподілу SFC. Результати засвідчили, що середня наскрізна затримка (E2E) значно зменшується, коли ланцюжки обслуговування розподіляються на межі. У [29] модель ML була використана для прогнозування вимог до ресурсів VNF з високою точністю (наприклад, ЦП). Цей підхід можна застосувати до таких проблем розподілу SFC, як автоматичне масштабування та оптимальне розміщення. Крім того, в [30] інфраструктура туману була змодельована як розподілена інтелектуальна система обробки під назвою SmartFog за допомогою методів ML і теорії графів. Даний підхід забезпечує прийняття рішень із малою затримкою та адаптивне керування ресурсами через архітектуру туману.

Таким чином, контрольовані та неконтрольовані методи ML були реалізовані в літературі для покращення прийняття рішень у хмарних інфраструктурах.

Більшість цитованих досліджень стосується проблем розподілу та автоматичного масштабування, що їх традиційні методи (наприклад, теоретичне моделювання, евристичні алгоритми) не змогли повністю вирішити через динамічну поведінку мережі.

Останнім часом методам RL приділяється значна увага в сфері розподілу ресурсів. У [31] була презентована техніка глибокого RL під назвою DeepRM для вирішення проблеми розміщення завдань у системі керування хмарою. Початкові результати показали, що DeepRM працює так само, як і розв'язання на основі евристики, і що він може вивчати різні стратегії залежно від стану мережі. У [32] проблема розподілу послуг IoT розглядається шляхом використання механізму RL для обчислення задовільних рівнів якості досвіду (QoE). Отримані оцінки підтвердили ефективність застосованих методів. Крім того, в [33] була описана основа оптимізації на базі RL для вирішення проблеми розподілу ресурсів у периферійних обчисленнях бездротового багатоканального доступу (MEC). Мета полягає в мінімізації витрат при оптимізації ресурсів. Було наведено результати моделювання, де запропоновані методи досягли значного зниження витрат. У [34] методи RL були вивчені на предмет їх застосовності до проблеми розподілу SFC в оптичних мережах із підтримкою NFV–SDN (Software Defined Networking).

Результати продемонстрували переваги використання оптимізацій на основі RL перед методами на основі правил. Крім того, в [35] було наведено механізм планування ресурсів туману на основі глибокого RL. Підхід тут зосереджений на транспортних сценаріях використання FC з метою мінімізації споживання часу додатками, пов'язаними з безпекою.

Результати показали, що запропоновані ними схеми можуть зменшити споживання часу порівняно з традиційними підходами.

Таким чином, протягом останніх років методи RL довели свою потенційну застосовність до питань розподілу ресурсів. Однак результативність методів RL глибоко взаємопов'язана з тим, як налаштовано середовище та система винагороди. Залежно від припущень, зроблених у системі, методи RL можуть дати абсолютно різні результати. Наскільки нам відомо, методи RL ще не застосовувалися для розподілу SFC, де передбачалися інфраструктури туманно-хмарних технологій і контейнерні служби. Крім того, динамічна поведінка мережі та різні стратегії планування (наприклад, низька затримка, енергоефективність) не були повністю розглянуті. Проте методи RL довели, що навчання безпосередньо з досвіду може працювати на практиці та запропонувати справжню альтернативу підходам, заснованим на евристичних дослідженнях. Таким чином, пропонується новий підхід RL для розподілу SFC у FC.

У цьому підрозділі розглянуто підхід RL для розподілу SFC у FC. По-перше, наведено формулювання проблеми розподілу IoT. Потім описуються простори спостереження та дії з нашого середовища RL. Нарешті, презентовано функцію винагороди та агента.

Як згадувалося, проблема розподілу IoT була змодельована як формулювання MILP, наведене раніше в [36]. Модель розглядає туманно-хмарну інфраструктуру, де можуть бути розміщені контейнерні ланцюжки

послуг. Програма IoT розкладається на набір мікросервісів, які мають певний коефіцієнт реплікації для балансування навантаження або резервування. Очікується, що кілька користувачів отримають доступ до цих мікросервісів. Інфраструктура туманно-хмарної хмари керує набором вузлів, у яких екземпляри мікросервісів мають бути виділені на основі її вимог і з урахуванням кількох обмежень. Наприклад, вузли мають обмежені можливості (наприклад, ЦП і пам'ять), і всі мікросервіси, що входять до складу певної програми, мають бути виділені в мережі, щоб програму можна було вважати розгорнутою. Формулювання MILP було переведено в середовище RL під назвою gum-fog, де можна виконувати дії та на кожному кроці часу дається нове спостереження, яке описує новий стан середовища. Потрібно зазначити, що для розробки підходу RL розглядалися лише хмарні формулювання, а бездротові аспекти, доступні в моделі, не використовувалися.

Для цієї роботи сформульовано нову мету для моделі MILP: мінімізація загальної вартості системи, що перетворюється на підвищення енерго-ефективності. Використовуючи номенклатуру складу MILP, наведену в таблиці 3.8, цю мету можна виразити, як показано в рівнянні (3.1). Агент намагатиметься дізнатися, як мінімізувати загальну вартість системи, оскільки формула MILP взаємодіє з середовищем спортивного туману.

$$\sum_{a \in A} \sum_{id \in ID} \sum_{s \in S} \sum_{\beta_i \in \beta} \sum_{n \in N} p_{s, \beta_i}^{a, id}(n) \times \varpi_n \times \omega_s \times \gamma_s \times \delta_s \quad (3.1)$$

Таблиця 3.8 – Змінні, що використовуються для мінімізації загальної вартості системи

Символ	Опис
$p_{s, \beta_i}^{a, id}(n)$	Матриця розміщення. якщо $p_{s, \beta_i}^{a, id}(n) = 1$, репліка β_i мікросервісу S виконується на вузлі n для програми a з ідентифікатором SFC id
ϖ_n	Відповідна вага для вузла n
ω_s	Вимога до ЦП (у ЦП) мікросервісу S
γ_s	Вимога до пам'яті (у ГБ) мікросервісу S
δ_s	Вимога до пропускної спроможності (у Мбіт/с) мікросервісу S

Простір спостереження відповідає стану, що подає середовище на даному кроці. Наприклад, розглянемо агента, який грає в шахи, спостереження буде мати статус дошки для конкретної гри. У впровадженому спортивному тумані простір для спостереження було спроектовано, як показано в таблиці 3.9. Щоб легше зрозуміти нашу методологію, давайте розглянемо невелику інфраструктуру, де всі запити користувачів, що надходять до нашої системи, виконуються на основі програми IoT, розбитої на два окремі мікросервіси. Простір спостереження буде складатися з п'яти метрик. Дві метрики (RS1 і RS2) показують співвідношення між схемою розподілу, запропонованою агентом, і моделлю

MILP для кожного мікросервісу. Потім дві метрики (RL і MILP) показують загальну вартість системи, задану агентом і моделлю MILP відповідно. Нарешті, остання метрика (UR) стосується точної кількості запитів користувачів, зроблених у мережі на цьому конкретному кроці. Таким чином, простір спостереження лінійно збільшується з кількістю додатків, доступних у моделі MILP, і відповідних їм мікросервісів.

