ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: **«МЕТОДИКА ПОБУДОВИ МЕРЕЖ ЗВ'ЯЗКУ ІЗ ЗАСТОСУВАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ»**

на здобуття освітнього ступеня магістр
за спеціальністю 123 Комп'ютерна інженерія
(код,найменування спеціальності)
освітньо-професійної програми Комп'ютерні системи та мережі
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ Денис БОДОРАЦЬКИЙ
(підпис) *(ім'я, ПРІЗВИЩЕ здобувача)*

Виконав: здобувач вищої освіти гр.КСДМ-61
Денис БОДОРАЦЬКИЙ
(ім'я, ПРІЗВИЩЕ)

Керівник: Наталія ЛАЩЕВСЬКА
К.т.н.,доцент *(ім'я, ПРІЗВИЩЕ)*

Рецензент: _____
науковий ступінь, вчене звання *(ім'я, ПРІЗВИЩЕ)*

Київ 2024

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Комп'ютерної інженерії
Ступінь вищої освіти «Магістр»
Спеціальність 123 Комп'ютерна інженерія
Освітньо-професійна прорама Комп'ютерні системи та мережі

ЗАТВЕРДЖУЮ
Завідувач кафедрою Комп'ютерної інженерії
Наталія ЛАЩЕВСЬКА
(ім'я, ПРІЗВИЩЕ)
"____" _____2024 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Бодорацькому Денису Анатолійовичу
(прізвище, ім'я, по батькові здобувача)

1.Тема кваліфікаційної роботи: Методика побудови мереж зв'язку із застосуванням штучного інтелекту
керівник роботи Наталія ЛАЩЕВСЬКА, к.т.н., доцент
(ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від 15.10 2024 р. № 320
2. Строк подання кваліфікаційної роботи 29.12.2024
3. Вихідні дані кваліфікаційної роботи:
Сучасні мережі зв'язку;
Програмно-визначена мережа;
Методи контролю та визначення трафіку;
Науково-технічна література.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
1. Огляд сучасних мереж зв'язку, стан та перспективи їх подальшого розвитку;
2. Дослідження методики розпізнання трафіка послуг в програмно-конфігуровахих мережах;
3. Контроль та прогрнозування навантаження складових мережі в умовах обмеження якості обслуговування послуг.

5.Перелік ілюстраційного матеріалу: *презентація*

6. Дата видачі завдання “ 01 ” вересня 2024р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Підбір технічної літератури	01.09.24 – 15.09.24	
2.	Огляд сучасних мереж зв’язку, стан та перспективи їх подальшого розвитку.	16.09.24 – 30.09.24	
3.	Аналіз методики розпізнання трафіка послуг в програмно-конфігуровахих мережах.	01.10.24 – 20.10.24	
4.	Контроль та прогрнозування навантаження складових мережі в умовах обмеження якості обслуговування послуг	21.10.24 – 10.11.24	
5.	Розробка демонстраційних матеріалів, доповідь.	11.11.24 – 30.11.24	
6.	Оформлення магістерської роботи	01.12.24 – 20.12.24	

Здобувач вищої освіти _____ Денис БОДОРАЦЬКИЙ
(підпис) *(ім’я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Наталія ЛАЩЕВСЬКА
(підпис) *(ім’я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття ступеня магістр: 73 стор., 27 рис., 5 табл., 28 джерел.

Мета роботи – реалізація методів побудови мережі та надання послуг із застосуванням штучного інтелекту.

Об'єкт дослідження – процес побудови мереж зв'язку та послуг із застосуванням штучного інтелекту.

Предмет дослідження – сучасні мережі зв'язку.

Короткий зміст роботи: Виконуючи поставлені задачі, у кваліфікаційній магістерській роботі проаналізовано концепцію сучасного стану та подальші перспективи розвитку мережі зв'язку. Проаналізовано сучасні технології та методи побудови мережі та систем надання послуг, які задають напрямок розвитку мереж та послуг у бік їх децентралізації. Враховуючи можливість SDN та функціональну можливість протоколу управління OpenFlow, було взято метод ідентифікації трафіку в мережах зв'язку на основі метаданих потоків та штучної нейронної мережі. Проведення дослідження обраного методу включає реалізацію програми контролера SDN та його тестування.

У третьому розділ було виконано дослідження сладності моніторингу контролеру SDN. Результат показує необхідність проведень моніторингу та прогнозу навантаження на контролер SDN з метою забезпечення стійкості систем в умовах жорстких рамок якості обслуговування послуг.

КЛЮЧОВІ СЛОВА: ШТУЧНИЙ ІНТЕЛЕКТ, ІНФРАСТРУКТУРА, МЕРЕЖА, ПОСЛУГИ, ПРОГНОЗ, ПРОТОКОЛ, МЕТОДИКА, КОНТРОЛЬ, ПРОГРАМНО-КОНФІГУРОВАНА МЕРЕЖА.

ABSTRACT

The text part of the qualification work for obtaining a master's degree: 73 pages, 27 figures, 5 tables, 28 sources.

The purpose of the work is to implement methods of building a network and providing services using artificial intelligence.

The object of research is the process of building communication networks and services using artificial intelligence.

The subject of research is modern communication networks.

Brief content of the work: Fulfilling the set tasks, the concept of the current state and further prospects for the development of the communication network were analyzed in the qualifying master's thesis. Modern technologies and methods of building networks and service delivery systems are analyzed, which set the direction of development of networks and services towards their decentralization. Taking into account the possibility of SDN and the functionality of the OpenFlow control protocol, a method of traffic identification in communication networks based on flow metadata and an artificial neural network was taken. Conducting a study of the chosen method includes the implementation of the SDN controller program and its testing.

In the third section, a study of the sweetness of SDN controller monitoring was performed. The result shows the need for monitoring and forecasting the load on the SDN controller in order to ensure the stability of the systems in the conditions of the strict framework of the quality of service.

KEY WORDS: ARTIFICIAL INTELLIGENCE, INFRASTRUCTURE, NETWORK, SERVICES, FORECAST, PROTOCOL, METHODOLOGY, CONTROL, SOFTWARE-CONFIGURED NETWORK.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

АПК - Апаратно-програмний комплекс;

БД - База даних;

ІКТ - Інфокомунікаційні технології;

ІТ- Інформаційні технології;

ОС - Операційна система;

ПЗ - Програмне забезпечення;

СУБД - Системи управління базами даних;

ТІ - Тактильний інтернет;

ШНМ - Штучна нейронна мережа;

AI - Artificial Intelligence;

API - Application Programming Interface;

CAPEX - Capital Expenditure;

D2D - Device-to-Device Communications;

DB - Data Base;

DPI - Deep Packet Inspection;

ETSI - European Telecommunications Standards Institute;

GW - Gateway;

ISC - Internal System Code;

IoT - Internet of Things;

IoTDM - Internet of Things Data Management;

IaaS - Infrastructure as a Service;

IP - Internet Protocol;

KPI - Key Performance Indicators;

NACF - Network Access Control Function;

NB-API - North Bound API;

NFV - Network Function Virtualization;

OPEX - Operating Expenditure;

SDN - Software-Defined Networking.

ВСТУП

Актуальність теми. Сучасний світ інформаційних технологій еволюціонує швидкими темпами, ще не встигаючи вичерпати свій потенціал як на ринку зв’язку з’являються вже нові технології. Мережі сучасного зв'язку а також послуги які вони мають можливість надавати є особливою і актуальною темою, якій приділяється валика увага в останні роки. Отриані результати досліджень та розробки які мають світовий масштаб стали результатом переходу до концепцій мережі зв'язку 2030.

Такий перехід здійснився у зв'язку з величезною кількістю новітнього типу послуг, які з'являються і засновані на концепціях IoT. МСЄ звертає увагу на три основні напрямки розвитку сучасних мереж зв’язку, серед яких: масштабованаміжмашинна взаємодія, високомобільні зв'язоки і мережа зв'язків які мабть ультрамалі затримки.

Сокупність запропонованої концепції послуг в даному напрямку вже поступово переходять на слідуюче покоління мережі зв'язку. Слід зазначити, що такі критерії як надщільність мережі додає додаткові вимоги саме до методу реалізації мережевої інфраструктури. Надщільність це одна з ознак не тільки мережі IMT-2020, але й усіх послідуючих.

Аналіз показав, що велику частину досліджень складають роботи з мережі 5G, нових послуг, безпроводових та mesh-мереж, технології IoT. Але питанню аналізу ШІ в мережі зв'язку є новою темою і наразі їй не приділено необхідної уваги.В межах даного напряму виділяється ціла низка завдань, важливим з яких є задача виконання ідентифікацій трафіків у мережах, здійсненя прогнозування навантажень на мережі, а також можливі простої. Високорівневою задачею також є ефективний розподіл обчислювального ресурсу. Ці задачі вимагають застосування певних методів на базі технологій ШІ, враховуючи вимоги показників якості обслуговування сучасних послуг мереж, потрібної швидкодії службового сервісу та знання прогнозуючих даних.

Мета роботи – реалізація методів побудови мережі та надання послуг із застосуванням штучного інтелекту.

Для виконання поставленої мети, у магістерській роботі розроблено та виконано наступні завдання:

– огляд сучасних мереж зв’язку, стан та перспективи їх подальшого розвитку;

– дослідження методики розпізнання трафіка послуг в програмно-конфігурованих мережах;

– контроль та прогрнозування навантаження складових мережі в умовах обмеження якості обслуговування послуг.

Об’єкт дослідження – процес побудови мереж зв’язку та послуг із застосуванням штучного інтелекту.

Предмет дослідження – сучасні мережі зв’язку.

Методи дослідження. Для виконання поставлених задач у магістерській роботі використано методи машинного навчання, теорії оптимізації, імітаційного та емуляційного моделювання.

Наукова новизна. Досліджено й реалізовано метод визначення трафіку послуг у мережі зв’язку, який дозволяє усунути додаткові затримки та зберегти структуру потоків без змін. Такий підхід сприяє підвищенню ефективності обробки даних, мінімізуючи вплив на якість обслуговування користувачів.

Практична значущість отриманих результатів. Практична значущість дослідження полягає, насамперед, в обґрунтуванні необхідності децентралізації мережі та обчислювальних систем обробки даних, а також у реалізації нових послуг. Окремо виділяється важливість модернізації архітектури програмного забезпечення через використання мікросервісних підходів, що дозволяє створювати мережі з ультранизькими затримками для підвищення ефективності й швидкості обробки інформації в реальному часі.

Цей варіант підкреслює практичне значення модифікації мережевої архітектури та покращення ефективності через застосування новітніх технологій.

Апробація результатів та публікації. Результати магістерської роботи доповідались на двох науково-технічних конференціях. По тематиці роботи публікована одна стаття в науковому журналі.

РОЗДІЛ 1 ДОСЛІДЖЕННЯ СУЧАСНИХ МЕРЕЖ ЗВ'ЯЗКУ, СТАН ТА ПЕРСПЕКТИВИ ЇХ ПОДАЛЬШОГО РОЗВИТКУ

1.1 Огляд концепції мереж зв'язку IMT-2020

Соціально-технічна еволюція останніх десятиліть, значною мірою стимульована розвитком технологій безпроводового зв'язку, внесла істотний внесок у економічний та соціальний розвиток як розвинених, так і країн. Безпроводовий зв'язок є невід'ємною часткою повсякденних потреб у житті людей. Згідно з прогнозами, соціально-технічні тенденції та розвиток систем рухомого зв'язку будуть, як і раніше, тісно пов'язані один з одним і сформують основу суспільства починаючи з 2020 року. Однак, як передбачають фахівці, у майбутньому з'являться нові вимоги, пов'язані, зокрема, зі збільшенням обсягу трафіку, появою нових пристроїв із різними експлуатаційними нормативами, високою оцінкою користувачем якості послуг (QoE), доступних за ціною завдяки подальшому зниженню витрат. У зв'язку з цим необхідне використання дедалі більшої кількості інноваційних технічних рішень.

IMT-2020 є гетерогенними, що єднают у собі велику кількість різного роду мережі, як традиційні фіксовані мережі зв'язку та безпроводові, до мереж із можливістю літати а також сенсорні мнрнжі [1]. В закордонних дослідженнях такі мережі мають назву Het_Net [2]. Але спершу можливість гетерогенності була поміченою у взаємодіях системи довгої еволюції та сенсорної мережі. Стандарти IMT-2020 є не тільки пристрої зв'язку для клієнтів, але і засіб, що надає можливість розвитку іншої галузі, якприклад медицини, транспортної, освітньої. МСЕ-Т вказується у рекомендаціях МСЕ-Р М.2083-0 наступні сценарії застосування IMT-2020 [5]:

- удосконалення безпроводового широкосмугового зв'язоку. Зацікавленість у послугах безпроводового широкосмугового зв'язку має зростаючі тенденції і в перспективі.

- наднадійність у передаванні інформації при малій затримці. В даному випадку можна привести приклади, а саме: дистанційної хірургії, автоматизації розподілення електроенергії в «розумній електромережй», безпека транспорту, а саме автономному транспортному засобі.

- масштабні технології міжмашинного зв'язку. Такий напрямок характеризується чималою кількістю пристроїв які підключені на певну ділянку.

МСЕ вказали такі сценарії як основу трьох «китів» мережі зв'язку IMT-2020, у даному з векторів де відбувається індивідуальний розвиток, а також спільне з іншим вектором розвитку, що передбачає принесення досить чималих позитивних синергетичних ефектів. Результатом синергії таких напрямків еволюції мереж зв'язку можемо спостерігати в даний час: застосування безпілотних транспортних засобів, поява системи типів «Розумного міста» та багато іншого. Напрацювання таких рішень, є на початкових етапах реалізації, але відчути та оцінити ефект застосування є змога відчути нам зараз.

Технологія широкосмугового стільникового зв’язку наступного покоління 5G. Розширена швидкість передачі даних означає, що 5G може запропонувати найшвидкісні послуги широкосмугового стільникового зв’язку, які коли-небудь були, і надати альтернативу доступу «останньої милі», наприклад, з’єднання «оптоволокно до дому» (FTTH).

Відповідно до ITU-R M.2083-0 [6], служби та програми мереж можна згрупувати в три сценарії використання:

1. Розширений мобільний широкосмуговий зв’язок (eMBB): мережа має забезпечувати набагато швидший і надійніший стільниковий широкосмуговий зв’язок, пропонуючи користувачам більш багатий досвід використання програм.

2. Супер надійний зв’язок із малими затримками (URLLC): найвищими пріоритетами для цього сценарію використання є параметри затримки та мобільності.

3. Масивні комунікації машинного типу (mMTC): Цей сценарій використання описується дуже великим масштабом або масовими додатками Інтернету речей (IoT).

На рис.1.1 показано можливості IMT-2020, а на рис.1.2 показано прикади впровадження IMT-2020.