Таблиця 3.9 – Частка вибірки простору спостереження середовища спортивного залу з туманом

Metric name	Description
Ratio S1 (RS1)	The relation of allocated micro-service 1 instances between the MILP model and the agent
Ratio S2 (RS2)	The relation of allocated micro-service 2 instances between the MILP model and the agent
Cost RL (RL)	The agent allocation scheme cost
Cost MILP (MILP)	The MILP allocation scheme cost
User requests (UR)	The number of user requests at the given moment

Простір дій відповідає всім діям, які агент може застосувати в середовищі. Враховуючи ту ж аналогію з шахами, що й раніше, простір дій у шаховій грі полягав би у виборі кожної фігури та переміщенні її на певну позицію на дошці.

В туманно-хмарній інфраструктурі простір дій має охоплювати розподіл і припинення всіх мікросервісів, доступних у системі. Простір дій у середовищі *gum-fog* було розроблено, як показано в таблиці 3.10, припускаючи, що той самий додаток IoT складається з двох мікросервісів, а інфраструктура туману та хмари подана лише одним вузлом. Простір дії складається з п'яти окремих дій. Простір дій також лінійно збільшується з кількістю мікросервісів і кількістю вузлів, доступних в інфраструктурі. Перша дія називається *DoNothing*, оскільки, якщо її застосувати, у мережі агента не буде виконано розподіл або завершення. Таким чином, агент має вибрати цю дію лише тоді, коли поточна схема розподілу відповідає поточному попиту мережі. Другий набір дій відповідає виділенню екземплярів мікросервісу (*Deploy-Si-Ni*). Агент може вибрати який саме екземпляр мікросервісу потрібно виділити та на якому вузлі його потрібно виконати. Простір дій розроблено таким чином, щоб гарантувати, що агент може знайти кращі варіанти розподілу, вибравши конкретний екземпляр мікросервісу для розгортання на певному вузлі. Агент може застосувати певну дію з цього набору кілька разів, якщо в мережі потрібно більше екземплярів одного мікросервісу для підтримки всіх запитів користувачів. Нарешті, останній набір дій (*Stop-Si-Ni*) відповідає припиненню екземплярів мікросервісу. Як і в діях розподілу, агент вибирає яку мікрослужбу потрібно припинити та який це екземпляр, оскільки вузол, де

розгорнуто мікрослужбу, також задано. Наша мета полягає в тому, щоб навчити агента, що певну кількість екземплярів мікросервісу необхідно виділити для належної роботи ланцюга та підтримки всіх запитів користувачів.

Таблиця 3.10 – Частка вибірки простору дії середовища спортивного залу з туманом

Action label	Description
DoNothing	The agent does nothing
Deploy-S1-N1	Allocate a micro-service 1 instance in node 1
Deploy-S2-N1	Allocate a micro-service 2 instance in node 1
Stop-S1-N1	Terminate the micro-service 1 instance in node 1
Stop-S2-N1	Terminate the micro-service 2 instance in node 1

Мета функції винагороди полягає в тому, щоб навчити агента, як максимізувати накопичену винагороду шляхом вибору відповідних дій залежно від спостереження, наданого середовищем. За кожну дію, яку обирає агент, отримується певна винагорода. Ця винагорода може бути позитивною або негативною. Таким чином, агент може дізнатися, чи була обрана ним дія доцільною, виходячи з отриманої винагороди. Розробка відповідної функції винагороди за допомогою ручного налаштування параметрів ML потрібна для забезпечення того, щоб агент дізнався, що він має робити. Функція винагороди, реалізована в середовищі gym-fog, показана в алгоритмі 3.1. Мета агента – навчитися розподіляти мікросервіси в інфраструктурі туманно-хмарних технологій відповідно до формулювання MILP. Модель MILP надає послуги в зоні мережі, мінімізуючи загальну вартість системи, як було показано раніше. Тому чим ближче агент до досягнення рішення MILP, тим вищу винагороду він отримує. По-перше, винагороди розраховуються на основі обмежень, внесених у модель MILP. Наприклад, було додано обмеження, щоб обмежити виділення одного екземпляра того самого мікросервісу на вузол. Таким чином, якщо агент вибере дію, яка скасовує це обмеження, агент отримає негативну винагороду (тобто, -1). Потім індивідуальні винагороди розраховуються для кожного коефіцієнта мікропослуги, як показано в алгоритмі 3.2. По-перше, якщо кількість виділених екземплярів мікросервісу агентом дорівнює нулю, винагорода -5 отримується, оскільки агент не виділяє жодного екземпляра цього мікросервісу, що перешкоджає належній роботі ланцюжка сервісів. По-друге, якщо кількість виділених екземплярів мікросервісу агентом дорівнює кількості, виділеній моделлю MILP, повертається винагорода 5.

Нарешті, винагорода у розмірі -1 отримується, якщо агент виділяє більшу кількість реплік, які не потрібні (тобто надмірне надання), або якщо агент виділяє менше екземплярів, ніж потрібно (тобто недостатнє надання).

Після розрахунку коефіцієнта винагороди виконується функція винагороди за вартістю, як показано в алгоритмі 3.3. По-перше, якщо вартість агента нижча, ніж MILP, повертається негативна винагорода, оскільки агент не може мати нижчу вартість, оскільки рішення MILP є оптимальним. Таким чином, агент, ймовірно, порушує кілька обмежень проблеми служби IoT. Тоді, якщо вартість агента дорівнює або до 10% перевищує вартість MILP, повертається 10, оскільки агент працює подібно до моделі MILP.

Алгоритм 3.1 – Функція винагороди в середовищі gym-fog

Input: Observation state after action step in

Output: Reward out

```

1: // Return the reward for the given state
2: getReward(obs):
3:   reward = 0
4:   ratioS1 = obs.get(1)
5:   ratioS2 = obs.get(2)
6:   costRL = obs.get(3)
7:   costMILP = obs.get(4)
8:
9:   // Reward based on Keywords for MILP constraints
10:  // Constraint: MAX micro-services on a single Node
11:  // Constraint: Terminate micro-service without deployment first
12:  // Constraint: MAX micro-service instances reached
13:  if constraintMaxServicesOnNode == True then
14:    return -1
15:  if constraintTerminateServiceFirst == True then
16:    return -1
17:  if constraintMAXServiceInstances == True then
18:    return -1
19:
20:  // Micro-service ratio Reward calculation
21:  reward = reward + getRatioReward(ratioS1)
22:  reward = reward + getRatioReward(ratioS2)
23:
24:  // Cost Reward Calculation
25:  reward = reward + getCostReward(costRL, costMILP)
26:
27:  // Ultimate Goal calculation
28:  if ratioS1 == 1 and ratioS2 == 1 then
29:    if costRL > costMILP then // High Reward
30:      reward = reward + 10
31:    if costRL == costMILP then // MAX Reward
32:      reward = reward + 100
33:
34:  return reward

```

Потім, залежно від того, наскільки вища вартість агента порівняно з MILP, винагорода повертається, тобто агента навчають, що чим ближче він залишається до вартості MILP, тим вищу винагороду він отримує. Тим не менш, кінцева мета полягає в тому, щоб досягти подібних витрат до моделі MILP і виділити всі потрібні екземпляри мікросервісів для прийняття всіх запитів користувачів. Таким чином, агенту можна надати дві бонусні винагороди, якщо коефіцієнти всіх мікропослуг дорівнюють 1. По-перше, якщо вартість агента вища за вартість MILP, надається бонусна винагорода 10, оскільки агент розподілив усі мікропослуги, що необхідні в мережі, незважаючи на більш високу вартість. По-друге, якщо вартість агента збігається з вартістю MILP, отримується бонус у розмірі 100, оскільки агент виконав саме те, що мав.

Агент навчився розмішувати мікросервіси в інфраструктурі туману та хмари як формулювання MILP.

У цьому підрозділі наведено агент Q-навчання, який використовується для оцінювання середовища з туманом у спортзалі. Q-навчання [37] – це класичний алгоритм RL, який вивчає найкращу дію для вибору в заданому стані, досліджуючи кожен пару стан–дія $Q(s, a)$.

Q-Learning – це метод RL поза політикою, оскільки агент дізнається оптимальну політику (π) незалежно від застосованих дій на основі двоетапного процесу. Перший процес називається експлуатацією, де Q-таблиця розраховується як базова лінія для всіх можливих дій для даного стану. Потім буде застосовано дію з вищим значенням (тобто максимальна винагорода). Друга операція називається дослідженням, оскільки замість вибору дій на основі максимальної майбутньої винагороди агент вибирає випадковим чином дію, яка дозволяє досліджувати та відкривати нові стани, які інакше не можна було б досліджувати через процес експлуатації. Ставки розвідки та експлуатації можна встановлювати під час виконання, таким чином забезпечується повний контроль над алгоритмом.

Основна проблема з агентами Q-навчання полягає в тому, що їм потрібно бачити всі пари дія-стан для даного середовища, щоб мати можливість застосовувати дії, які максимізують винагороду. Зі зростанням розміру проблеми подання всіх пар стан-дія в пам'яті стає непомірним. Наприклад, ускладнення середовища тренувального туману (наприклад, додавання вузлів до інфраструктури, додавання додаткових сервісів до ланцюга обслуговування) має серйозний вплив на пам'ять і час виконання, оскільки це безпосередньо пов'язано з розміром дії та простір стану.

Алгоритм 3.2 – Розрахунок винагороди коефіцієнта мікропослуги

Input: Micro-service Ratio observation state in

Output: Ratio reward out

```
1: // Return the reward for the given micro-service ratio
2: getRatioReward(ratio):
3:   if ratio == 0 then // No service deployed - Bad solution
4:     return -5
5:   else if ratio == 1 then // Equal to the MILP - Good solution
6:     return 5
7:   else then // Under / Over-provisioning scheme
8:     return -1
```

Алгоритм 3.3 – Розрахунок винагороди за вартість

Input: CostRL, CostMILP in

Output: Cost reward out

```
1: // Return the reward for the relation between the CostRL and CostMILP
2: getCostReward(costRL, costMILP):
3:   if costRL < costMILP then // Lower than MILP - Bad solution
4:     return -10
5:   else if costMILP ≤ costRL ≤ 1.10×costMILP then // Best Solution
6:     return 10
7:   else if 1.10×costMILP < costRL ≤ 1.25×costMILP then
8:     return -2
9:   else if 1.25×costMILP < costRL ≤ 1.75×costMILP then
10:    return -4
11:  else if 1.75×costMILP < costRL ≤ 2.0×costMILP then
12:    return -6
13:  else if 2.0×costMILP < costRL ≤ 3.0×costMILP then
14:    return -8
15:  else if 3.0×costMILP < costRL ≤ 4.0×costMILP then
16:    return -10
17:  else then // > 4×costMILP
18:    return -20
```

Таким чином, щоб зменшити складність простору, простір спостереження було дискретизовано, як показано в таблиці 3.11, де визначено певний діапазон для кожної метрики спостереження, що значно зменшує кількість станів, які має враховувати агент Q-навчання. Припускаючи попередню туманно-хмарну інфраструктуру, простір спостереження було б скорочено до 288 окремих станів. По-перше, метрики

спостереження щодо розподілу мікросервісів було зведено до трьох областей. Наприклад, коефіцієнт S1 може дорівнювати лише 0, 1 (тобто, всі інші можливості групуються). Ці три стани є єдиними станами, які агент Q-навчання має враховувати, щоб знайти правильні дії щодо метрики Ratio S1. Крім того, дві метрики спостереження за витратами (costRL і costMILP) були об'єднані в нову метрику під назвою «вартість», де різниця між цими двома використовується для формулювання восьми станів на основі попередньо показаної функції компенсації витрат. Нарешті, запити користувачів також агрегуються в чотири стани на основі рішень, наданих моделлю MILP, які змінюються залежно від ланцюжків послуг, які потрібно розподілити, і від розглянутої інфраструктури туману та хмари.

У цьому підрозділі описано туманно-хмарну інфраструктуру, яка використовується для оцінювання середовища спортивного туману. Потім детально описується реалізація середовища, а потім відповідна конфігурація, застосована при оцінюванні.

Інфраструктура туман-хмара, зображена на рисунку 3.10, була подана в середовищі туману в спортзалі. Розглянуто загальну площу 324 км². Туманно-хмарна інфраструктура розгорнута в п'яти місцях L, де можливе розміщення мікросервісу. Кожна локація керує набором із трьох вузлів.

Таблиця 3.11 – Зменшення складності простору спостереження, який має враховувати агент Q-навчання

Metric name	Number of states
RatioS1	Three states (ratio calculation): [RS1 = 0, RS1 = 1, else]
RatioS2	Three states (ratio calculation): [RS2 = 0, RS2 = 1, else]
Cost	Eight states (cost calculation): [RL < MILP, MILP ≤ RL ≤ 1.10×MILP, ..., RL > 4.0×MILP]
UserRequests	Four states: [UR ≤ 20, UR ≤ 32, UR ≤ 40, UR ≤ 50]
Total	288 states (3 × 3 × 8 × 4)

Апаратні конфігурації кожного вузла наведено в таблиці 3.12. Кожен вузол має задану обчислювальну потужність (тобто, процесор, оперативну пам'ять і пропускну спроможність) і певну вагу, що є необхідною інформацією для розрахунку загальної вартості системи на основі формулювання MILP.