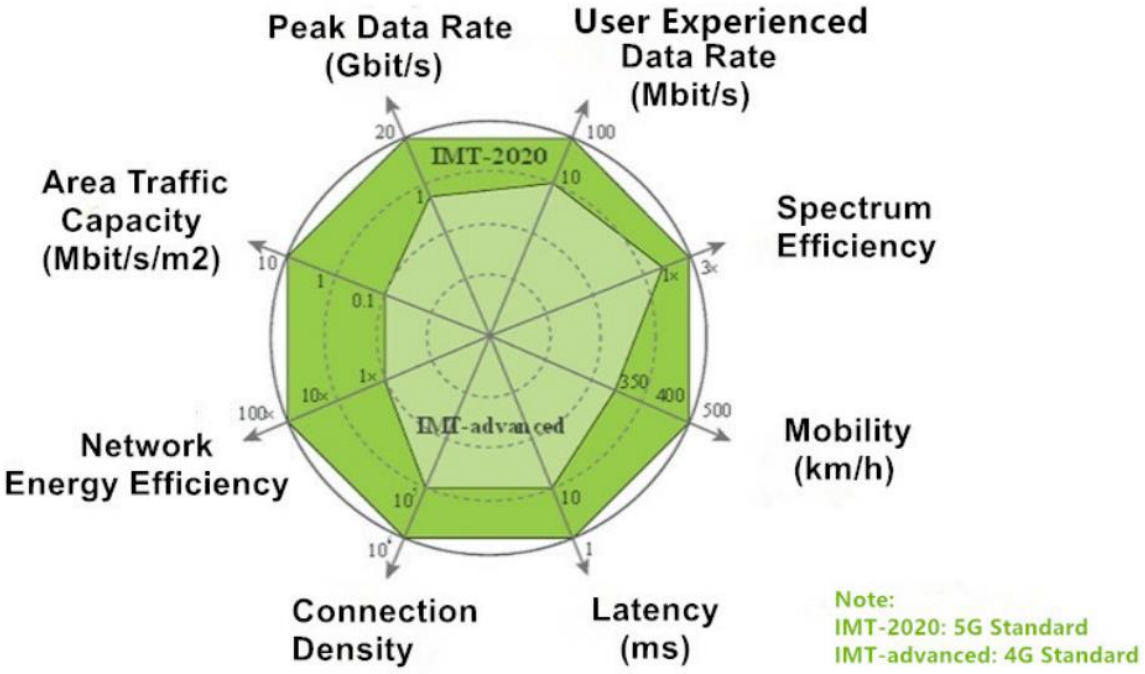


Рисунок 1.1 Ключові можливості від IMT-advanced до 5G

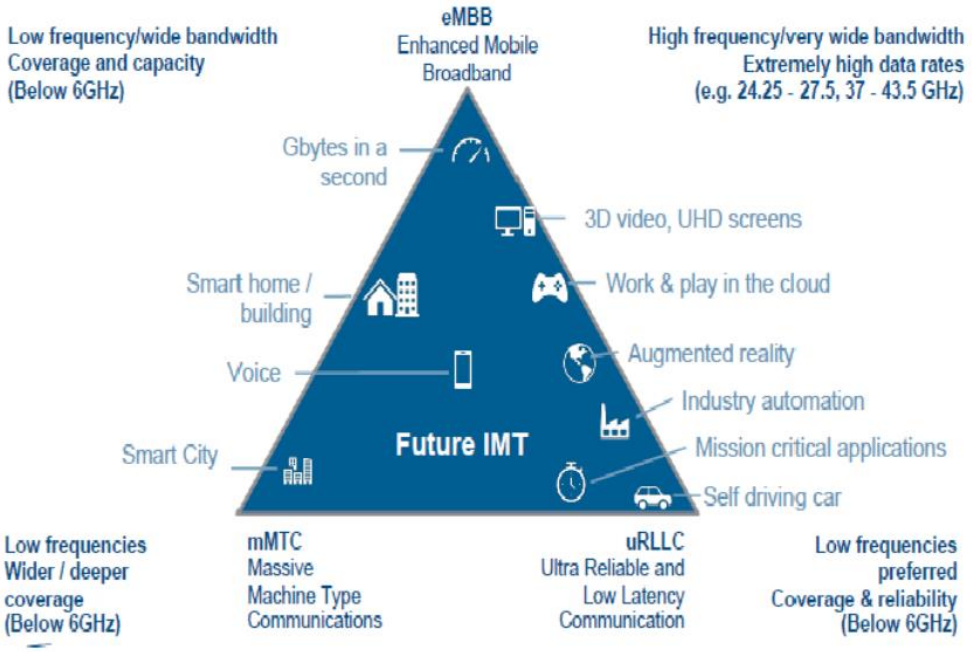


Рисунок 1.2 Сценарії застосування IMT-2020

Як основні мережі МСЕ у побажаннях [7] дає рекомендації використання концепцій SDN та Віртуалізації мережевих функцій.

Загальними засадами у проектуванні мереж IMT-2020 є можливість гнучкості, масштабованості і можливістю надання різноманітних послуг. До основних критеріїв, які мають ключові можливості мережі 5G можливо віднести [7]:

- максимальна швидкість для користувачів. Максимально-доступна швидкість передачі інформації в ідеальній умові для усих користувачів пристроїв у межах 10 Гбіт/с;
- повинні бути забезпечені мінімально можливі затримки (як приклад, для послуг Тактильного інт. має не перевищувати 1мс);
- можливість мобільності: є необхідним забезпечувати здійснення вимог QoS при високих швидкостях руху користувачів чи пристроїв;
- чимала щільність підключення засобів;
- вимоги енергоефективності: необхідність забезпечення енергоефективності на стороні клієнта, а також на стороні операторів;
- спроможність пропускання: Максимальна пропускна здатність мереж на одиницю простору. Як показав аналіз IMT-2020, передбачає застосування концепцій IoT, яка надає можливість збільшити кількість підключень, у результаті чого збільшується навантаження на пристрої в декілька раз.

У результі таких вимог, що ставляться перед мережою, і має на меті досягти виконання найнеобхідніших критеріїв для показників якості, які висуваються для нової послуги. Прикладом, якої є Тактильний інтернет, варто виконати опрацювання питань мігрування мережних технологій у такі рішення, які б змогли забезпечувати потрібну роботу мережевої інфраструктури, її масштабованості, модульності, високих абстракційних рівнів управління, які дають змогу реалізувати суцільну гнучку систему контролювання та управляючі дії із застосуванням стандартного програмного інтерфейсу.

1.2 Програмно-визначена мережа з використанням OpenFlow

Програмно-визначена мережа (SDN) привернула багато уваги останніми роками, оскільки вона усуває відсутність програмованості в існуючих мережевих архітектурах і забезпечує легшу та швидшу мережеву інновацію. SDN чітко відокремлює площину даних від площини керування та полегшує програмну реалізацію складних мережевих програм на вершині. Є надія на менш специфічне та дешевше обладнання, яким можна керувати програмними додатками через стандартизовані інтерфейси.

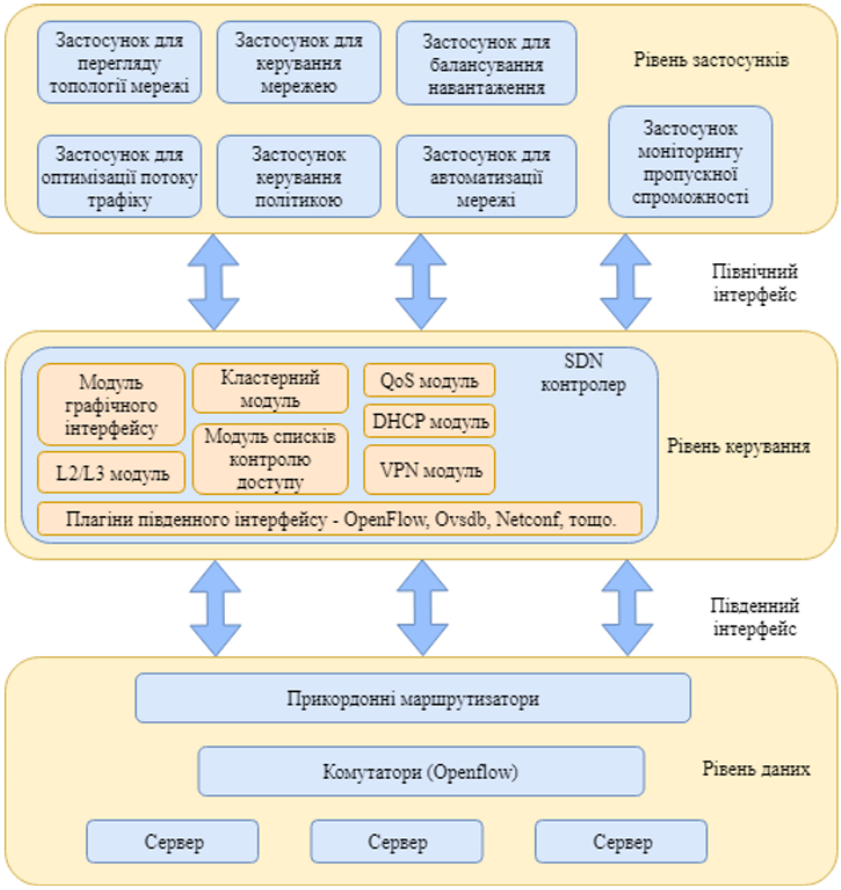


Рисунок 1.3 - Трирівнева програмно-визначена мережева архітектура (SDN)

Крім того, очікується більша гнучкість завдяки динамічному додаванню нових функцій до мережі у вигляді мережевих програм. Ця концепція відома з операційних систем мобільних телефонів, таких як iOS від Apple і Android від Google, де «програми» можна динамічно додавати до системи.

У даному підрозділі мною проаналізовано SDN та розглянуто можливі протоколи керування мережі. Ми відокремлюємо SDN від активних і програмованих мереж, які в деяких аспектах нагадують її. Мною проаналізовано SDN у відповідності визначенню, наданого ONF [4]. Це найбільш прийнятне визначення SDN у всьому світі, оскільки більшість глобальних гравців у мережевій індустрії та багато IT-корпорацій беруть участь у ONF.

На рис.1.3 показано структуру SDN, яка базується з трьох рівнів. Найнижчий шар рівень інфраструктури, який також називають площиною даних. Він містить елементи мережі пересилання.

Обов'язки експедиторської площини в основному полягають у пересиланні інформації, а також у локальному моніторингу інформація та збір статистики.

Одним шаром вище ми знаходимо контрольний рівень, який також називають контрольною площиною. Це відповідає за програмування та керування експедиційним літаком. З такою метою він застосовує представлену інформацію площиною пересилання та вказує функціонування мережі а також маршрутизацію. Складається з одного або декількох програм контролери, які спілкуються з складовими мережі передачі через стандартизовані інтерфейси, що мають назву південний інтерфейс. Open_Flow, який є одним із найбільш використовуваних південних інтерфейсів, в основному розглядає комутатори, тоді як інші підходи SDN розглядають інші елементи мережі, такі як маршрутизатори.

Прикладний рівень має у собі мережеві програми, що мають змогу вводити нові мережеві функції, такі як безпека та керованість, схеми пересилання або допомога керуючому рівню в конфігурації мережі.

Приклади мережевих додатків будуть розглянуті в наступних розділах. Прикладний рівень може отримати абстрактне та глобальне уявлення про мережу від контролерів і використання цієї інформації для надання відповідні вказівки для рівня управління. Інтерфейс між прикладним рівнем і елементом керування шар називають північним інтерфейсом. Для останнього на даний час не існує API, який є стандартизованим, а в на практиці керуюче програмне забезпечення надає власний API для програм.

Параметри протоколу для південного інтерфейсу. Найпоширенішим програмним інтерфейсом є OpenFlow, стандартизований Open Networking

OpenFlow є протокол, який здійснює опис взаємодії одного або декількох серверів контролю з перемикачами, які також є сумісними з Open_Flow. Контролер Open_Flow вказує записи таблиці трафіків у комутатори, скільки ці комутатори мають можливість передавати потоки інформації у відповідності даних записів. В такому разі, перемикач OpenFlow має залежність від конфігураційних характеристик контролерів. Трафік класифікується по полям відповідності, які подібні спискам керування доступом (ACL) та містити символи підстановки. У розділі 3 ми надаємо детальний опис OpenFlow і описати функції, які пропонують різні версії протоколу.

Іншим варіантом для південного інтерфейсу є розділення елементів пересилання та керування (ForCES) [5,6], який обговорюється та стандартизований Інженерною групою Інтернету (IETF) з 2004 року. ForCES також є фреймворком, а не лише протоколом; також структура ForCES розділяє площину керування та площину даних, але вважається більш гнучкою та потужнішою, ніж OpenFlow [8]. Пристрої пересилання моделюються за допомогою логічних функціональних блоків (LFB), які можуть бути скомпоновані за модульним принципом для формування складних механізмів пересилання. Кожен LFB надає певну такі функції, як IP-маршрутизація. LFB моделюють пристрій пересилання та співпрацюють, щоб створити ще більше складні мережеві пристрої. Елементи керування використовують протокол ForCES для конфігурації взаємопов’язаних LFB для зміни поведінки елементів пересилання.

Архітектура SoftRouter [9] також визначає окремі функції керування та площини даних. Це дозволяє динамічні зв'язки між елементами керування та елементами площини даних, що дозволяє призначення елементів площини керування та даних. У [9] автори представляють архітектуру SoftRouter і висвітлити його переваги щодо протоколу Border Gateway Protocol (BGP) щодо надійності. ForCES і SoftRouter мають схожість з OpenFlow і можуть виконувати роль південного напрямку інтерфейс. Також обговорюються інші мережеві технології, а

також можливі південні інтерфейси в IETF. Наприклад, Path Computation Element (PCE) [8] і Locator/ID Separation Protocol (LISP) [10] є кандидатами на програмні інтерфейси.

Північні API для мережевих програм. Як показав аналіз, протокол OpenFlow надає інтерфейс, який дозволяє керуючому програмному забезпеченню програмувати комутатори в мережі. По суті, контролер може змінити поведінку переадресації комутатора, змінивши таблицю переадресації. Контролери часто надають подібний інтерфейс додаткам, який називається північним інтерфейсом, щоб розкрити програмованість мережі. Північний інтерфейс не стандартизований і часто дозволяє детально керувати комутаторами. Додатки не повинні мати справу з деталями південного інтерфейсу, наприклад, додатку не потрібно знати всі подробиці про топологію мережі тощо. Наприклад, додатки мережі інженерного трафіку повинні повідомляти контролеру структуру шляху потоків , але контролер повинен створити відповідні команди для зміни таблиць переадресації комутаторів. Таким чином, мови мережевого програмування необхідні для полегшення та автоматизації налаштування та керування мережею.

Вимоги до мови для SDN обговорюються в [8]. Автори акцентують увагу на трьох важливих аспектах. (1) Мова мережевого програмування має забезпечити засоби для запиту стану мережі. Середовище виконання мови збирає стан мережі та статистичні дані, які потім надаються програмі; (2) Мова повинна мати можливість виражати мережеві політики, які визначають поведінку пересилання пакетів. Має бути можливість комбінувати політики різних мережевих програм. Мережеві програми, можливо, створюють суперечливі мережеві політики, і мережева мова має бути достатньо потужною, щоб виражати та розв’язувати такі конфлікти; (3) Реконфігурація мережі є складним завданням, особливо з різними мережевими політиками. Середовище виконання має ініціювати процес оновлення пристроїв, щоб гарантувати збереження контролю доступу, уникнення петель пересилання або дотримання інших інваріантів.

Популярними мовами програмування SDN, які задовольняють представленим вимогам, є Frenetic [10], її наступник Pyretic [11] і Procera [12]. Ці

мови забезпечують декларативний синтаксис і базуються на функціональному реактивному програмуванні. Завдяки природі функціонального реактивного програмування, ці мови забезпечують компонований інтерфейс для мережевих політик. Існують також різні інші пропозиції. Європейський дослідницький проект FP7, NetIDE, стосується північних інтерфейсів мереж SDN [13].

SDN, активні та програмовані мережі. У минулому були розроблені різні технології, щоб забезпечити можливість програмування в комунікаційних мережах. Активні мережі (AN) були розроблені в 1990-х роках. Основна ідея AN полягає у впровадженні програм у пакети даних. Комутатори витягують і виконують програми з пакетів даних. За допомогою цього методу нові механізми маршрутизації та мережеві служби можуть бути реалізовані без модифікації апаратного забезпечення пересилання. Однак цей підхід не переважає через проблеми безпеки та проблеми з продуктивністю, які можуть виникнути під час виконання програм на пристроях пересилання. Наприклад, зловмисник може впроваджувати шкідливі програми в мережеві пакети та пересилати їх у мережу.

Програмовані мережі (PN) [14] також забезпечують засоби для програмування в мережі шляхом виконання програм на пакетах, подібних до AN. Однак програми не включаються в пакети даних, як у AN. Програми встановлюються всередині вузлів мережі, які виконують програми на пакетах. Це явно зменшило проблеми з безпекою, які виникають з AN, оскільки вузол мережі приймає програми лише після попередньої сигналізації та налаштування вузла. Раніше були зроблені різні пропозиції щодо ПН. Наприклад, xbind було запропоновано для мереж з асинхронним режимом передачі (ATM), які адаптовані до якості обслуговування (QoS) і мультимедійних програм. Мета xbind полягала в тому, щоб подолати складність, пов’язану з банкоматом, і полегшити створення сервісу шляхом відокремлення алгоритмів керування від протоколу банкомату, що стало можливим завдяки програмованості пристроїв пересилання за допомогою мов програмування.

Обидва підходи, AN і PN, вводять нову гнучкість, дозволяючи програмам виконуватися в мережі. Вони є більш гнучкими, ніж підхід SDN на основі

OpenFlow, оскільки вони дозволяють програмувати площину даних розширюваним способом. Нова функціональність площини даних може бути реалізована за допомогою програм, які або є частиною пакетів даних з AN, або встановлені всередині вузлів мережі з PN. Підхід SDN на основі OpenFlow не може розширити функціональність площини даних без оновлення комутаторів через те, що OpenFlow забезпечує лише фіксований набір мережевих операцій. Контролер OpenFlow може програмувати комутатор лише з підтримуваним набором операцій.