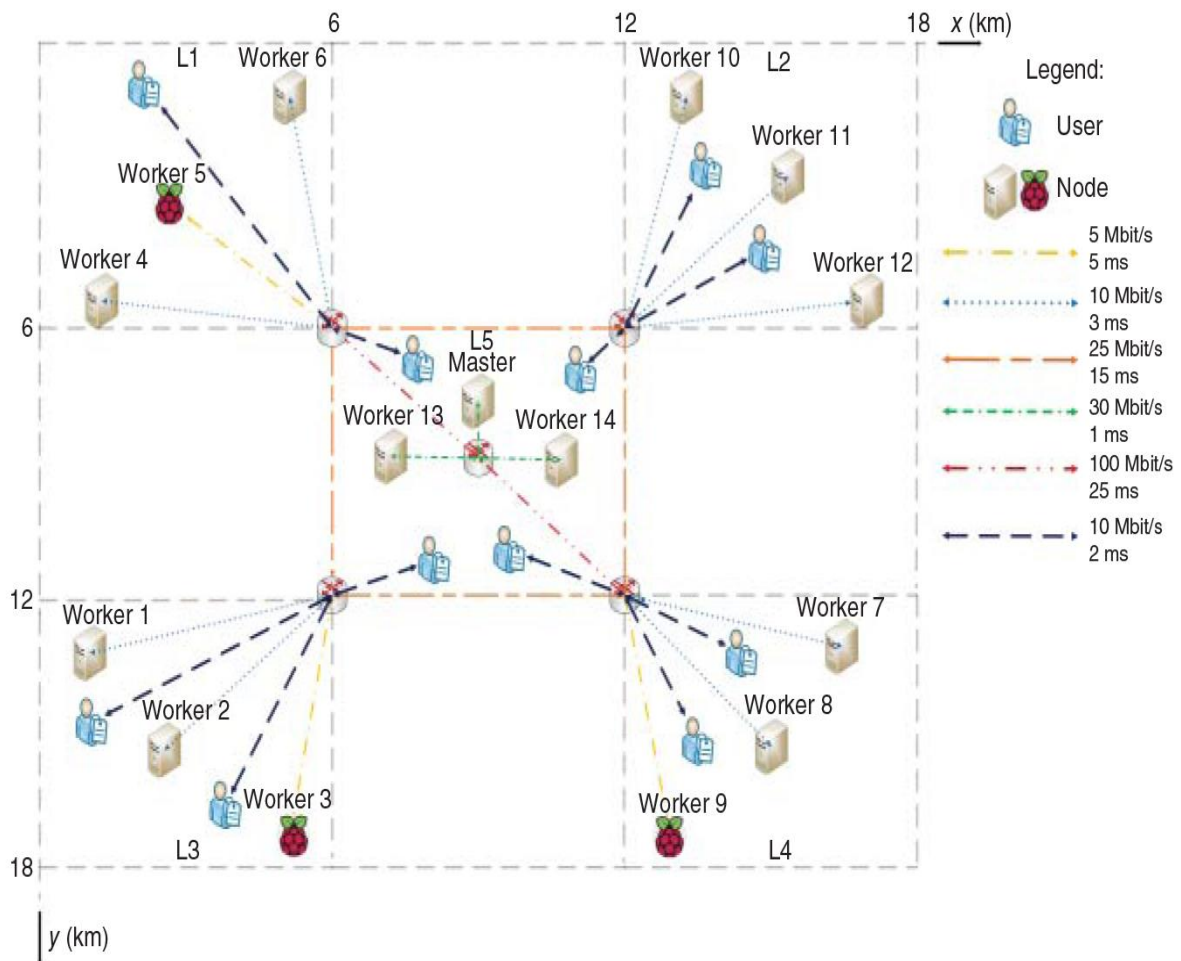


Рисунок 3.10 – Інфраструктура туман-хмара для оцінювання середовища спортивного туману

Середовище gym-fog було розроблено на основі OpenAi gym [38]. OpenAi gym – це інструментарій з відкритим вихідним кодом для дослідження RL, написаний на Python. Він містить набір тестових задач, які демонструють стандартизований інтерфейс, що порівнює алгоритми RL з точки зору продуктивності. Формулювання MILP, спочатку розроблену на Java, було переписано на Python, щоб полегшити взаємодію між моделлю MILP і OpenAi gym. Середовище gym-fog було побудовано на основі структури OpenAi gym, як показано на рисунку 3.11. Щоб почати експеримент, запускається функція ініціалізації. Потім, під час навчання, на кожній ітерації агент вибирає дію, яка потім передається в середовище OpenAi gym через функцію кроку, де повертається новий стан і відповідна винагорода. Крім того, функція скидання використовується на початку або після завершення кожного епізоду, щоб відновити початковий стан середовища. Крім того, функція відтворення може використовуватися для відтворення середовища після кожного кроку. Нарешті, функція close викликається, коли процес навчання завершено, щоб правильно завершити середовище.

Таблиця 3.12 – Апаратна конфігурація кожного вузла

Node	CPU (cpu)	RAM (Mi)	Band. (Mbit/s)	Weight
Worker 1	2.0	4.0	10.0	2.0
Worker 2	2.0	4.0	10.0	2.0
Worker 3	1.0	2.0	5.0	1.0
Worker 4	2.0	4.0	10.0	2.0
Worker 5	1.0	2.0	5.0	1.0
Worker 6	2.0	4.0	10.0	2.0
Worker 7	2.0	4.0	10.0	2.0
Worker 8	2.0	4.0	10.0	2.0
Worker 9	1.0	2.0	5.0	1.0
Worker 10	2.0	4.0	10.0	2.0
Worker 11	2.0	4.0	10.0	2.0
Worker 12	2.0	4.0	10.0	2.0
Worker 13	6.0	16.0	30.0	3.0
Worker 14	6.0	16.0	30.0	3.0
Master	8.0	24.0	30.0	3.0

Реалізований gym-fog використовує згадані функції для взаємодії з моделлю MILP. Насправді, спортзал OpenAi діє як міст між нашою моделлю MILP та агентом.

Конфігурація середовища спортивного залу та туману показана в таблиці 3.13 на основі описаної інфраструктури туману та хмари. Доступна одна програма, яка розкладається на три мікросервіси, що утворюють ланцюжок послуг. Максимальний коефіцієнт реплікації відповідає 5, тобто модель MILP або агент може розгорнути лише п'ять екземплярів мікросервісу одного типу мікросервісу.

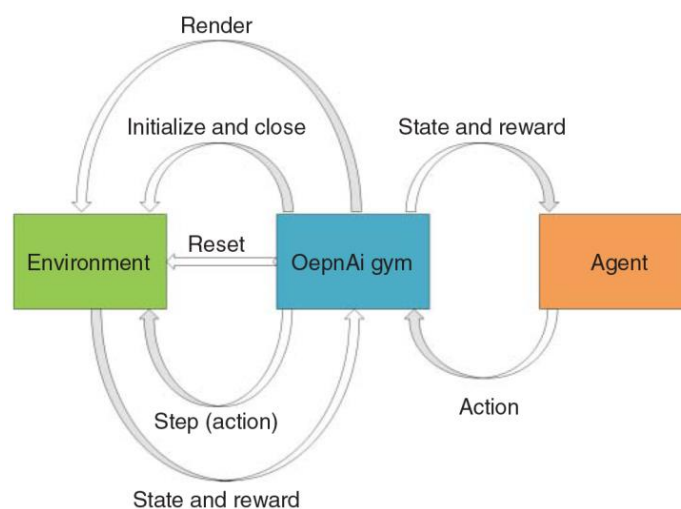


Рисунок 3.11 – Структура середовища тренажерного залу OpenAi

У цьому підрозділі наведено отримані результати. По-перше, було оцінено статичний сценарій, коли кількість запитів користувачів залишається постійною протягом усього оцінювання. Потім оцінюється динамічний варіант використання, коли попит на мережу постійно змінюється, оскільки користувачі приєднуються та залишають випадковим чином.

Таблиця 3.13 – Конфігурація середовища спортивного залу з туманом

Name	Description
Number applications	1
Number of micro-services	3
Number of locations	5
Number of nodes	15
The SFC structure	$a_1 : s_1 \rightarrow s_2 \rightarrow s_3$
Max. replication factor	5
Action space	91 actions
Observation space	6 observation Metrics
Reduced observation space	864 states
Episode duration	100 steps
Agent explore/learning rate	0.01/0.001

Простір дій складається з 91 дії (3 мікросервіси, 15 вузлів), тоді як простір спостереження складається з 6 показників спостереження, які були зведені до 864 дискретних станів. Тривалість кожного епізоду складається зі 100 кроків. Швидкість дослідження та швидкість навчання агента встановлено на рівні 0,01 і 0,001 відповідно. Для оцінювання обчислення агента та MILP було виконано на 6-ядерному процесорі Intel i7-9850H з частотою 2,6 ГГц і 16 ГБ пам'яті.

Як перша оцінка агент Q-Learning був навчений протягом 10 000 епізодів шляхом розгляду статичного варіанта використання, коли кількість запитів користувачів залишалася постійною. Таким чином, динамічних змін щодо запитів користувачів у цьому експерименті не очікується. Агент Q-навчання має мати можливість вивчати адекватні дії значно швидше в цьому випадку використання, ніж у динамічному сценарії, оскільки кількість запитів є постійною протягом усього навчання. На рисунку 3.12 показано накопичену винагороду та різницю у вартості між агентом Q-навчання та моделлю MILP.