Протокол OpenFlow. Протокол OpenFlow є найпоширенішим протоколом для південного інтерфейсу SDN, який відокремлює площину даних від площини керування. Білий документ про OpenFlow вказує на переваги гнучко настроюваної площини пересилання. OpenFlow спочатку був запропонований Стенфордським університетом, а зараз стандартизований ONF [4]. Далі ми спочатку даємо огляд структури OpenFlow, а потім описуємо функції, які підтримуються різними специфікаціями.

Архітектура OpenFlow складається з трьох основних концепцій:

1. Мережа побудована за допомогою OpenFlow-сумісних комутаторів, які складають площину даних;
2. Площина керування складається з одного або кількох контролерів OpenFlow;
3. Безпечний канал керування з’єднує комутатори з площиною керування.

Комутатор, сумісний з OpenFlow, — це базовий пристрій пересилання, який пересилає пакети відповідно до своєї таблиці потоків. Такі таблиці містять набори заміток таблиць потоків, який має складову із полів певної відповідності, лічильника а також необхіднх інструкцій, як показано на рис.1.4. Записи таблиці потоків також називаються правилами потоку або записами потоку.

«Поля заголовка» в записі таблиці потоків описують, до яких пакетів цей запис застосовується. Вони складаються зі збігу з підстановкою для заданих полів заголовка пакетів. Щоб дозволити швидке пересилання пакетів за допомогою OpenFlow, комутатор потребує потрійної пам’яті з адресацією вмісту (TCAM), яка дозволяє швидко шукати збіги з підстановкою. Поля заголовка можуть відповідати

різним протоколам в залежності від специфікацій. «Лічильник» призначений для зборів статистичних даних про потік. Вони виконують збереження кількості отриманих пакетів а також байтів і тривалості потоків.

«Дія» виконує визначення, яким чином виконують обробку пакетів цього потоку. Типовими діями є «переслати», «скинути», «змінити поле» тощо.

Заголовок	поля	Лічильники Дії

Рисунок 1.4 - Запис таблиці потоку для OpenFlow 1.0

Програмне забезпечення, яке називається контролером, відповідає за заповнення та керування таблицями потоків перемикачів. Вставляючи, змінюючи та видаляючи записи потоку, контролер може змінювати поведінку комутаторів щодо переадресації. Специфікація OpenFlow визначає протокол, який дозволяє контролеру передавати команди комутаторам. Для цього контролер використовує безпечний канал керування.

Протокол OpenFlow має 3-и класи зв’язку: «контролер-комутатор», асинхронного і симетричного зв’язоку. Зв'язоки між контролерами і комутаторами має відповідальність за визначення функції, конфігураційніхарактеристики, можливість програмувати комутатори і пошукирізного роду інформації. Асинхронні зв’язоки ініціюються комутатором, які сумісні з OpenFlow, без будь-якого запиту з зі сторони контролерів. Він використовується для інформування контролера про надходження пакетів, зміни стану комутаторів і помилки. На додаток, симетричне повідомлення надсилається не виконуючи запит з будь-якої із сторін, мається на увазі що комутатори чи контролери здатні вільно встановлючвати зв’язокне виконуючи запитів з іншого боку. Прикладами симетричного зв’язку є повідомлення привітання або ехо-повідомлення, за допомогою яких можна визначити, чи канал керування все ще активний і доступний.

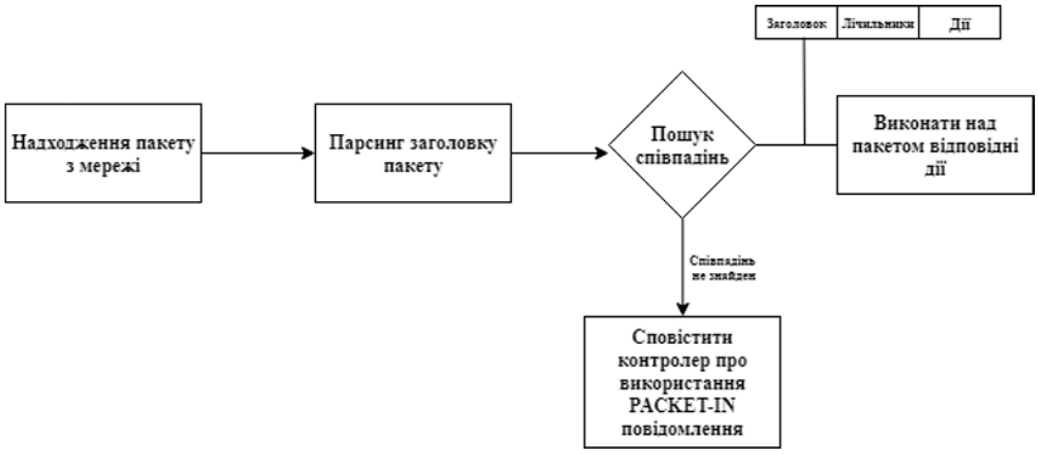


Рисунок 1.5 - Базове пересилання пакетів за допомогою OpenFlow у комутаторі

Основний механізм пересилання пакетів за допомогою OpenFlow проілюстровано на рис.1.5. Коли комутатор отримує пакет, він аналізує заголовок пакета, який зіставляється з таблицею потоку. Якщо знайдено запис таблиці потоку, у якому символ підстановки поля заголовка відповідає заголовку, цей запис розглядається. Якщо знайдено кілька таких записів, пакети зіставляються на основі пріоритезації, тобто вибирається найбільш конкретний запис або символ підстановки з найвищим пріоритетом. Потім комутатор виконує оновлення лічильників даного запису таблиці потоків. Нарешті, комутатор здійснює дію, яка визначена записами таблиці потоків для пакетів, приколадом може бути ситуація коли комутатор виконує пересилання пакетів на певні порти. У випадках, коли жоден із записів у таблицях потоку не відповідають заголовкам пакетів, комутатор зазвичай сповіщає свій контролер про пакет, який буферизується, коли комутатор здатний до буферизації. З цією метою він інкапсулює або небуферизований пакет, або перші байти буферизованого пакета за допомогою повідомлення PACKET-IN і надсилає його до контролера; зазвичай інкапсулювати заголовок пакета, а кількість байтів за замовчуванням дорівнює 128. Контролер, який отримує сповіщення PACKET_IN, визначає правильну дію для кожного з пакетів та виконує встановлення один або більше відповідних записів у комутаторах, що запитують. Далі пакети які буферизовані передаються відповідно до правил; це ініціюється

встановленням ідентифікаторів буфера у повідомленнях вставок потоку чи у явному повідомлені PACKET_OUT. Дуже часто контролери встановлюють усі шляхи для пакетів в мережі, змінюючи таблиці потоку всіх комутаторів на шляху.

1.3 Дослідження перспектив подальшої еволюції мереж зв'язку

Як показав аналіз, поява новітніх технічних рішень у послузі, мереж зв'язку є загальними причинами поступового переходів до послідуючих концепцій– мережі зв'язку у ту саму годину, коли роботи з мережами наступного поколінь завершеними є помилкою як на рівнях стандартизуванні, дослідів, а і у виробничих рівнях. В такому разі, для отримання результау і виконанні мети контролю концепції 2030 в МСЕ створені спеціалізовнаі робочі групи для досліджень та наступного стандартизування поколінь мережі 2030, що повинна виявити загальну характеристику та напрямок виконання стандартизування мереж і послуги. Варто вказати, щоб зазирнути майбутнє, варто проаналізувати загальні напрямки у еволюції мереж, які виникли сьогодні у момент функціонування концепції IMT-2020. Як приклад, вже не одного разу згадувалися послуги «Тактильного Інтернету». Така послуга з кількох технологічних причин, що вимагає більш ретельних досліджень та доопрацювання до того моменту, як зможе зайняти місце серед інших послуг. Можливо згадати, саме для використання в ТІ поставлені вимоги до затримок, які не мають бути більшими 1мс та на теперішній момент еволюції технологій, також маються на увазі технології які відносяться до фізичних рівнів, є тяжкими, а в кількох напрямках застосування – невиконаною задачею. Розглядаючи концепцію мережі зв'язку 2030, що прийнята вже МСЕ і описується в документації. Можливо вказати наступну фундаментальну різницю у еволюції мережі зв'язку:

- мережі які мають ультрамалу затримку. ТІ призводить до більшої зміни у напрямку реалізації мережі зв'язку, оскільки в даному випадку необхідно було виконувати передачу інформації із затримками в 1мс, що є у 100 раз меншою, які є у мережі зв'язку на теперішній час. В даному випадку слід звернути на те що

сукупність положень ТІ може призвести до певної дії, а саме децентралізування мережі зв'язку, бо фундаментальне обмеження швидкостей і передавання світла перебороти не можливо [4].

- мережа з надщільною можливістю. Надщільністю мереж є одна із ознак не лише мереж IMT-2020, а й також мереж подальших. Як зазначалось раніше, за передбаченнями орієнтовно мільйону замобів на 1 квадратний метр та у відповідності до передбачень, обмежене число IoT буде 50 трил., що ймовірно буде досягнено орієнтовно до 2030 років. Отже, поняття надщільної мережі буде все більш підтверджуватись у процесі еволюції мережі та різного роду послуг до 2030 [4].

- інтернет навичка. Досить цікавою та перспективною концепцію послуг є Інтернет навичка, що виникла у 2017 р. для якої необхідна характеристика мережі що включає в себе ультрамалу затримку. Сокупність інтернет знань як концепція надає змогу виконати реалізацію в мережі з ультрамалою затримкою нового виду послуги. Такі послуги нададуть можливість використовувати мережі для набуття користувачами і пристроями нового бачення та знань [7].

- мережі які мають можливість літати. Варто вказати, ще одною фундаментальною зміною у еволюції мережі зв'язку є інтегрованість літального і сегменту що знаходиться на землі. Також слід вказати, якщо використовується невелика висота польоту БПЛА на базі подібних мереж, виконується вимірювання в мережах зв'язку із ультрамалою затримкою.

- транспорт який має безпілотне керування. Цей перспективний напрямок реалізується не тільки спеціалістами ІКТ, але і спеціалістами автомобільних галузей. У 2021 році, у цьому напрямку, виробництво «переганяє» стандартизаційні рівні, не згадуючи навіть про мережеву та зовнішню математичну складову (за межами борта транспорту). В даному випадку варто згадати «Tesla», компанія яка стрімко розвивається, не беручи до уваги негативні відгуки певної частини споживачів, за попередні 10 р., мала успіху і до сих пір застосовує найефективнішу послугу «автопілот». І все ж, досвід використання літаючого зразку який керується дистанційно показав, що важкорозв'язні а також певною

мірою неможливих для реалізації сценаріїв. Таким чином, науковці дійшли висновків, що варто виконати об'єднання безпілотних засобів руху у єдину мережу, яка може підтримати цифрові моделі дороги чи невеликої частки дороги, по якій функціонує безпілотник. В такому разі, «транспорт» має змогу зазделегіть про те, що є через 8 км, і знаходиться поза баченням бортової систем.

У результаті огляду фундаментальної зміни у еволюції мережі зв'язку, можливо стверджувати, про те що мережа у 2030 має бути надщільною мережею з ультрамалою затримкою, яка прагне децентралізованої структури мереж, а ще дана мережа зв'язку має набути дещо нової характеристики на базі розвивання технології у галузях мереж а також суміжної галузі.

Слід звернути увагу на наступні тенденції мереж зв'язку майбутнього, які згадувались в документах МСЕ:

- послуга яка забезпечує телеприсутість. Персоналізована мережа. Є однією із найперспективніших напрямків для надання послуг мереж майбутнього. В якій прогнозується напрямки використання пару аватарів однією людиною. Для можливості застосування такого типу мережі потрібно розгортання мережі з ультрамалою затримкою. В такому разі послуга присутності не має обмежень тільки у відтворенні якихось дій особи яка користується. У разі використання голографічного додатку і аватару буде змога, як приклад, спостерігати баскетбольний матч не використовуючи телевізори, а спостерігати голографічні моделі зрізного кута [4].

- мережі з можливістю літати. У наш час еволюція мереж, які мають змогу літати, передбачає все більшої кількості аналізу для можливості їх використання в цивільному житті людини. В такому разі, згідно з перспективними трендами літальна мережа зможе використовуватись у широкому цивільному використанні у найближчому майбутньому[4].

- наномережі. У майбутньому різного роду використання таких мереж [9] мінімум в медицині.

У відповідності рекомендацій МСЕ мережа майбутнього зможе підтримувати різного роду і обмежені функційні і нефункційні вимоги,

передбачаючи чималі вимоги до невеликих затримок і чималого обсягу обміну інформації.

Нові додаткові характеристики та можливості мереж майбутнього перераховані нижче[10]:

- детермінізація в затримці та передавання без втрати;
- вмонтована підтримка декількох видів послуг, можливість вчасної активації та доступності послуги;
- можливість гнучких налаштувань мережних сервісів та мережевих функцій;
- ефективний програмований мережевий протокол;
- безпекова мережа;
- надстійкість при виникненні відмов;
- інтегрованість чималої кількості інтелект методів у мережні інфраструктури, на меті якої є контроль і управління;
- модернізованість архітектур у сторону розподілення управління.

Архітектурний принцип мереж і послуг є напрямки рішення проектувань, які є основою для функціонування систем. Всі принципи застосовані з певним набором міток зору в архітектурі. З точки зору користувачів, принцип таких систем розуміється як певна характеристика систем, які відображають на значення у використанні систем та її ефективна робота [10]. В рекомендаційних документах МСЕ пропонують наступні архітектурні засади, що відображені на рис.1.6.

Для більшості характеристик мереж майбутнього визначатимуться технологіями які зараз та наступними, які мають знайти широке застосування для швидкої реалізації такої мережі. ШІ (а саме перша його реалізація/прототипи) матимуть змогу інтелектуально керувати потоком трафіків, квантові машини нададуть змогу пристроям користувача використовувати чималу кількість нових складних задач, нанотехнологія надасть можливість відкрити світ телекомунікацій і надати нові види послуг. А також, концепції Індустрії 4.0 зможе вийти на вищий рівень та можливі кращі зміни, що в теперішню годину є складно оцінювальною [4].

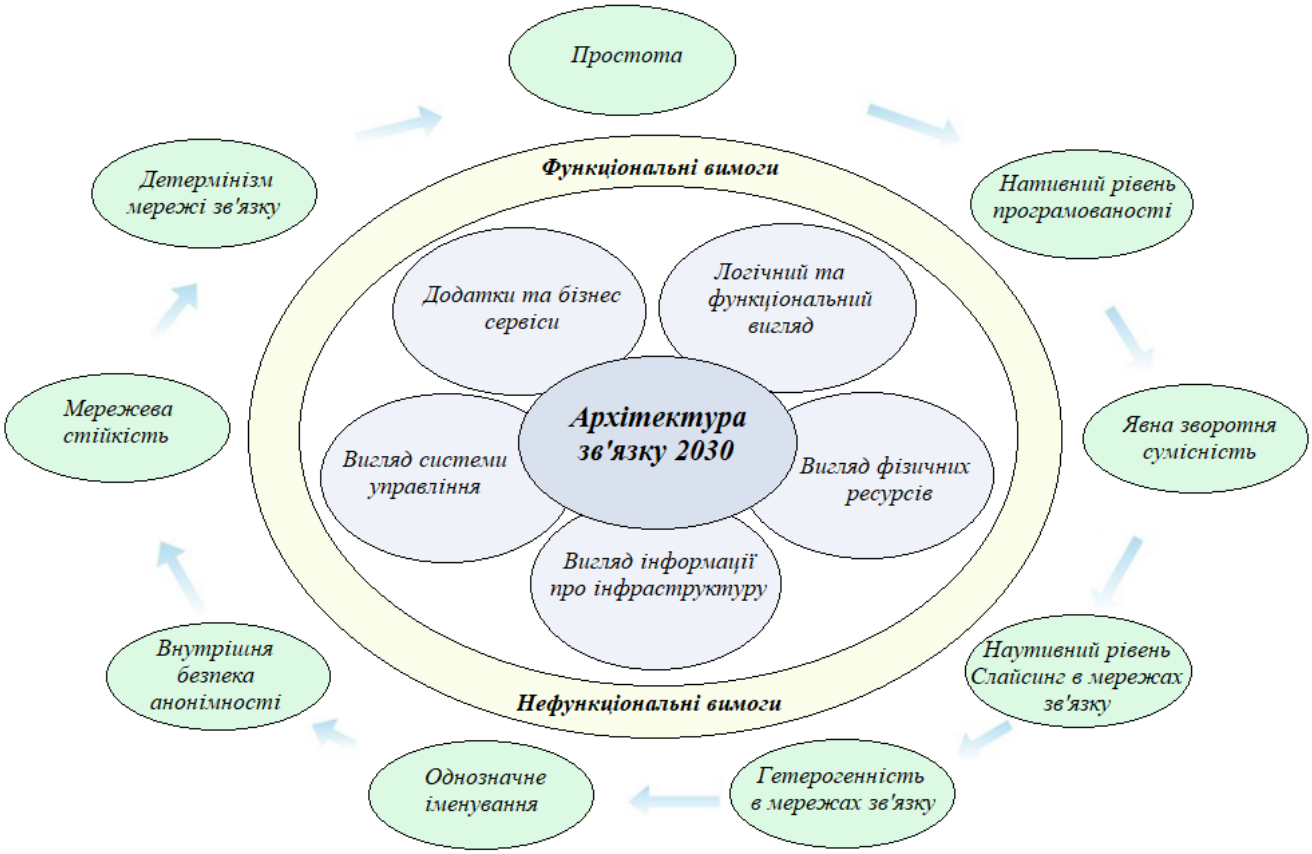


Рисунок 1.6 - Принципове рішення розробки архітектури мережі

1.4. Особливості застосування штучного інтелекту у мережі

Передумовами чималого інтересу використання технологій ШІ в мережах полягає у надії на яку сподіваються для вирішення певної кількості задач, які є дуже важкими у вирішенні перед сучасними мережами та їх подальших релізів. У стратегічних планах мережі мають виконувати реалізацію вимог наднадійності мережі з ультрамалими затримками [11], не кажучи вже про все зростаючу кількість трафіку міжмашинних комунікацій в мережі, а також мати можливість контролю та управління безліччю сучасних та нових послуг з високими вимогами щодо якості обслуговування та сприйняття, наприклад, для послуг тактильного інтернету або послуг телеприсутності [12].