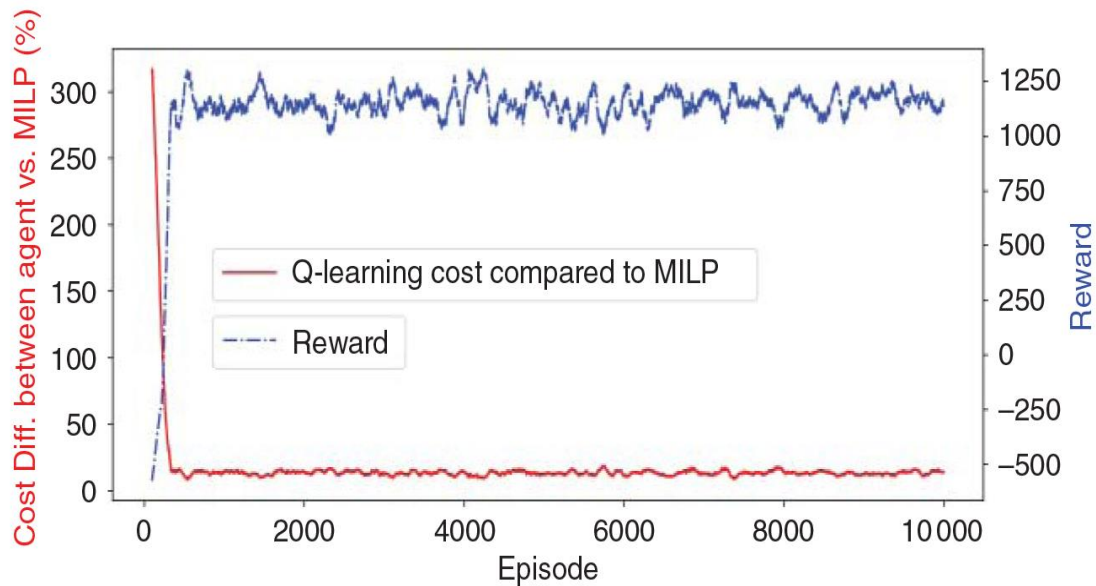


Рисунок 3.12 – Накопичена винагорода та різниця у вартості для статичного варіанта використання

Вікно згладжування 100 було застосовано для зменшення стрибків на обох кривих. Як показано, агент може знизити загальну вартість системи, досягаючи на 5% гіршу, ніж модель MILP.

Крім того, за один епізод було отримано 1200 винагород, що означає, що агент отримує в середньому 12 винагород за крок, що, виходячи з реалізованої нами функції винагороди, означає, що агент розподіляє всі потрібні екземпляри мікросервісу в інфраструктури, хоча вона не здатна повністю оптимізувати загальну вартість системи, як модель MILP. Крім того, ще одним важливим фактором, який потрібно враховувати, є відсоток прийнятих запитів у кожному епізоді, показаному на рисунку 3.13, оскільки агент може досягти низьких витрат, не розподіляючи всі мікросервіси, що призведе до неприйнятих запитів користувачів. Перші 10 кроків не враховуються (щодо прийняття запитів) як період розминки, гарантуючи, що агент має достатньо кроків для правильного вибору дій. Як показано, агент Q-навчання може приймати всі запити користувачів (тобто 100%) послідовно після 500 епізодів.

Нарешті, на рисунку 3.14 наведено час виконання кожного епізоду. Агент Q-навчання розв'язує один епізод у середньому за 0,15 і 0,20 секунди, що значно швидше порівняно з обчисленнями на основі ILP, оскільки ILP потребує розрахунку оптимальної схеми розподілу на кожному кроці епізоду. Результати показують, що Q-навчальний агент не тільки здатний вивчати схеми розподілу з низькими витратами, але й приймати всі запити користувачів для статичного сценарію.

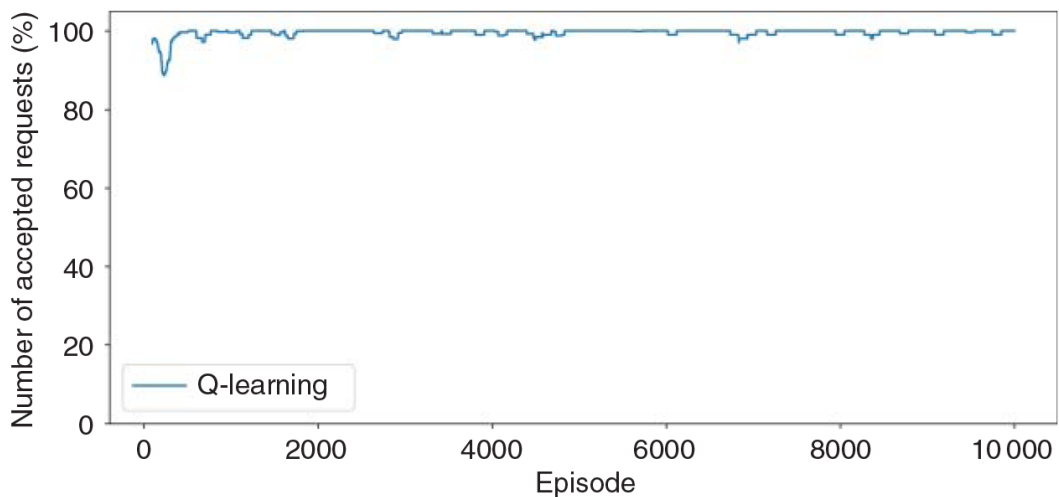


Рисунок 3.13 – Відсоток прийнятих запитів для статичного сценарію

У динамічному сценарії попит на мережу постійно змінюється протягом епізоду. Кількість запитів користувачів може зменшуватися або збільшуватися, і агент має адаптувати свою схему розподілу відповідно до вимог мережі. Кількість запитів користувача змінено кожні 5 кроків від 1 до 50 на основі конкретних ймовірностей (збільшення: 50%, рівне: 35%, зменшення: 15%).

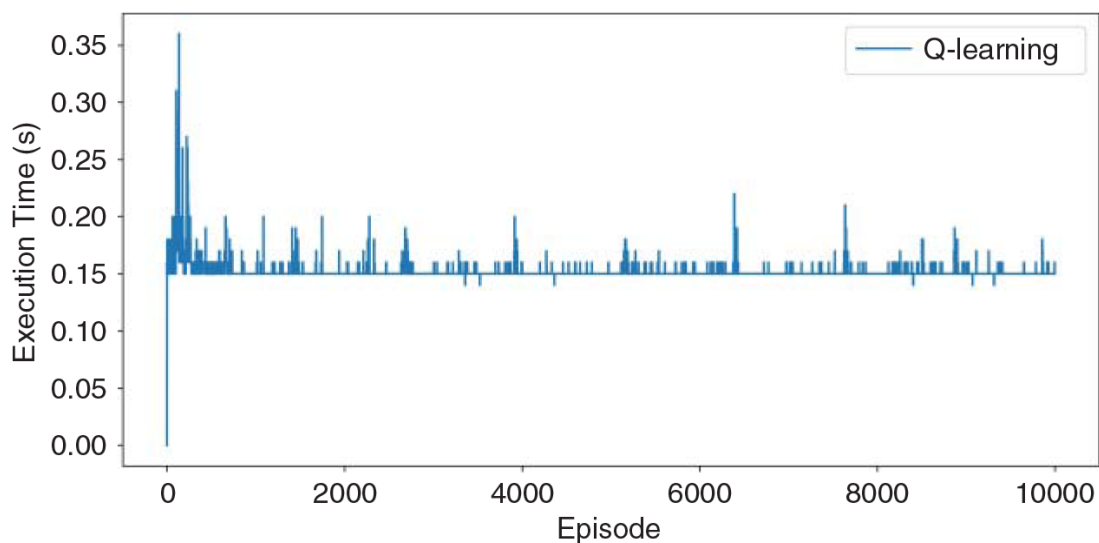


Рисунок 3.14 – Час виконання кожного епізоду

Загальне збільшення або зменшення є випадковим, що робить цей динамічний сценарій більш складним, ніж попередній статичний випадок, оскільки протягом усього експерименту агенту не надається шаблон, оскільки кілька шаблонів виникають у різних епізодах. У таблиці 3.14 показано час виконання MILP для кожної конфігурації. Наприклад, для

запитів користувачів, що перевищують 25, моделі MILP потрібно щонайменше п'ять секунд, щоб отримати оптимальну схему розподілу.