У вирішенні завдань моніторингу та управління трафіком у мережах планується використання технологій ШІ. Потреба у впровадженні таких ресурсомістких технологій обробки даних з'явилася в результаті вимоги до

динамічності мережевої інфраструктури та її здатності адаптуватися до змін умов. В даний час генерується безліч службових даних, які необхідно ефективно обробляти і це стає під силу лише технологіям ШІ [2]. У вирішенні подібних завдань проявляється одна з позитивних фундаментальних переваг мереж зв'язку, побудованих на технологіях SDN/NFV – програмованість. Програмування мережі означає якісну зміну моделі управління мережею: мережевий адміністратор – мережа. Завдяки «прозорості», що є змогу надати із застосуванням певної кількості технологій та протоколів SDN, мережею може керувати програмне забезпечення, яке може реалізовувати саме ті самі алгоритми обробки великих даних та машинного навчання. Завдяки закладеним функціональним можливостям протоколу OpenFlow, OFConfig, програмованим північним інтерфейсам контролера (наприклад, REST, SOAP та ін.), програмне забезпечення верхнього рівня, представлене у вигляді службових сервісів, може в режимі онлайн отримувати всю необхідну інформацію про потоки даних, що передаються на рівні передачі даних (ПД).

Задача ШІ у мережі. Велика кількість задач що стосується реалізації ШІ в мережі є можливість розділити на два типи: розпізнання та прогноз, а саме розпізнання типів передаючого трафіку, або розпізнавання атак які надходять на контролер мережі. Можливо виконувати прогноз навантажень на комплекси оператора в цілому з урахуванням множини критеріїв.

В такому разі, технології реалізації мережної інфраструктури SDN/NFV надають можливість вирішити чималу кількість задач, які ставились перед мережами майбутнього. При цьому їхнє рішення лежить через інтелектуалізацію мережі шляхом розробки службових сервісів із застосуванням технологій ШІ.

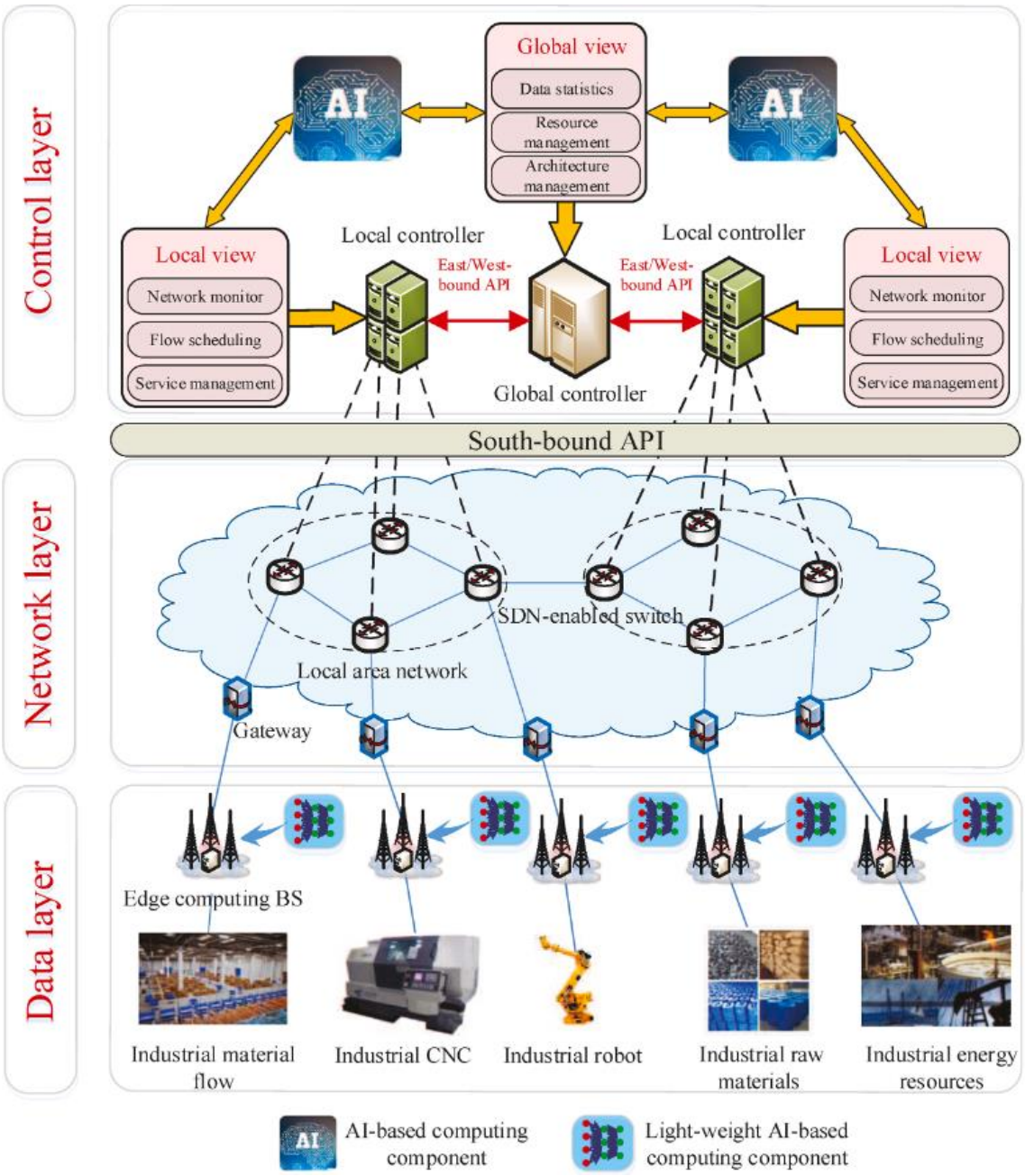


Рисунок 1.7 - Модель мережі із застосуванням ШІ

На рис. 1.7 приведена схема мережі з ШІ, яка реалізується із застосуванням технологій SDN/NFV [3].

Наразі в якості досліджень у галузі застосування ШІ [3] у мереж варто виділити певну кількість задач:

- ідентифікування трафіку в мережі, що не вводить допоміжних затримок у потік і надає змогу забезпечувати вимоги мереж з ультрамалою затримкою;

- системтичний моніторинг в онлайн режимі мережі від потоку даних до багатопараметричних моделей у сегменті мережі що має множинний доступ пристроїв;

- не тривале та тривале виконання прогнозу навантажень на елементи мережі, а також на цілий сегмент;

- не тривалий та тривалий прогноз поведінки потоку передачі інформації на рівні ПД і службових потоків на рівні управління;

- тривалий прогноз навантаження на мережну та обчислювальну структуру;

- ефективне розподілення радіоресурсів покриття мереж з прогнозом навантажень на вічки;

- збільшення показників якості сигналу із використанням прогнозного кодека;

- прогноз у русі клієнтів географічно;

- розпізнавання і прогноз атак злочинців на системи які формують випереджальні реакції на ймовірну атаку;

- використання технологій ШІ щоб виконувати узгоджуючий розподіл сервісів по мережі.

Такий перелік задач у галузях ШІ в мережах з, природно не є всі завданнями в галузі аналізу ШІ. За результатом проведеного огляду існуючих робіт, в планах вести аналіз застосування ШІ у мережах, як розділеної і незалежної системи з урахуванням децентралізованості мережі. Суть децентралізованості зображено рис.1.8.

В мережах має бути функціональний оркестр (модуль) міжмашинних навчань, який здійснює моніторингові задачі, контролює передавання відповідних функціоналів міжмашинних навчань системам управління мережами. Зважаючи на складнощі та компонентність мереж, такий оркестратор не централізований та визначальний на даній мережі. Дані оркестратори мають мати можливості взаємодій між собою для можливості отримування набагато більшого синергетичної ефективності від запровадження ШІ у мережі. У випадку, як представлено на рис.1.8, при оркестраторі є певні сховища функцій а також модулів

машинних навчань, які при потребі переїжають як службові мікросервіси у необхідну систему керування мережею. Це супроводжується безперервним обміном даними між оркестратором та всіма йому підконтрольними модулями функціями міжмашинних навчань [3].

У такому випадку варто використовувати концепцію ШІ у мережі зв'язку, оскільки є важливим принципом мікросервісних підходів, які надають змогу виділити необхідні функціонали машинних навчань у необхідні модулі, які мають взаємодію один з одним, так і з вищим (керуючим) модулем. Виникає гнучкість управлінню системами, її оновленні (по бажанню), та більш легшому та незалежних масштабувань.[13].

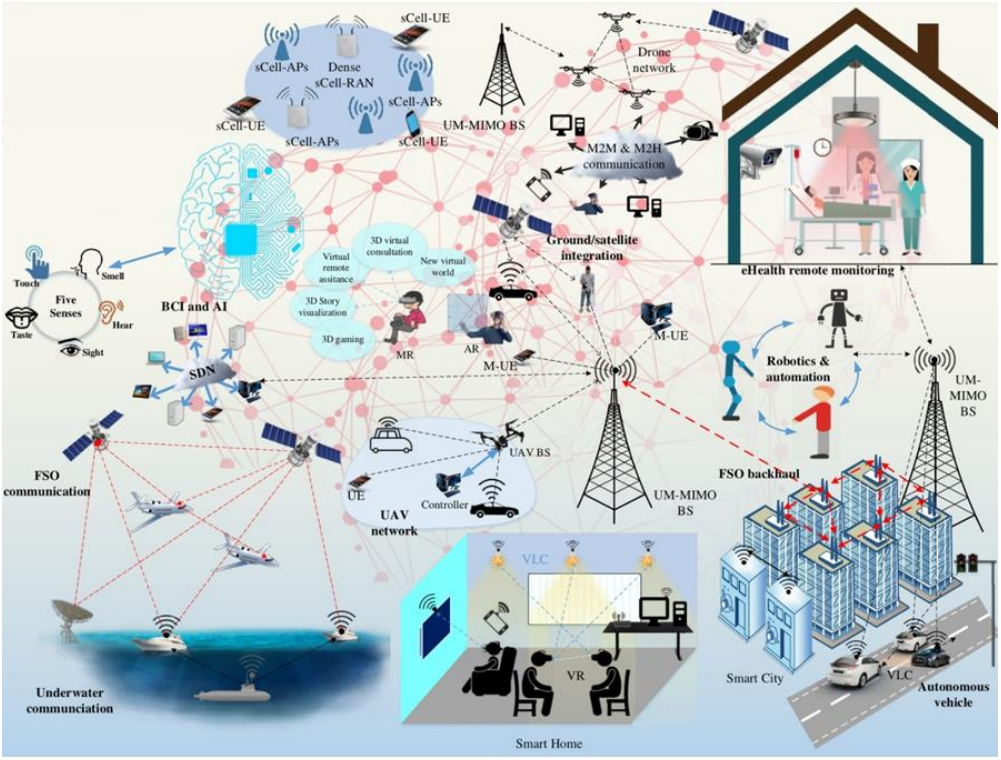


Рисунок 1.8 - Приклад архітектури розподіленого ШІ в мережах та перспектива подальших досліджень

Варто вказати, що аналіз з застосуванням ШІ у мережах є особливо актуальним в теперішній час [9].

Аналітичний огляд стандартизації ШІ в мережах зв'язку. В даний час у МСЕ-Т ведеться активна робота з аналізу та розробки проектів рекомендації в полі

ШІ у мережах. На даний момент, МСЕ-Т вже випустило ряд рекомендацій у галузі машинного навчання, але основною вважається Рек.Y.3172. Функціонально одним з основних елементів в рік. Y.3172 є конвеєром міжмашинного навчання, а саме набором вузлів логічного типу, які мають деякі функції. Дані вузли можливо комбінувати для формування програми машинного навчання в телекомунікаційній мережі. У рекомендації також представлено високорівневу архітектуру конвеєра з визначенням усіх основних функціональних блоків та їх взаємодією між собою [3]. Архітектура представлена рис.1.9.

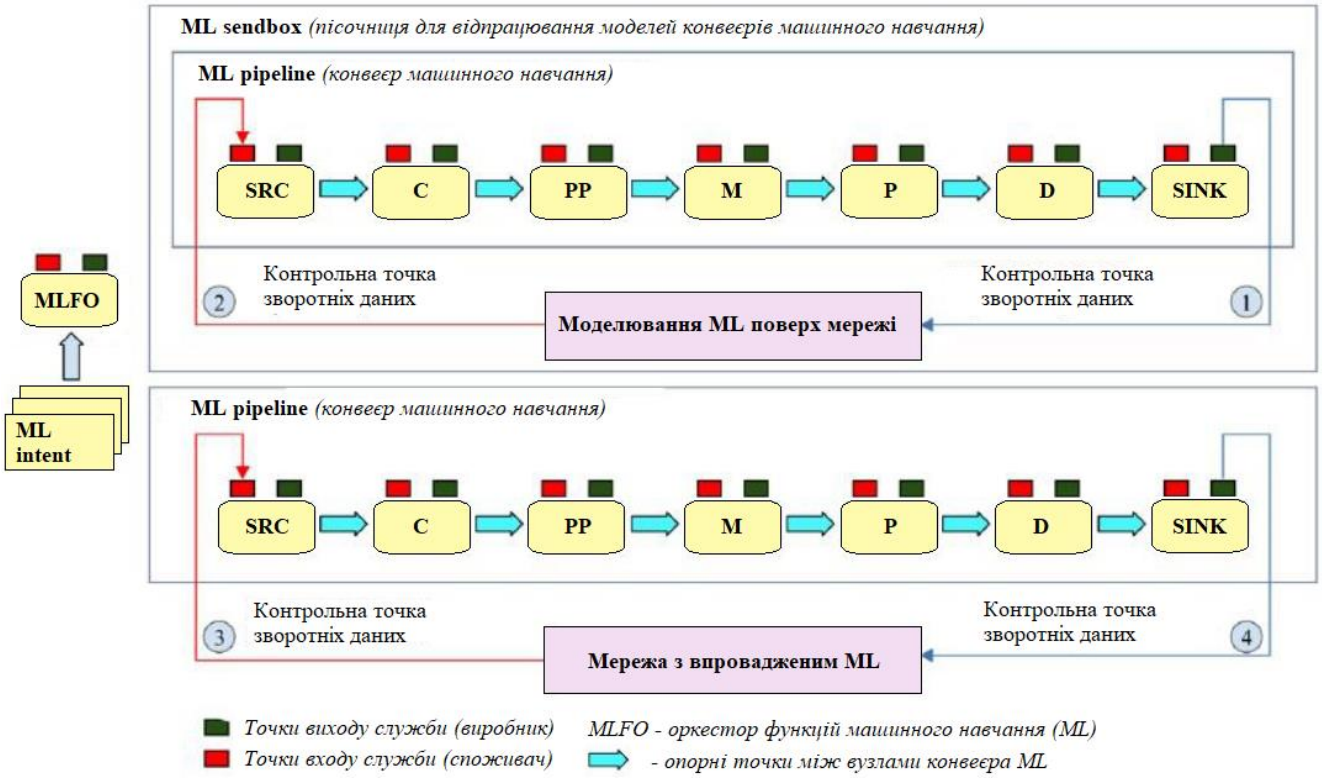


Рисунок 1.9 - Конвеєр машинного навчання для мереж зв'язків

Складові високорівневої архітектури конвеєра машинного навчання, згідно з річками. МСЕ-Т Y.3172 «Архітектура для машинного навчання в майбутніх мережах, включаючи IMT-2020»:

- SRC (Source) – джерело запиту на обслуговування. Потенційне джерело включає обладнання користувача (з англ. UE-User Equipment), Функції управління сесіями (з англ. SMF - Session Management Function, згідно з рек. МСЕ-Т Y.3104

[15]) і функції додатків (з англ. SMF). AF - Application Function, згідно з річками MCE-T Y.3104 [15]);

C (Collector in ML pipeline) – колектор збору запитів для системи машинного навчання на рівні 1 для мережі доступу та на рівні 2 для ядра мережі. Колектор збору запитів може конфігурувати SRC. Наприклад, протокол функцій контролю радіоресурсів (з англ. RRC – Radio Resource Control, може бути використаний для конфігурування користувача пристрою (UE), представленого в якості SRC пристрою Колектор може використовувати спеціальні протоколи операцій, адміністрування та обслуговування (з англ. OAM – Operations, administration and maintenance) для налаштування функцій управління сесіями (SMF), що виступає як вузл SRC. Такі конфігурації можуть використовуватися для управління характеристиками даних, їх деталізацією та періодичністю, у процесі їх генерації з SRC;

- PP (Preprocessor) – елемент попередньої підготовки даних. Цей вузол відповідає за «очищення» даних, агрегування даних або виконання будь-якої іншої попередньої обробки, щоб інформація була у відповідному форматі для їх подальшого використання у моделях машинного навчання;

- M (Model) – це модель машинних навчань у форматі, який може використовуватись у конвеєрі машинного навчання;

-P (Policy) – вузол керування політикою. Цей вузол дозволяє застосовувати різні політики щодо вихідних даних вузла моделі. Конкретні правила можуть бути введені мережним оператором для забезпечення працездатності мережі, наприклад, серйозні оновлення можуть виконуватися лише в нічний час або за низького трафіку даних у мережі;

- SINK – це вузол-мета виведення ML. Наприклад, UE, що діє як вузол SINK, може регулювати періодичність вимірювання каналу на основі виведення ML.

Примітка: "Точка виходу служби" вузла - це точка, де можна отримати доступ до служб, що надаються вузлом (забезпечує виробник). "Точка входу служби" вузла - це точка, звідки можуть споживатися служби інших вузлів (забезпечує виробник);

- D (Distributer) – вузол розподілу. Цей вузол відповідає за ідентифікацію про SINK і розподіл вихідних даних M-вузла на відповідні вузли SINK;

- MLFO (Machine Learning Function Orchestrator) – оркестратор функцій машинних навчань. Є логічний вузол з функціями, які управляють та координують вузли конвеєрів ML на основі даних ML та/або динамічних мережних умов;

- ML sandbox - це ізольований домен, який дозволяє розміщувати окремі конвеєри машинного навчання для їхнього навчання, тестування та оцінки перед розгортанням у діючій мережі. Для навчання або тестування пісочниця машинного навчання може використовувати дані, що згенеровані з імітованих базових мереж машинного навчання та/або живих мереж.

- ML Intent – це декларативний опис, який використовується для визначення програми ML. ML Intent не визначає жодних мережних функцій, а забезпечує основу для зіставлення програм ML з різними мережевими функціями, що залежать від технологій. ML Intent може використовувати метамову, специфічну для машинного навчання, визначення додатків ML.

Реалізація конвеєра, враховуючи архітектуру мережі зв'язку 5G/IMT-2020[13], наведено на рис.1.10. Тут прийнято такі позначення з річок. MCE-T Y.3102 [19]:

- SMF (Session Management Function) – функція керування сесіями;

- NACF (Network Access Control Function) – Функція контролю доступу до мережі. Функціональні можливості NACF включають керування реєстрацією, керування з'єднанням та вибір функції керування сеансом (SMF).

- AF (Application Function) – функції програми. Ця функція надає інформацію про сеанс у PCF, щоб SMF міг використовувати цю інформацію для керування сеансом. AF взаємодіє зі службами програм, які потребують динамічного управління політиками. AF витягує інформацію, що стосується сеансу (наприклад, вимоги QoS) із сигналізації програми, і надає її PCF для підтримки генерації правил.;

- PCF (Policy control function) – функція контролю політик. Ця функція надає функціональні можливості для контролю та керування правилами політики, включаючи правила забезпечення QoS.

- UPF (User Plane Function) – ця функція забезпечує функції для маршрутизації трафіку, керування тунелем сеансу PDU (IP або не-IP протокольна одиниця – з англ. Protocol Data Unit) та забезпечення QoS.

- CEF (Capability exposure function) – ця функція надає функціональні можливості для мережевих функцій та мережевих сегментів, щоб надавати їх можливості як послугу третім сторонам.

- NFR (Network function registry function) – ця функція надає функції, які допомагають виявляти та вибирати необхідні мережеві функції. Кожен екземпляр мережевої функції реєструється під час створення та оновлює свій статус (тобто активація/деактивація), так що NFR може підтримувати інформацію про доступні екземпляри мережевих функцій.

- USM (Unified Subscription Management function) – ця функція зберігає та керує уніфікованим UE та інформацією про підписку, включаючи, крім іншого, інформацію про реєстрацію UE та управління мобільністю, інформацію про мережеві функції, які обслуговують UE, інформацію про управління сеансом для встановлення сеансу PDU. USM також надає інформацію про аутентифікацію UE в ASF.

- ASF (Authetification Server Function) – ця функція виконує аутентифікацію між UE та мережею. NACF ініціює автентифікацію UE, викликаючи ASF. На основі інформації аутентифікації UE, отриманої від USM, ASF вибирає метод аутентифікації та виконує процедури аутентифікації UE (відповідно до стандарту 3GPP TS 33.501) [18].

- NSSF (Network Slice Selection Function) – ця функція надає функціональні можливості для вибору відповідних екземплярів мережевого сегменту (т.зв. «Слайса») для UE. Коли UE запитує реєстрацію в мережі, NACF відправляє запит вибору мережного сегмента NSSF з переважною інформацією про вибір мережного

сегмента. NSSF відповідає повідомленням, що включає список відповідних екземплярів мережевого сегмента для UE.

- UE-AN data transport – рівень передачі даних між пристроєм користувача та мережею доступу

- RP-au (Reference Point) – опорна точка взаємодії між власним обладнанням та UPF;

- RP-ud (Reference Point) – опорна точка між UPF та пакетною мережею передачі даних.

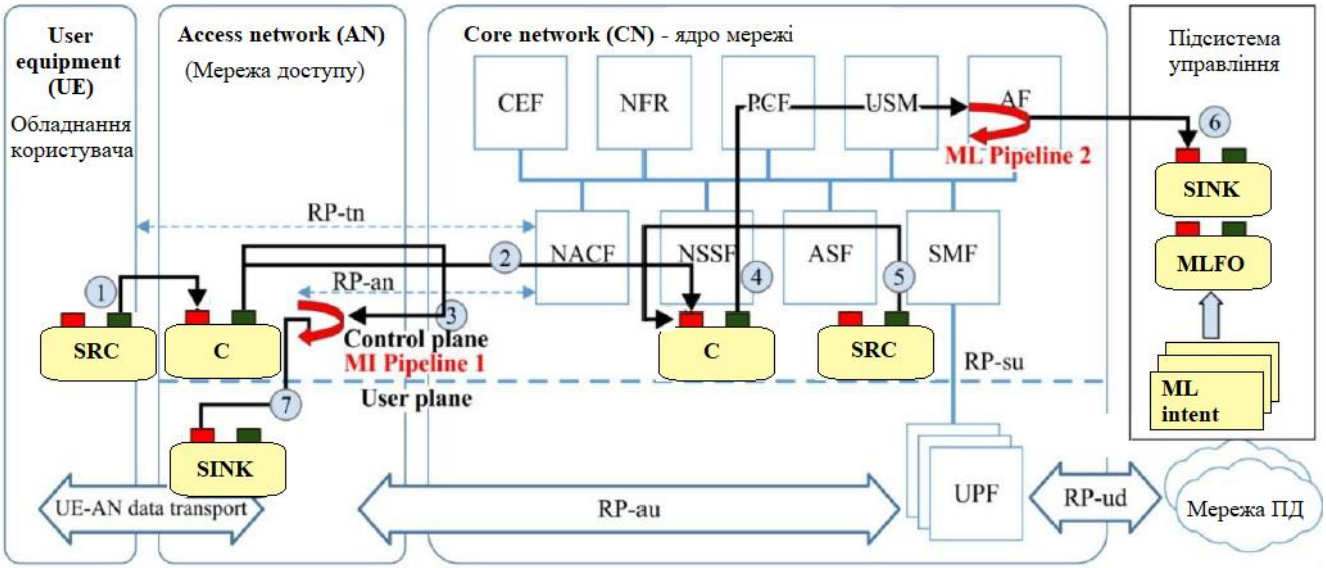


Рисунок 1.10 - Високорівнева архітектура конвеєра машинного навчання та його реалізація в мережі МТ-2020

З рис.1.10 видно, що досліджуваний конвеєр машинного навчання охоплює всі рівні мережі IMT-2020 [15], впроваджується в основні модулі управління мережею, рис.1.9, і забезпечує прозорий моніторинг і контроль функціонування мереж. Варто також зазначити, у межах таких рекомендацій виявлено функцію важливих елементів мережі, серед яких оркестратор міжмашинного навчання. Такі модулі мають змогу реалізувати функцію управління та оркестрації інших модулів, що відносяться до конвеєрів міжмашинних навчань, та враховує у прийнятті рішень динаміку функціонування мереж та їх характеристики.

РОЗДІЛ 2 АНАЛІЗ МЕТОДИКИ РОЗПІЗНАННЯ ТРАФІКА ПОСЛУГ В ПРОГРАМНО-КОНФІГУРОВАНИХ МЕРЕЖАХ

2.1 Принципи побудови SDN та розпізнання передачі даних послуг у мережах зв'язку

У попередніх дослідженнях, при огляді сучасної та перспективнї концепції мережі зв'язку, представлено три базових принципи, на базі яких реалізуються мережі наступних поколінь які в свою чергу є пріоритетними і в наступному поколінні:

- широкі міжмашинні взаємодії;
- модернізований безпроводовий зв'язок;
- зв’язок з більшою надійністю та ультрамалою затримкою передачі;

В теперішній час, URLLC є одною з найскладнішою задач, яка існує перед науково-дослідницьким персоналом. Такий тип мережі передбачає мати більш надійним а також використовувати наднизькі затримки передавання інформації. Отже, URLLC передбачає більш серйозні вимоги до якості в обслуговуванні (QoS), що надасть життя новому типу сервісів. Така послуга може реалізовуватись в наступних напрямках: цифрова медицина, автономні транспортні засоби, промислове виробництво, яке має статус небезпечного, які мають формат 4.0. Самим відомим типом сервісу URLLC є Інтернет на дотик. В такому разі слід вказати особливість а саме що на архітектури мереж найбільшого впливу матимуть ультрамалі затримки передачі:

- застосовується децентралізованість мереж у наслідок обмежень якіздійснюються на відстанях, при яких можливо здійснювати надання послуги Tactile Internet. Здійсненя взаємодії в режимі онлайн тактильних відчуттів, у випадку коли мережа надасть менші затримки, чим відповідні фізіологічні константи, які виражають інерційність сенсорних складових людини в 1мілісекунду. Задача здійснення вимоиг по величині затримок 1мс здійснює

накладання великих обмежень на системно-мережові вирішення реалізації мережі зв'язків. Tactile Internet як послуга, можливо реалізувати у медичній сфері, як приклад: огляд пацієнта на відстані, операція на відстані.

Зважаючи на останні події а також ситуацію з епідеміями в цілому світі можливо підбити підсумок, що послуги Інтернет дотику мають можливість зберігати життя медичного персоналу, які ризикуючи, лікували різні захворювання різної степені небезпеки. До прикладу, в лікарні з хворими на ковід можливо використовувати медичного робота, який керується лікарями на відстані та виконує передачу усіх необхідних відчуттів дотику у момент огляду. У такому разі і Інтернет навички також змогли б зайняти своє місце у застосуванні, як варіант надбудови тактильного Інтернету. Але, рівні розвитку технологічних рішень існуючої мережі не дає змоги застосувати такий тип сервісів [16].

Таким чином, з більшим застосуванням технологій як мережних, обчислювальних, варто виконувати перегляд принципів контролю та управління мережею. Виконуючи врахування можливостей програмованості, що закладаються в SDN s систем оркестрації обчислювальної структури, бажано виконувати їх реалізацію, враховуючи технологію ШІ. Теперішні обсяги трафіку, а також його властивості гетерогенності і різноманітності сервісів IoT, а також тих, які відносяться до URLLC, вказує все новіші вимоги до швидкості приймання рішення які забезпечують якість в обслуговуванні. Інструменти які є на даний момент які невзмозі надавати потрібний рівень [16]. У цьому разі вже в більшій частині рішень потрібно щоб були прогноз навантаження на різного роду послуг, виконуючи врахування географічне пересування користувача, а також враховувати їх швидкості. Тоді оператори мають потреби у повноцінній системі прогнозуванні еволюції інфраструктур, аналізуючи швидкості при яких виконуються впровадження сучасних технологічних сервісів в мережах, заміни способів життєдіяльності людини.

Варто вказати наступні KPI, досягнення яких нададуть змогу говорити про готовність мережі майбутнього з боку вимоги до затримки. Вимоги наскрізної затримки в мережах:

- 4 мілісекунд для послуги які мають тип eMBB;

- 1 мілісекунд для послуги які мають тип URLLC;

- наскрізні затримки на керуючому рівні– 10 мілісекунд;

А ще, надійність передавання інформації: $1 \cdot 10^{-5}$ для пакетів 32 байта із наскрізними затримками у мережах 1 мілісекунд послуги URLLC. Та щільність підключених клієнтський апаратів: один мільйон на один кілометр квадратний.

Слід сказати, у процесі реалізації варіантів вирішення в мережі наступного покоління, розроблення стандарту, виконання дослідження прийшли до висновків, що вказана вимога URLLC дуже складна для здійснення, у варіанті широкого застосування таких типів послуг. Такі послуги надаватимуться в обмеженій географічній плоскості, виділеному мережевому каналі, та інших обмежуваннях. Також є очікування, що в межах реалізації концепції мереж майбутнього до, такі послуги будуть реалізовані. Щоб виконати реалізацію аналізуються одночасно декілька шляхів дозволу по обмеженню:

1. Заміна технології фізичних рівнів і використання так званих «квантових комунікацій», які мають змогу реалізувати інші фізичні принципи передавання даних. Вони базуються на принципах квантових заплутаностей. Але такий метод є ще не повністю досліджений і перебуває на стадії розробки;

2. Застосування децентралізації мережі зв'язку, а також децентралізованість системи обчислення, та їх можливість гнучкого управління з використанням імплементаціїї ШІ у якості систем які виконують моніторинг а також управління.

В даний час можливо спостерігати еволюцію мереж в досліджуваному питанні наступного напрямку: змінювання інфраструктурного рішення на програмні структури які мають спільні розробки та почергове застосування технології ШІ від простих задач до поступово ускладнених. Важливим кроком розвивання мереж настане у момент переходу до квантових комп’ютерів та квантових комунікацій. В такому випадку керування усіма системами буде неможливе систем, які виконують реалізацію у замкнених циклах усі функціонали ШІ в мережах.