Таблиця 3.14 – Час виконання моделі MILP

	Number user requests							
	1	5	10	20	25	30	40	50
Execution time (s)	0.05	0.12	0.20	0.27	5.09	5.95	9.33	48.83

Для навіть більш високих значень запитів користувачів модель MILP займає в середньому не менше 10 секунд. Ці обчислення являють собою один крок в епізоді, що доводить недолік рішень на основі ІЛР, оскільки кожна зміна в мережі потребуватиме нового обчислення, що робить ці рішення непрактичними. На рисунку 3.15 показано як накопичену винагороду, так і середню різницю у вартості між агентом Q-навчання та моделлю MILP. Агент Q-Learning може знизити загальну вартість системи, досягаючи рішень на 50% гірших, ніж модель MILP. Крім того, агент накопичує лише 300 винагород за один епізод, тобто агент отримує в середньому 3 винагороди за крок.

Виходячи з реалізованої нами функції винагороди, це означає, що агент не може виділити всі потрібні екземпляри мікросервісу в інфраструктурі, що впливає на відсоток прийнятих запитів, як показано на рисунку 3.16. Через динамічний попит агенту потрібно постійно адаптувати схему розподілу в інфраструктурі, що перетворюється на схеми недостатнього та надлишкового забезпечення протягом кількох кроків в одному епізоді.

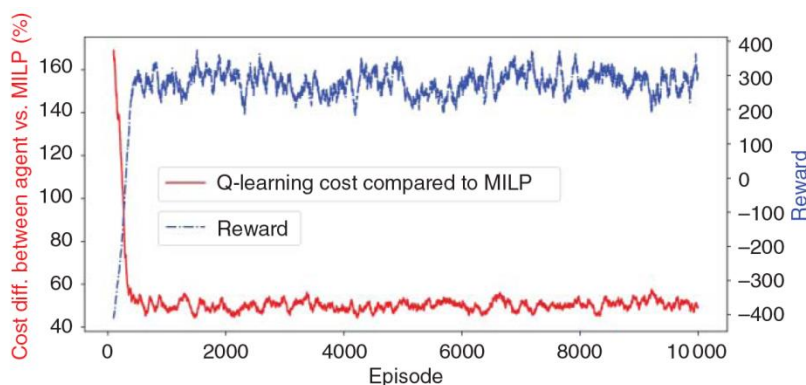


Рисунок 3.15 – Накопичена винагорода та різниця у вартості для динамічного випадку

Таким чином, прийняття запитів коливається між 40% і 90% під час навчання з 10 000 епізодів, коли застосовується вікно згладжування 10. Агент постійно реагує на зміни попиту, що робить цей сценарій значно складнішим, ніж попередній статичний варіант використання. Результати показують, що ефективні рішення для динамічного розподілу ресурсів все

ще відсутні через складність проблеми. Важко знайти практичні підходи, які б задовольнили потреби користувачів і при цьому знизили загальну вартість системи. Тим не менш, наші перші результати показують, що RL можна застосувати до розподілу SFC у FC і його потрібно додатково вивчити в майбутніх дослідженнях. Розширення простору для спостереження залишено для майбутньої роботи, оскільки це може покращити отримані результати щодо вартості та прийнятності.

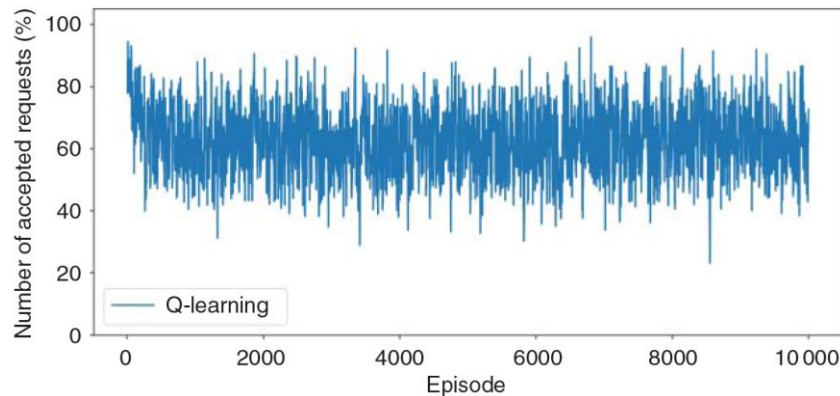


Рисунок 3.16 – Відсоток прийнятих запитів для динамічного сценарію

За останні роки методи ML стали цікавою галуззю досліджень у сфері мереж. Було зроблено кілька зусиль, щоб адаптувати методи ML до типових мережних проблем. Цей розділ присвячено агентам RL, щоб забезпечити ефективне рішення для розподілу SFC в інфраструктурах туманно-хмарних технологій. Надання ресурсів вивчалось протягом багатьох років у сфері керування мережею.

Однак мережі та послуги постійно розвиваються, з'являються нові протоколи та технології для вирішення поточних проблем і покращення загальної якості обслуговування. Останні приклади містять прийняття SFC, парадигми мікросервісів і FC. Сервіси підключаються в певному порядку для підвищення гнучкості та ефективності розподілу ресурсів. Крім того, шаблон мікросервісу революціонував спосіб, у який розробники зараз створюють свої програмні додатки. Додаток перетворився з єдиного монолітного в набір невеликих контейнерів, які можуть бути розгорнуті на кількох серверах. Таким чином, традиційні централізовані хмари перетворилися на невеликі розподілені туманні місця для розподілу обчислювальних ресурсів у зоні мережі. І коли всі ці концепції стають на місце, розподіл ресурсів стає досить складним онлайн-завданням. Дослідження забезпечення ресурсами в туманно-хмарних інфраструктурах все ще знаходяться на ранніх стадіях. Розповсюдження інфраструктури збільшило експлуатаційні витрати для постачальників послуг, а споживання енергії стало все більшою проблемою. Ми описали вирішення цієї проблеми (з залученням агентів RL для пошуку правильних рішень щодо розподілу,

спрямованих на зниження загальної вартості системи) в даному розділі. Середовище під назвою gym-fog було розроблено, щоб подолати розрив між рішеннями на основі ILP і алгоритмами RL. Для проблеми розподілу ресурсів розроблено простори спостереження та дій, щоб навчити агентів RL розподіляти послуги в FC. Було створено систему винагороди, щоб стимулювати агентів RL вибирати відповідні дії для розподілу SFC, зосереджених на зниженні загальної вартості системи, що приводить до підвищення енергоефективності. Результати довели, що наш розроблений агент може отримати продуктивність, порівнянну з найсучаснішими формулами ILP для статичних випадків використання, де 100% запитів було прийнято з загальними витратами на 5% нижчими, ніж наша модель MILP. Навпаки, динамічні сценарії використання також довели свою складність, показавши, що практичних рішень, здатних зменшити загальну вартість і прийняти всі запити користувачів, досі немає. Наш агент може знизити витрати до 50% і прийняти в середньому 60% запитів.