До того ж така система передбачає наявність необхідної кількості статистичних даних, де вона має можливість навчатись і в подальшому самостійно переходити на керуванням мережами на базі квантової комунікації

Вертаючись до теперішнього рівня технологічних рішень і прилеглих задач, в рамках задач використання ШІ у мережах зв’язку є дуже актуальні задачі:

- виконання ідентифікації трафіків з можливістю подальшого виконання прогнозу;

- задача ефективного розподілення обчислень;

- прогноз та ідентифікування перевантажень систем керування.

В такому разі задачі ідентифікування трафіку передбачає включення необхідності визначення чималої кількості видів сервісів, не виконуючи внесення надлишкових затримок, можлива робота по розширенню і налаштуванню алгоритмів до певних точок місцезнаходження мереж та сервісу [16].

Задача ідентифікування трафіків послуг. За винятком нового рішення по всіх рівням мережі, а також новіших рішень у сфері хмарної технології, які надають смогу деякою мірою виконати наближення до тої вимоги, яка пред'являє сервіси для інфраструктури, поставлені досить важливі задачі виконання модернізації логічного оброблення трафіків. В якості прикладу, в даний час якісні показники обслуговування надаються із використанням рішень DiffServ, та інших ТЕ рішень, до прикладу MPLS-TE, яке застосовує на базі протоколів резерву ресурси RSVP-TE. Але такі напрямки рішень передбачають наявність декількох недоліків при їх використанні у певних сервісах у мережі наступного покоління. Вагомим недоліком є брак динамічних можливостей управління в залежності від можливості змін профілей трафіку у мережі, у здатності виконання швидких переналаштувань алгоритмів обслуговувань доменів мереж які є підконтрольними, і в обмеженні наборів класифікаторів трафіку, в тих умовах, які виникають та є відмінними за ті вимоги якості які використовуються для IoT, та є достатньо критичним недоліками. У розглянутих стандартах, сумуються карти вимоги QoS до різного виду послуги. Вона визначає загальні відмінності, від голосу і відеопослуг, у eMBB та Доповнених реальностей. Наводяться основні поняття Слайсингу, що пов'язане з

представленням QoS. Але не всі послуги які визначаються в 3GPP. Слід зазначити що в кожному із напрямку можливо виконати визначення пріоритетності послуги. Як приклад у Інтернет додику варто виділити медико-санітарну сферу яка є основним напрямком, а інші вказати як нижчий пріоритет [17].

В такому випадку варто вказати, що пакет який у трафіку, якому необхідно дозволити певну якість обслуговувань, мають бути позначеними наперед певними мітками, а саме поля ToS в заголовку IP. В наш час поля ToS є поля "differentiated_services" що мають шість біт полів DiffServ_Code_Point та два біта полів "Explicit _Congestion_Notification". Механізми які є зручними для застосування, на меті якої є надання якостей обслуговувань в в наступних напрямках життєдіяльності: телефонний зв'язок, відеопередача даних та багато інших. Но їх використання для гарантування якості в обслуговуванні множини сервісів IoT є досить не зручним, а в деякій мірі недосяжне. При врахуванні вказаних недоліків в даний час, трафік який передається в IoT є змішаним з загальним потоком трафіку передавання інформації[18]. Передачу даних IoT можливо передавати з даними сервісів Video_on_Demand, а також як приклад серфінгу сторінок. Тривалий час є складність "великий заголовок з малими корпусами" що має вплив на роботу пристрою рівня передавання таким чином, що буфер пристрою є переривчастими, і має вплив рівні якості надання сервісу. Чимало досліджень показують вражаючі статистики екстенсивних і інтенсивних зростань трафіків, які генеруються різноманітними міні-пристроями. Неоднорідність трафіків, складність розрахунків їх збільшення, і швидких розрахунків зміни профілей зумовлює , що «Ручний» метод розрахунку втрачає власну актуальність. Щоб виконувалось забезпечення швидкого втручання систем контролю зростання трафіку, зміна профілей, необхідно дослідити інший метод розрахунків і управління. У випадку забезпечення оперативного реагування, мережа та пристрій комутації а також маршрутизація трафіків повинна включати в себе потрібний рівень обстругувань від «фізики» процесу. Потрібний рівень обстругувань від фізичного процесу [18] виконують технологічні рішення SDN та NFV.

Для швидкого спрацювання системи управління, бажано виконувати високоточні ідентифікації трафіку без безпосередніх втручань до потоків на рівнях передавання інформації у відповідності до внесень різного роду заміни у його профіль, і безпосередньо затримок.

Слід вказати, у випадку зареєстрованого потоку інформації-трафіку зарання, не має задачі розпізнавання, прикладом може бути – інформація VoIP чи IpTV. У магістерській аналізується задача виявлення потоку даних IoT у мережі SDN що має загальні потоки. Щоб вирішити вказані задачі мною зроблено комплексний огляд математичних методів класифікування, який враховує особливості вхідної інформації про потік, та особливості початкових вимог – функціонування систем що має режим «польоту». Результатом аналізу стало обрання підходу застосування нейронних мереж деяких архітектур. У відповідності поставленим задачам а також специфіку наборів даних на вході, обрано архітектури нейронних мереж.

2.2 Огляд архітектури модельної мережі та трафіку досліджуваних сервісів

Виконання дослідження методу який розглядається в роботі включає реалізацію програмного забезпечення (додатки контролерів SDN), виконання його тестувань. Для даних робіт необхідно реалізувати стенд Програмно-конфігурованої мережі, яка підтримується за допомогою північно-програмного інтерфейсу, та розгорнутого на ньому сервісу IoT, завдання якого виконувати моделювання роботи одної з програми Розумних міст, а також сервіси потокових відео. Присутність трафіків двох типів у мережах потрібне виконання тесту методу.

Основою стенду є модельна мережа SDN. Як трафіку генератора Інтернету Речей використовуватиметься рішення, розроблене у роботі [18]. Такі модельні мережі мають можливість виконувати досліди в галузях використання методів алгоритмів ШІ в рамках IMT_2020 для SDN. Приклад даної архітектури мережі зображено на рис.2.1.

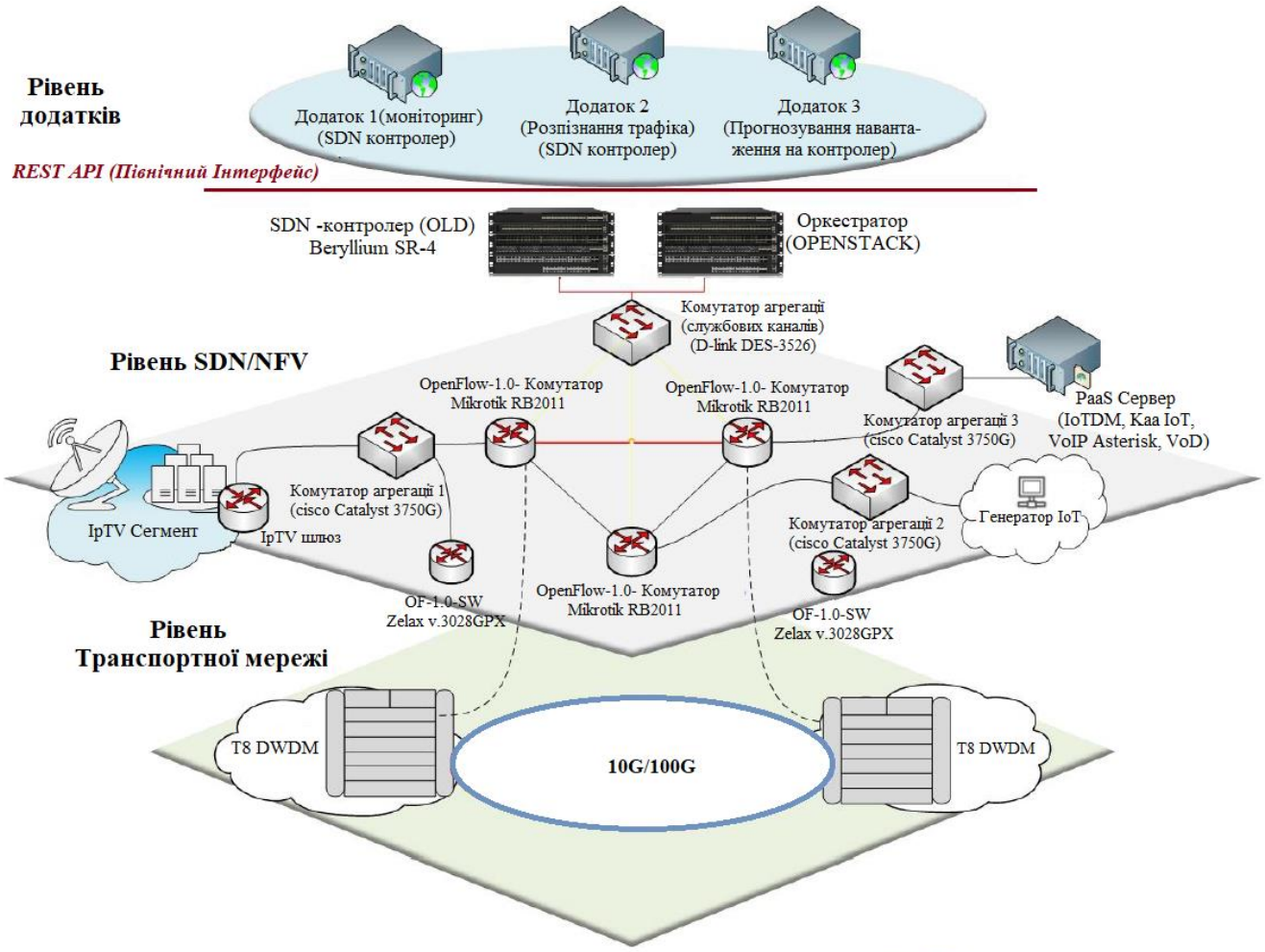


Рисунок 2.1 - Приклад архітектурної моделі мережі

Досліджуваний стенд представляє з'єднані у кільце трьох програмно-керованих комутаторів SDN, що реалізує протокол OpenFlow.

Обрано контролер який має відкритий код OpenDaylight SDN виконує розподілення навантажень що виникають у мережі по замовчуванню. Оскільки метою досліджуваного стенду є виконання дослідів, рівень контролю інтегрується звичайним комутатором Ethernet, що агрегує усі потоки які є службовими та забезпечє можливість доступності контролера. Слід вказати що мережі комутатора виконується доступ до північних рівнів контролера SDN.

В цьому разі, програма мереж розгортається на іншому сервері який функціонує із контролерами чи оркестраторами мереж з використанням даного комутатора.

На даному рівні комутатор дає можливість виконувати маштабування мережі із допомогою нарощування кількості комутаторів які підключаються до одного комутатора у мережі. Для того щоб підключити більшу кількість обладнання до комутатора конфігурованої мережі виконується послідовне підключення звичайного Ethernet-комутатора, який здійснює функції агрегації в такій топології.

Слід акцентувати увагу на один із контролерів, а саме OpenDaylight. Цей приклад рішення є ймовіно кращим який пропонується на ринку і має відкрите ПЗ контролеру програмно-конфігурованої мережі з урахуванням таких критеріїв: відкритість, архітектурне вирішення контролерів, велика функціональність, технічні підтримки, масштабоність та можливість швидкої обробки потоку OpenFlow. Базою програмного забезпечення такого контролера лежать модульні архітектури Java_Karaf, завдяки чому вдається забезпечити потрібну розширеність контролеру завдяки установки потрібних модулів які надалі мають змогу інтегрування у програмну шину контролера. Слід вказати що у даного виду контролерів є досить зручний а також наявні програмні інтерфейси які мають формат REST що дозволяють застосовувати додаток, який функціонує на верхньому рівні північного інтерфейсу. Слід відзначити, завдячуючи модульному рішенню архітектури цих контролерів є можливість застосувати необхідного програмного модуля (OSGi-контейнери) в першочерговій конфігурації. В такому випадку не має потреби у використанні додаткового модуля, що здійснює зниження навантажень на апаратні частини контролерів. Необхідно також зазначити, у випадку використання програмного модуля, який розроблено власноруч. У якості оркестратора мережі обрано рішення – OpenStack, у якого є особливість інтегрування з проектом OpenDaylight з використанням модуля «OpenStack Neutron».

Щоб виконати реалізацію трафіку-генератора звичайних сервісів IoT виконано дослідження реалізацію серверів з ОС Linux_Ubuntu яка має версію LTS на якому налаштовано сервіс IoTDM, реалізований компаніями Open_Daylight консорціуму Linux_Foundation. Такий сервер виконує організацію характеристик OneM2M по M2M взаємодії. Застосування рішення що пропонує Open_Daylight

обумовлюється тим, що до такої спільноти входять чимала кількість відомих світових компаній, які використовують такі розробки щоб застосувати у своїх продуктах.

Рисунок .2.1 демонструє на рівнях сервісу мереж декілька додатків, що виконують реалізацію досліджених і представлених у даній роботі методи. Ці програми розроблені на мові Python що мають відповідні фреймворки і виконують взаємодіють SDN контролером на рівні логіки з інтерфейсом REST, а по фізиці у мережі контролю.

Основне обладнання, що застосовується в момент реалізації стенду на базі якого виконуються дослідження, представлено в табл.2.1.

Таблиця 2.1 – Характеристики основного обладнання

№	Назва	Технічний опис
1	SDN-контролер (ODL) Beryllium SR-4	ЦПУ: Intel Xeon® E3-1220V2 ОЗУ: 16 Гб ОС: Linux Ubuntu 14.04 LTS Версія ПЗ: Opendaylight Beryllium SR4
2	OpenFlow 1.0 комутатор Mikrotik RB 201 1UI AS-RM	Mikrotik RB 201 1UI AS-RM Версія ПО: OpenFlow 1.0
3	Комутатор агрегації №1	Cisco Catalyst 3750G series PoE-24
4	Комутатор агрегації №2	D-Link DES 3526
5	Комутатор агрегації №3	Cisco Catalyst 3750G series PoE-24
6	Комутатор агрегації (службових каналів)	D-Link DES 3526
7	PaaS Сервер (IoTDM – сервер)	ЦПУ: Intel Xeon(R) E3-1220V2 ОЗУ: 12 Гб ОС: Linux Ubuntu 14.04 LTS Версія: Opendaylight Boron SR2
8	Д1 (моніторингу)* Д2 (Розпізнавання трафіку) Д3 (Прогнозування навантаження на контролер) *	ЦПУ: Intel Core i5 ОЗУ: 8 Гб ОС: Linux Ubuntu 16.04 LTS

Розгортання усіх програм виконувалось на одному фізичному сервері. В процесі застосування роль сервера виконує ПК , а при тестуванні вже стаціонарний ПК.

Передаюча інформація досліджуваного сервісу. У зв'язку з тим що модельні мережі які використовуються в магістерській є SDN мережа в реальних умовах, щоб виконати проведення реалізації а також тест необхідно зробити відтворення трафіку додатків і на рівнях передавання інформації також на рівні керування мережею. Справжній потік даних в мережі що моделюється, можливо відтворити або існуючим обладнанням IoT, чи генератором трафіку, який виконує емулює працездатність обладнання IoT. У моїй магістерській взято шлях застосування трафік-генератора, який виконує функціонування обладнання IoT, що реалізується у дослідженнях [18]. В дослідженнях [18] генератор трафіку застосовувався для виконання тесту платформи Інтернет речей IoTDM і аналіз взаємодій трафіків IP в SDN мережі, в тому числі аналіз функціонування динамічного автоматичного балансування навантаження в мережі яка контролюється.

Для даного аналізу з метою наближення функціонування емулятора до реальних масштабів виконано дослідження моделі на базі одного тривіального сервісу Розумний міст – системи контролю екологічного параметру з використанням трьох видів датчиків.

Дане модельне рішення у трафік-генераторі функціонує таким чином:

- у відповідності до вибраної архітектури – реалізація логічних дерев ресурсу;

- ініціалізованість пристрою IoT, в такому разі кожен з пристроїв в період авторизації до дерева ресурсів платформ передає відповідні запити до API_REST з даними про відповідну готовність датчиків.

- генерування потоку даних IoT. Під час процесу функціонування кожне обладнання виконує відсилку запитів кожної хвилини в напрямку дерева ресурсів, виконуючи його наповнення.