Розробка систем RL, здатних навчатися безпосередньо на досвіді без будь-яких попередніх знань і здатних перерозподіляти послуги в інфраструктурі, реагуючи на раптові зміни мережі, стане наступною головною темою в цій галузі досліджень.

Методи RL вже довели свою потенційну застосовність до домену надання ресурсів. Однак продуктивність цих методів глибоко взаємопов'язана зі способом налаштування системи RL. Середовище є ключем до проблеми. Взаємодія між агентом і середовищем сильно впливає на продуктивність цих алгоритмів. Крім того, стан і простір дій проблеми можуть зростати експоненціально залежно від розміру інфраструктури (тобто кількості вузлів, кількості служб), яка використовується в середовищі, що може призвести до нерозв'язної проблеми. Нарешті, не потрібно нехтувати важливістю системи винагород. Агент навчиться правильно розподіляти послуги, лише якщо він буде компенсований позитивними винагородами в процесі навчання, навіть якщо він не зміг досягти бажаних рішень. Головне – давати агенту вищі винагороди, чим коли він ближче до досягнення кінцевої мети, інакше агенту буде досить складно навчитися правильних дій. Навпаки, методи на основі ILP важко реалізувати на практиці через час їх вирішення. Крім того, вони потребують багато початкової інформації, яка надходить в алгоритм, щоб можна було знайти оптимальні схеми розподілу. Ці методи можуть зайняти години або навіть дні, щоб знайти оптимальний розподіл послуг, і коли відбуваються зміни в мережі, перерозподіл послуг має бути зроблений якомога швидше. Іншою проблемою є відсутність досвіду як у сферах RL, так і в сфері розподілу ресурсів. Невелика кількість експертів мають фундаментальні знання в обох сферах, що ускладнює реалізацію рішень RL, адаптованих до проблем розподілу ресурсів. Більшість методів RL, які використовуються в мережах, була створена для інших типів програм (наприклад, відеоігор).

Підводячи підсумок, у сфері розподілу ресурсів залишається кілька проблем.

Тим не менш, враховуючи динамічну поведінку мережі та потребу в ефективних стратегіях планування (наприклад, енергоефективність, низька затримка), методи RL довели, що за достатнього навчання вони можуть бути адекватним рішенням для забезпечення ресурсами в туманно-хмарних інфраструктурах. Крім того, ці методи продемонстрували свій потенціал у практичних сценаріях, де поточні рішення на основі ILP мають кілька недоліків, особливо з точки зору масштабованості. У майбутньому ми розширимо наше середовище gym-fog, розробляючи складніші функції винагороди, здатні повністю вирішувати проблеми динамічних випадків використання, а також експериментуватимемо з різними агентами RL, такими як мережі глибокої черги та методи акторської критики.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Alpaydin, E. (2020). Introduction to Machine Learning, vol.4. MIT Press. ISBN 9780262043793.
2. Hardegen, C., Pfulb, B., Rieger, S., and Gepperth, A. (2020). Predicting network flow characteristics using deep learning and real-world network traffic. *IEEE Transactions on Network and Service Management* 17 (4): 2662–2676.
3. Khatouni, A.S., Seddigh, N., Nandy, B., and Zincir-Heywood, N. (2021). Machine learning based classification accuracy of encrypted service channels: analysis of various factors. *Journal of Network and Systems Management* 29 (1): 1–27.
4. Elsayed, M., Erol-Kantarci, M., Kantarci, B. et al. (2020). Low-latency communications for community resilience microgrids: a reinforcement learning approach. *IEEE Transactions on Smart Grid* 11 (2): 1091–1099. <https://doi.org/10.1109/TSG.2019.2931753>.
5. Wilkins, Z. and Zincir-Heywood, N. (2020). COUGAR: clustering of unknown malware using genetic algorithm routines. In: *GECCO '20: Genetic and Evolutionary Computation Conference, Cancún Mexico (July 8-12, 2020)* (ed. C.A.C. Coello), 1195–1203. ACM. <https://doi.org/10.1145/3377930.3390151>.
6. Dietz, C., Dreo, G., Sperotto, A., and Pras, A. (2020). Towards adversarial resilience in proactive detection of botnet domain names by using MTD. *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, pp. 1–5. <https://doi.org/10.1109/NOMS47738.2020.9110332>.
7. Le, D.C., Zincir-Heywood, N., and Heywood, M. (2020). Analyzing data granularity levels for insider threat detection using machine learning. *IEEE Transactions on Network and Service Management* 17 (1): 30–44.
8. Bag, T., Garg, S., Rojas, D.F.P., and Mitschele-Thiel, A. (2020). Machine learning-based recommender systems to achieve self-coordination between son functions. *IEEE Transactions on Network and Service Management* 17 (4): 2131–2144. <https://doi.org/10.1109/TNSM.2020.3024895>.
9. Renga D., Apiletti D., Giordano D. et al. (2020). Data-driven exploratory models of an electric distribution network for fault prediction and diagnosis. *Computing* 102 (5): 1199–1211. <https://doi.org/10.1007/s00607-019-00781-w>.
10. Steenwinckel, B., Paepe, D.D., Hautte, S.V. et al. (2021). FLAGS: a methodology for adaptive anomaly detection and root cause analysis on sensor data streams by fusing expert knowledge with machine learning. *Future Generation Computer Systems* 116: 30–48. <https://doi.org/10.1016/j.future.2020.10.015>.
11. Park, S., Kim, H., Hong, J. et al. (2020). Machine learning-based optimal VNF deployment. *21st Asia-Pacific Network Operations and Management Symposium, APNOMS 2020, Daegu, South Korea (22–25 September 2020)*, IEEE, pp. 67–72. <https://doi.org/10.23919/APNOMS50412.2020.9236970>.