Результатом є отримання моделі, яка виконує імітацію роботи 960 апаратів IoT. А також, одночасно функціонувало з платформами 239 пристроїв. Для виконання задач у дипломі виконано огляд принципів розгортань трафіку-

генераторів та досліджено варіанти модифікації. Слід зазначити що раніше, такі трафіки-генераторів реалізовувались у кожному з агрегаційного сегменту, так для виконання задач, потік даних генераторів були розгорнуті лише на базі одного сегменту, який виконує агрегацію, а безпосередньо платформа IoT була розташована на базі іншого сегменту агрегації. Слід теж зазначити, що в атестаційній роботі не має потреби виконувати тест на різного роду протоколах IoT, в такому разі взяти генератор трафіку, який функціонує із платформою з використанням протоколу HTTP 2.0, в ньому містить інформація яка має формат JSON. Під кінець, для функціонування з ідентифікації потоку даних такий інтенсивний трафік немає необхідності використовувати, в якості експериментальних досліджень симуляція функціонування 230 засобів.

Платформа GUI яка була задіяна, виконує відтворення дерева ресурсу, рис.2.2, яке отримуємо при його реалізації програмними способами і слідуючими заповненнями інформацією, які пересилаються з пристроїв (в такому разі-симуляційними засобами).

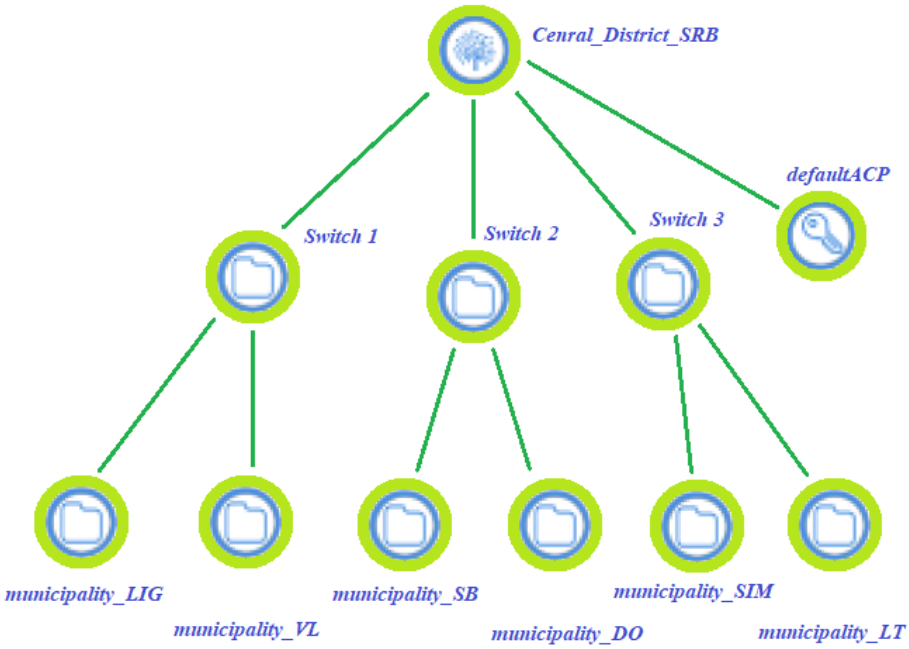


Рисунок 2.2 Приклад структури дерева ресурсів

2.3 Реалізація методу контролю і визначення потоку даних послуг у мережах

Наразі вже є велика кількість досліджень, що спрямовані на дослідження та розпізнавання різного роду трафіків у мережі зв'язку. Велика кількість підходів, які використовуються в більшій мірі основані на тимчасовому захопленні потоку даних та виконання аналізу заголовків трафіку. Даний метод передбачає декілька недоліків, серед яких: введення затримок в потоки, використання аналітичних модулів у вигляді додаткових апаратно-програмованих рішень рівня передачі інформації, складність даного засобу [19]. Варто приймати до уваги також недолік в масштабованості таких пристроїв. Щоб виконати глибоке дослідження наповненості потоку даних застосовують засоби глибокої перевірки пакеті. Мною аналізується використання аналітичної системи яка розташована на сервісному рівні архітектури SDN.

Під завершення системи можуть представлятись у якості апаратно-програмних комплексів, а також як програмні рішення. Загальна архітектура таких рішень представлена рис.2.3.

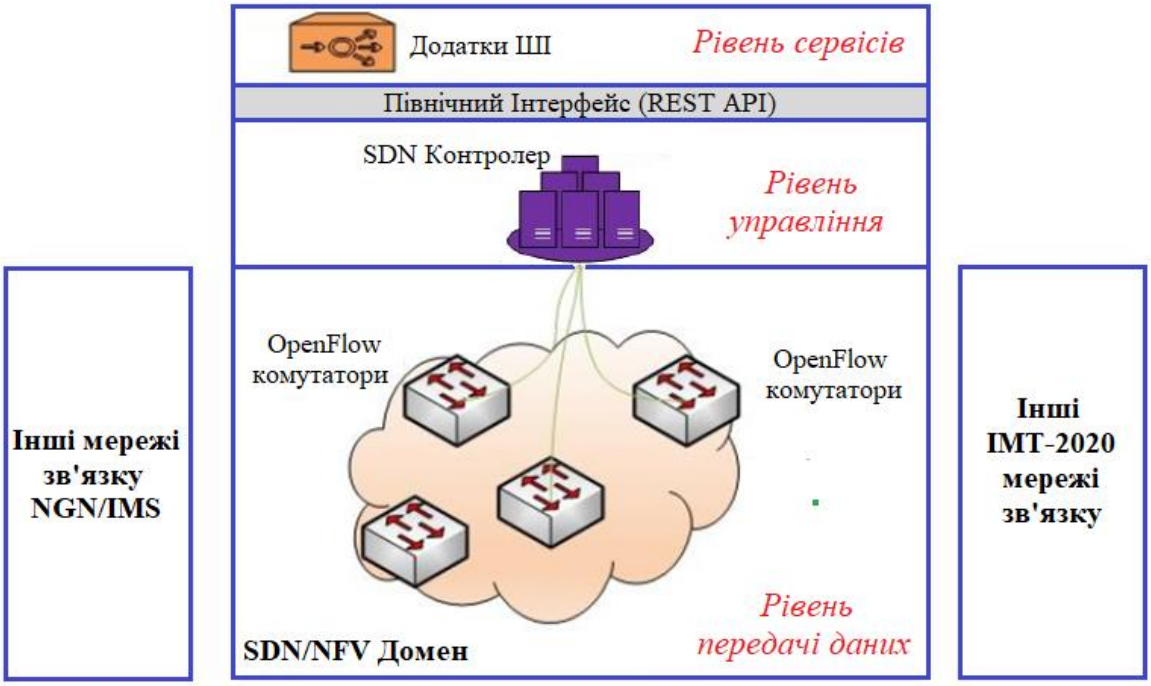


Рисунок 2.3 Архітектура рішень

В такому разі є можливість системі бути незалежною в передаванні інформації і тим паче інтеграції з рівнем передавання інформації. У випадку аналітичності системи усі засоби і потоки мають бути «цифровими об'єктами», які мають перелік характеристик та певний функціонал, який наданий певним набором методів. Даний рівень образів надає змогу виконати реалізацію систем аналітики, що функціонуватиме з інформацією про трафік який має режим «польоту». В даному разі, система надає дозвіл не виконувати внесення деякої кількості затримок у потік даних, та будь-яким способом виконувати заміну його активності. Слід зазначити, система «виконує спостереження» за активностями трафіку та вимальовує «основну картину», а саме яка ситуація в мережі яка підконтрольна мережі.

Досліджувані дані. Виконуючи врахування архітектури яка розглядається у роботі, на сам перед програмної реалізації, яка функціонує на рівні сервісу із контролером SDN та REST, вхідна інформація виконує формування на базі тої інформації, яку система отримує через північний контролер і оркестратор. Зважаючи на те, що досліджується активність трафіку і визначення трафіку IoT не тільки на передавальному рівні, система у змозі виконати запит інформації таблиць трафіку з комутаторів SDN які підконтрольні. Аналізуючи дані які наведені у 2-х частинах таблиці: Match_Field та Actions, слідує висновок, на базі можливо створити модель потоку. Приклад такої структури таблиць які є у комутатора наведено на рис.2.4.

Match Field						Action
#	In Port	TimeStamp	.....	Flow Statistics		
				Byte Count	Packet Count	
1				*B_count_1*	*P_count_1*	

Рисунок 2.4 Будова таблиці потоків SDN

Рисунок відображає червоними квадратами інформацію, яка застосовуватиметься для виконання формувань моделі про потоки у мережі, які досліджуються.

Вадливою особливістю таких даних є наступне, на базі лічильнику "Byte_Count" та "Packet_Count" немає можливості в точності виконати визначення точного розміру пакета в трафіку. Через те що за один відлік часу лічильники мають можливість бути рівними: "Byte_Count" -1501, "Packet_Count" - 3. У відповідності, на базі таких даних івідсутня можливість з повною точністю виконати визначення довжини кожного із пакету, що зареєстрований у трафіку за такий період часу:

$$\Delta T = 1 \text{ [c]}.$$

Варто вказати, відображенням на лічильнику є сумарні відрізки параметрів [Byte _Count», «Packet_Count»]. Але, окрім інформації лічильників в таблицю потоку є ще такий параметр як "Time_Stamp", що надає можливість виконати оцінку у кожному моменті часу миттєвого результату [Byt_Count_delta] та [PacketCount_delta]. В такому разі, за випадковий термін T, отримавши переліки значення, є можливість зібрати набір інформації з єдиною структурою інформації, в якому відлік показує значення [Byte_Count_delta] а також [Packet_Count_delta]. Такі відліки як [Byte_Count], [Packet_Count], [Time_Stamp] виконують формування шлху щосекундні запити які подаються на контролер SDN з використанням програмного інтерфейсу REST_API. Формована структурно на базі питань DataSetRQ (2.1) а також формула (2.2) його зміна в потрібний формат DataSetML із значеннями (2.3) представлено далі.

Допустимо, Packet_Count_delta-PC_{delta},

Byte_Count_delta-BC_{delta}, а

Time_Stamp_deltas-TS=1[sec]=const, тоді:

$$DataSet_{RQ} = \begin{matrix} [\text{Time\_Stamp}] & [\text{Byte\_Count}] & [\text{Packet\_Count}] \\ \text{Time\_Stamp}_{11} & \text{Byte\_Count}_{12} & \text{Packet\_Count}_{13} \\ \text{Time\_Stamp}_{21} & \text{Byte} - \text{Count}_{22} & \text{Packet} - \text{Count}_{23} \\ \cdots & \cdots & \cdots \\ \text{Time\_Stamp}_{N1} & \text{Byte\_Count}_{N2} & \text{Packet\_Count}_{N3} \end{matrix} \quad (2.1)$$

$$\begin{cases} \text{BC\_delta}_{N2} = \text{ByteCount}_{N2} - \text{ByteCount}_{(N-1)2}, \quad if \;\; N \geq 1 \\ \text{PC\_delta}_{N2} = \text{PacketCount}_{N2} - \text{PacketCount}_{(N-1)2}, \quad if \;\; N \geq 1 \end{cases} \tag{2.2}$$

$$\text{DeltaSet}_{ML} = \begin{matrix} [\text{TimeStamp}] & [\text{ByteCount}] & [\text{PacketCount}] \\ TS & \text{BC\_delta}_{12} & \text{PC\_delta}_{13} \\ TS & \text{BC\_delta}_{22} & \text{PC\_delta}_{23} \\ \dots & \dots & \dots \\ TS & \text{BC\_delta}_{N2} & \text{PC\_delta}_{N3} \end{matrix} \tag{2.3}$$

В такому випдку розрахування сумарного значення параметру за встановлені проміжки часу виконуються по наступним формулам (2.4, 2.5):

$$\text{ByteCount}_{\Delta T} = \sum_{N=1}^{N=\Delta T/TS} \text{BC\_delta}_{N2} \tag{2.4}$$

$$\text{PacketCount}_{\Delta T} = \sum_{N=1}^{N=\Delta T/TS} \text{PC\_delta}_{N2} \tag{2.5}$$

Як було зазначено вище, на вирішення вищеописаної задачі було виконано комплексне дослідження різних математичних методів класифікування із врахуванням особливості вхідного, аналізованої інформації про потоки, і навіть специфіки вимог які на початку, а саме робота в режимі «в повітрі. По результатам дослідження за основу взято застосування нейронних мереж [16]. У відповідності до поставлених задач та специфіки вхідної інформації DataSetML взята певна архітектура нейронної мережі.

Нейронна мережа та Deep Learning. Аналізуючи розвиток сучасного стану можливо зробити висновок що в наш час є досить велика кількість різного виду нейронних мереж. Сучасним напрямком класифікації є спосіб заснований на описуванні об’єктів на базі ознак, де кожен обєкт має характерні набори ознак.

Но певні види даних є відкриті ознаки не можуть надати потрібної точності при класифікації, прикладом є відтінок точок малюнка або дискретних сигналів звуку. Причиною є те , що така інформація містить невидиму ознаку. Deep_Learning це набір різного роду алгоритмів M2M, які надають можливість моделювати абстракції, тобто, здійснювати виділення з інформації невидимі ознаки, але в даному випадку нейронні мережі мають більше, чим два невидимих

шари. В такому разі рахуються особливості потоку даних а також та його характеристики, взято нейронну мережу із використанням Deep_Learning.

Реалізація та виконання навчання нейронної мережі виконувалось на базі мови Python з застосуванням різного роду бібліотек і фреймворків. Для виконання задач в якості ШНМ взято рекурентну НМ яка має додатковий шарами LSTM. LSTM вважається універсальною, оскільки має необхідну кількість нейронів. Також вона має можливість спроможна виконувати різної складності обчислення, що відсутнє у звичайних ПК.

Оскільки у обраній архітектурі застосовується «навчання з вчителем», варто сформувати набори інформації для навчань які мають промарковані дані, на виконати збереження. Щоб виконати навчання НМ Data_Set_ML який є вхідним, перетворений у Data_Set_ML_train за допомогою введення нового стовпця інформації. В кожному ряді є ідентифікатор статистичної добірки. Для виконання більш ефективного навчання розрізнення чималого типу потоку інформації, навчальний набір інформації доцільно розширити, зробивши певні відмітки статистичної вибірки потоку інформації, для прикладу - IoT. Тоді , структура навчальних $\text{DataSet}_{\text{ML}train}$ має вигляд (2.6):

$$DataSet_{ML} = \begin{matrix} [TypeOfTraffic] & \{TimeStamp] & [ByteCount] & [PacketCount] \\ IoT & TS & BC_{delta_{12}} & PC_{delta_{13}} \\ IoT & TS & BC_{delta_{22}} & PC_{delta_{23}} \\ \dots & \dots & \dots & \dots \\ Video & TS & BC_{delta_{N2}} & PC_{delta_{N3}} \\ others & TS & BC_{delta_{(N+1)2}} & PC_{delta_{(N+1)3}} \end{matrix} \quad (2.6)$$

Мережам LSTM надаються дані які мають фіксовану довжину, в такому разі інформація ділиться на частки які складаються з 200 рядків або 10 с.

Теги які мають тип трафіків змінюються на унітарні коди із допомогами вмонтованих до бібліотеки Python необхідної функції. Дані поділяються по наступних особливостям які поділяються на тренувальний а також тестовий набір у співвідношенні 9:3.

До архітектури входять 2 рівні типу LSTM та 2 рівні типу RNN. Рекурентні нейронні мережі, кожна з яких має 7 схованих нейронів.

Характеристики тренувань ШНМ:

- оптимізатори: Adam;
- чисельність епохи навчань: 40;
- чисельність прикладів на ітерацію: 1024;
- бистрота навчання 0,0015.

Рисунок 2.5 відображає вхідні шари, які складаються із кількості 3-х нейронів, до входу яких передавались певна інформація Data_Set$_{MLtrain}$. Перші три є першим рівнем, який складається з нейронів LSTM. До другого рівня входять нейрони які мають тип LSTM і пов'язаний з першим та третім. Наступні два рівня ШНМ застосовані на нейронах RNN.