12. Zhang, Q., Wang, X., Lv, J., and Huang, M. (2020). Intelligent content-aware traffic engineering for SDN: an Ai-driven approach. *IEEE Network* 34 (3): 186–193. <https://doi.org/10.1109/MNET.001.1900340>.
13. Zhang, J., Ye, M., Guo, Z. et al. (2020). CFR-RL: traffic engineering with reinforcement learning in SDN. *IEEE Journal on Selected Areas in Communications* 38 (10): 2249–2259. <https://doi.org/10.1109/JSAC.2020.3000371>.
14. Le, D.C. and Zincir-Heywood, N. (2020). A frontier: dependable, reliable and secure machine learning for network/system management. *Journal of Network and Systems Management* 28 (4): 827–849.
15. Alam, I., Sharif, K., Li, F. et al. (2020). A survey of network virtualization techniques for internet of things using SDN and NFV. *ACM Computing Surveys (CSUR)* 53 (2): 1–40.
16. Yilma, G.M., Yousaf, Z.F., Sciancalepore, V., and Costa-Perez, X. (2020). Benchmarking open source NFV MANO systems: OSM and ONAP. *Computer Communications*. 161 86–98.
17. Rodriguez, V.Q., Guillemin, F., and Boubendir, A. (2020). 5G E2E network slicing management with ONAP. *Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, IEEE, pp. 87–94.
18. Rodriguez, V.Q., Guillemin, F., and Boubendir, A. (2020). Network slice management on top of ONAP. *IFIP/IEEE Network Operations and Management Symposium (NOMS)*, IEEE, pp. 1–2.
19. Yao, H., Ma, S., Wang, J. et al. (2020). A continuous-decision virtual network embedding scheme relying on reinforcement learning. *IEEE Transactions on Network and Service Management*. 17 (2): 864–875
20. Yan, Z., Ge, J., Wu, Y. et al. (2020). Automatic virtual network embedding: a deep reinforcement learning approach with graph convolutional networks. *IEEE Journal on Selected Areas in Communications* 38 (6): 1040–1057.
21. Bunyakitanon, M., Vasilakos, X., Nejabati, R., and Simeonidou, D. (2020). End-to-end performance-based autonomous VNF placement with adopted reinforcement learning. *IEEE Transactions on Cognitive Communications and Networking*. 6 (2): 534–547
22. Pawlicki, M., Choras', M., and Kozik, R. (2020). Defending network intrusion detection systems against adversarial evasion attacks. *Future Generation Computer Systems*. 110 148–154.
23. Musumeci, F., Ionata, V., Paolucci, F. et al. (2020). Machine-learning-assisted DDoS attack detection with P4 language. *IEEE International Conference on Communications (ICC)*, pp. 1–6.
24. Abiko, Y., Saito, T., Ikeda, D. et al. (2020). Flexible resource block allocation to multiple slices for radio access network slicing using deep reinforcement learning. *IEEE Access* 8: 68183–68198.

25. Pei, J., Hong, P., Pan, M. et al. (2020). Optimal VNF placement via deep reinforcement learning in SDN/NFV-enabled networks. *IEEE Journal on Selected Areas in Communications* 38 (2): 263–278.
26. Han, S., Jang, S., Choi, H. et al. (2020). Virtualization in programmable data plane: a survey and open challenges. *IEEE Open Journal of the Communications Society* 1: 527–534.
27. Benzaid, C. and Taleb, T. (2020). Ai-driven zero touch network and service management in 5G and beyond: challenges and research directions. *IEEE Network* 34 (2): 186–194. <https://doi.org/10.1109/MNET.001.1900252>.
28. Ramanan, P. and Nakayama, K. (2020). BAFFLE: blockchain based aggregator free federated learning. 2020 IEEE International Conference on Blockchain (Blockchain 2020), November 2020, IEEE, pp. 72–81.
29. Antevski, K. and Bernardos, C.J. (2020) Federation of 5G services using distributed ledger technologies. *Wiley Internet Technology Letters* 1–6. <https://doi.org/10.1002/itl2.193>.
30. Antevski, K., Martín-Pérez, J., Garcia-Saavedra, A. et al. (2020). A Q-learning strategy for federation of 5G services. 2020 IEEE International Conference on Communications (ICC 2020), June 2020, IEEE, pp. 1–6.
31. Schwarzmann, S., Marquezan, C.C., Trivisonno, R. et al. (2020). Accuracy vs. cost trade-off for machine learning based QoE estimation in 5G networks. *IEEE International Conference on Communications: Next-Generation Networking and Internet Symposium (IEEE ICC NGNI Symposium)*, Dublin, Ireland, June 2020, pp. 1–6.
32. Sieber, C., Schwarzmann, S., Blenk, A. et al. (2020). Scalable application- and user-aware resource allocation in enterprise networks using end-host pacing. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems* 5 (3): 1–41.
33. Khalili, A., Akhlaghi, S., Tabassum, H., and Ng, D.W.K. (2020). Joint user association and resource allocation in the uplink of heterogeneous networks. *IEEE Wireless Communications Letters* 9 (6): 804–808.
34. Ibrahim, H., Tabassum, H., and Nguyen, U.T. (2020). The meta distributions of the SIR/SNR and data rate in coexisting Sub-6GHz and millimeter-wave cellular networks. *IEEE Open Journal of the Communications Society* 1: 1213–1229.
35. Sayehvand, J. and Tabassum, H. (2020). Interference and coverage analysis in coexisting RF and dense terahertz wireless networks. *IEEE Wireless Communications Letters* 9 (10): 1738–1742.
36. Hussain, F., Hussain, R., Hassan, S., and Hossain, E. (2020). Machine learning for resource management in cellular and IoT networks: potentials, current solutions and open challenges. *IEEE Communication Surveys and Tutorials* 22 1–26.
37. Chen, M., Semiari, O., Saad, W. et al. (2020). Federated echo state learning for minimizing breaks in presence in wireless virtual reality networks. *IEEE Transactions on Wireless Communications* 19 (1): 177–191.

38. Shafin, R., Liu, L., Chandrasekhar, V. et al. (2020). Artificial intelligence-enabled cellular networks: a critical path to beyond-5G and 6G. *IEEE Wireless Communications* 27 (2): 212–217.
39. Goethals, T., Volckaert, B., and De Turck, F. (2020). Adaptive fog service placement for real-time topology changes in kubernetes clusters. *CLOSER2020, The 10th International Conference on Cloud Computing and Services Science*, pp. 161–170.
40. Гупта, Р. і Рафгарден, Т. (2020). Проектування керованого даними алгоритму. *Повідомлення ACM* 63 (6): 87–94.
41. Bengio, Y., Lodi, A., and Prouvost, A. (2021). Machine learning for combinatorial optimization: a methodological tour d’horizon. *European Journal of Operational Research* 290 (2): 405–421.
42. Zhang, D., Yin, J., Zhu, X., and Zhang, C. (2020). Network representation learning: a survey. *IEEE Transactions on Big Data* 6 (1): 3–28.
43. Herodotou, H., Chen, Y., and Lu, J. (2020). A survey on automatic parameter tuning for big data processing systems. *ACM Computing Surveys* 53 (2): 1–37.
44. Alnafessah, A.S. and Casale, G. (2020). TRACK: optimizing artificial neural networks for anomaly detection in Spark Streaming systems. *Proceedings of ACM VALUETOOLS ’20*, pp. 188–191.
45. Ni, X., Li, J., Yu, M. et al. (2020). Generalizable resource allocation in stream processing via deep reinforcement learning. *AAAI* 34 (01): 857–864.
46. Abbasloo, S., Yen, C.-Y., and Chao, H.J. (2020) Classic meets modern: a pragmatic learning-based congestion control for the internet. *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*.pp. 632–647.

*Навчальне електронне видання
комбінованого використання.
Можна використовувати в локальному та мережному режимах*

**Микола Володимирович Васильківський
Геннадій Григорович Бортник
Василь Мартинович Кичак**

Програмні технології в інфокомунікаційних системах. Навчальний посібник для студентів спеціальності 172 «Телекомунікації та радіотехніка»

Навчальний посібник

Рукопис оформлено М. Васильківським

Редактор В. Дружиніна

Оригінал-макет виготовлено Т. Старічек

Підписано до видання 13.02.2023 р.
Гарнітура Times New Roman.
Зам. № P2023-014.

Видавець та виготовлювач
Вінницький національний технічний університет,
Редакційно-видавничий відділ.
ВНТУ, ГНК, к. 114.
Хмельницьке шосе, 95,
м. Вінниця, 21021.
Тел. (0432) 65-18-06.
press.vntu.edu.ua;
Email: irvc.vntu@gmail.com

Свідоцтво суб'єкта видавничої справи
серія ДК № 351 від 01.07.2009 р.