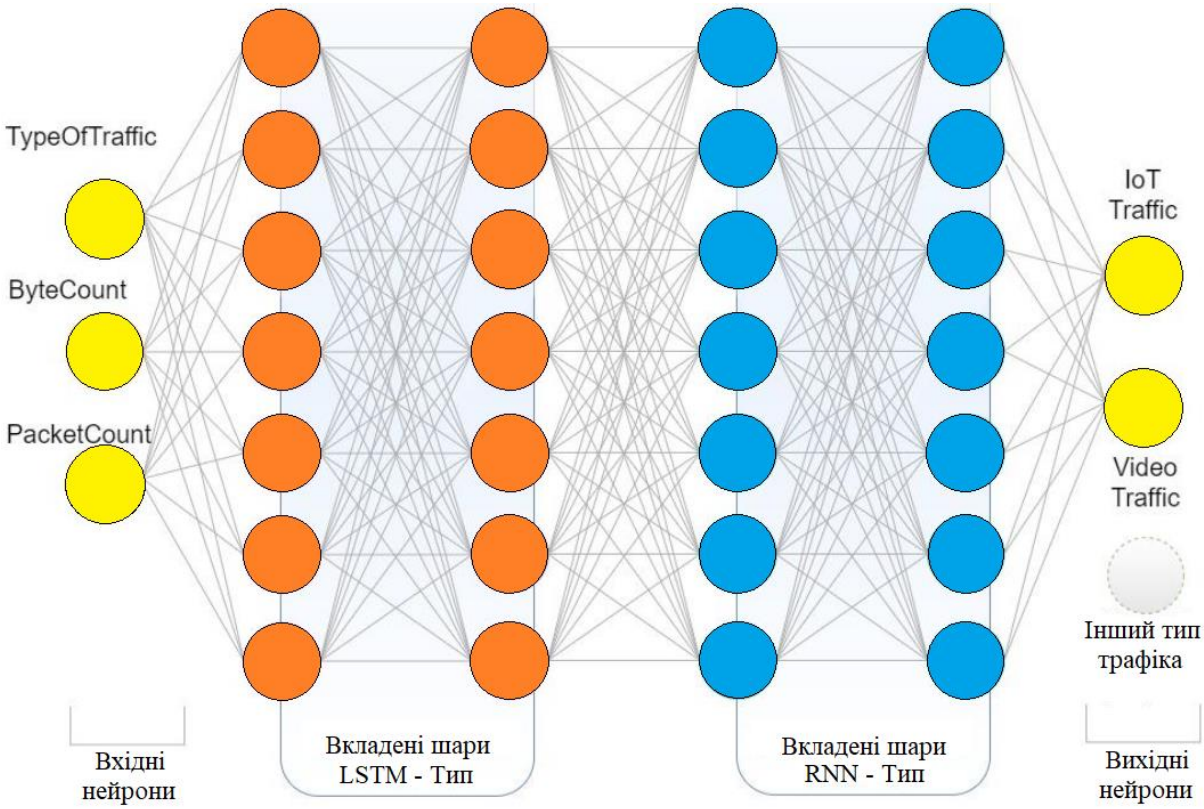


Рисунок 2.5 Загальна архітектура ШНМ

Вихідні нейрони це двійка нейронів, які відтворюють результати роботи нейронних мереж. З цього випливає, у випадку пов'язання DataSet$_{MLtrain}$ з чималим

переліком типів поміченої статистики потоку інформації, в такому разі перелік вихідних нейронів виконує зростання.

Є деяка грань, при перевищенні, для виконання навчання НМ є необхідність накопичувати нейрони в кожному шарі.

Модель. Метод який аналізується застосований під видом модуля програми на мові Python для використання у системі аналітики, що застосовані на базі MVC патерну. Таке ПЗ реалізується зверзу API контролера SDN а також оркестратор аналізованої мережі.

Для організації наборів інформації ШНМ вказано особливі умови:

- потік інформації який не відноситься до дослідження, яке виконується, має бути включений та не надходити в SDN-секції;

- мережі повинні модифікуватись так щоб мати змогу зробити визначення потоку інформацію з інформацією який зроблено в таблицях потоків інформації комутатора. Наступним кроком додатку присвоюється номер потоку у таблиці потоків і далі ПЗ виконувала певну дію на вимогу статистичної інформації на базі північного інтерфейсу контролера SDN та їх наступного залишання для виконання навчань ШМ. Таким способом виконується формування набір інформації для навчань для потоку інформації «Video on Demand», у своїй генератори трафіку ВВ були дезактивовані та лишались умови які проговорені раніше.

По виконанню того як виконано збір потрібної інформації із сегменту SDN, аналітичні програмні модулі запускають відповідні алгоритми для первинних обробок інформації для виконання підготовок їх як навчальних наборів ШНМ. Після виконання успішних навчань досліджувана ШНМ залишалася у свому стані. Переходячи до наступної стадії, у разі «бойових» розпізнавань трафіку, цей збережений стан ШНМ можливий у подальшому застосуванні наступним ПЗ, шо виконує створення набору для тесту інформації після призначення цифри потоку, який аналізується в таблиці потоків.

Рисунок 2.6 відображає архітектуру сегмента та аналізуючий сценарій відтворень.

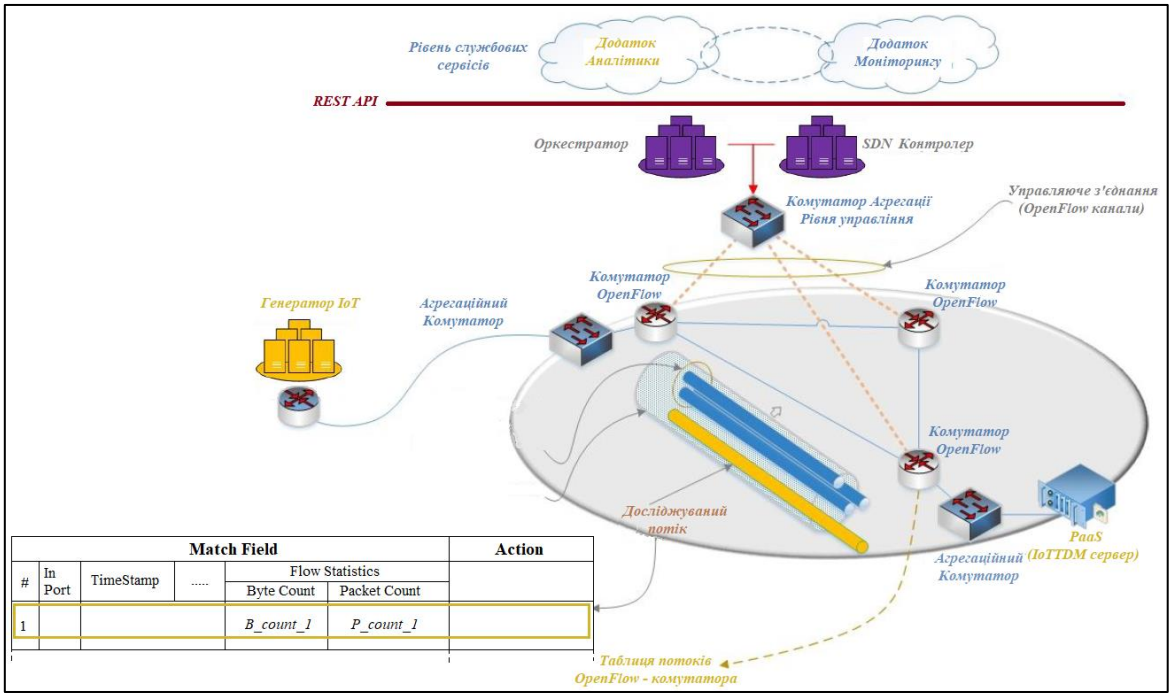


Рисунок 2.6 - Архітектура частини мережі яка анілізується по взятому сценарію

Даний рисунок відображає всі елементи які є головні у модельному стенді SDN із сервером, а також відтворено аналізуючий сценарій з об'єктом аналізу і характеристиками.

2.4 Результати практичних досліджень

Як вже вказувалось, для виконання навчання ШНМ необхідно виконано деякого дотримання певного ряду умов: трафік який був стороннім вимкнено (дезактивовано), виконано модифікацію мережі в такий спосіб, щоб можливо було виконати детектування трафік інформації з трафіком IoT або Video_on_Demand у таблиці SDN для комутаторів. Генератор трафіку IoT виробляв функціонування 230 засобів IoT. Кожен засіб виконував генерацію пакету HTTP 2.0 і в основі пакета висилав значення датчиків. Наступним кроком, модулям ПЗ надавались точні номера потоків у таблицях потоків комутатора SDN, далі програма виконувала запуск процесу зберігання інформації. Паралельно виконується формування набір для навчання даних для потоку інформації Video_on_Demand, в даному випадку

застосоване вже ПЗ для генерування потоку інформації по запиту (VLC). Результат - отримання $Data\_Set_{ML\_train}$, який був сформований що зібраний з двох за допомогою кількох промаркованих наборів. Наступним кроком $Data\_Set_{ML\_train}$ йшов на вхід НМ, конфігурацію такої мережі розглянуто раніше.

Наступним кроком отриманні $DataSet_{MLtrain}$ виконані побудови діаграми розкидування значення. Дана діаграма представлена рис.2.7 На ній візуально можливо виконати виділення кількох кластерів розділення точок, в полі яких більшостю є певне значення середніх значень довжини пакетів в трафіку. Можливо привести такий приклад, в трафіку IoT, у більшій кількості випадків, середньою довжиною пакета є орієнтовно 401 байт, в такому випадку параметри можуть змінюватись. Теж варто побачити викиди значень які мають в середньому довжину пакета 4690 байт.

Під кінець є помітним викид значення в статистичних рядах метаданих у відео потоці, до того ж у більшій кількості випадків середньою довжиною пакетів складає орієнтовно 1340 байт.

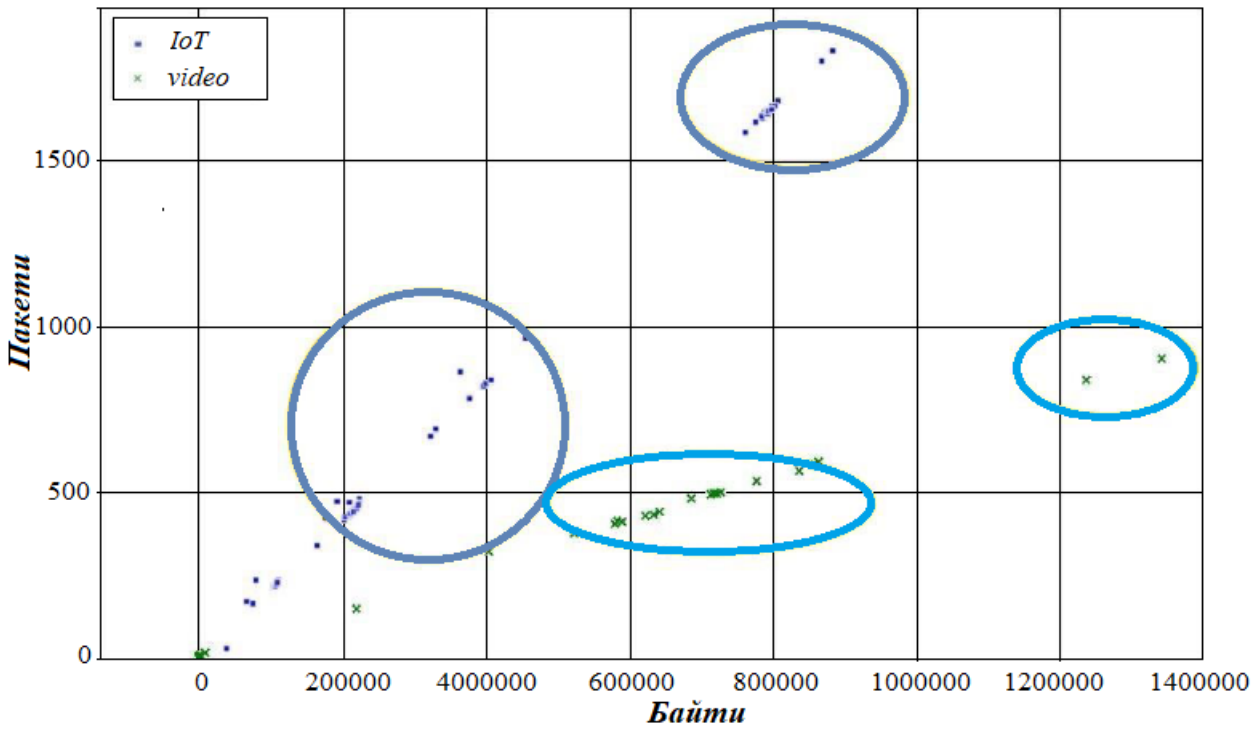


Рисунок 2.7 - Отримана діаграма розкиду множини $Data\_Set_{ML\_train}$

Після формування навчального набору інформації, відповідно алгоритмом, проаналізованим раніше у магістерській, програмний модуль аналізуючого додатку, що застосовує ШНМ, починав виконувати функціонування. Під час періоду навчань НМ додатком можливо було бачити певні параметричні характеристики:

- Train_Accurancy (Точність навчань);
- Test_Accurancy (Точність виконання тестів ШНМ);
- Train_loss (Похибки в момент навчання);
- Test_loss (Похибки в момент виконання тестів ШНМ).

Графічне зображення яке відображає такі характеристики у періоді виконання епохи навчань, зображено рис.2.8. На додаток, на рис.2.9 показано особливості наближень (виконано збільшення масштабу частин графіку, рис.2.8) щоб детальніше відобразити відмінності між отриманими графіками.

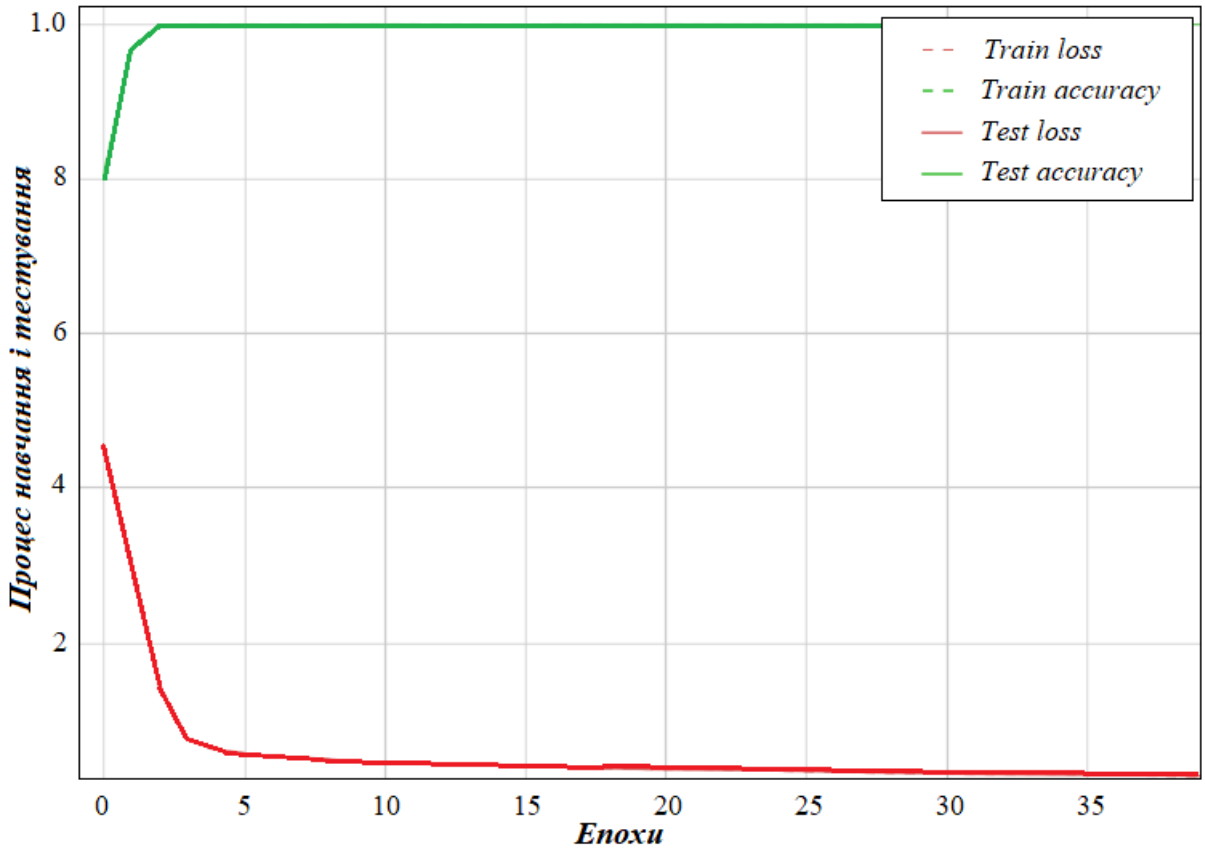


Рисунок 2.8 Криві періоду навчання і тесту ШНМ

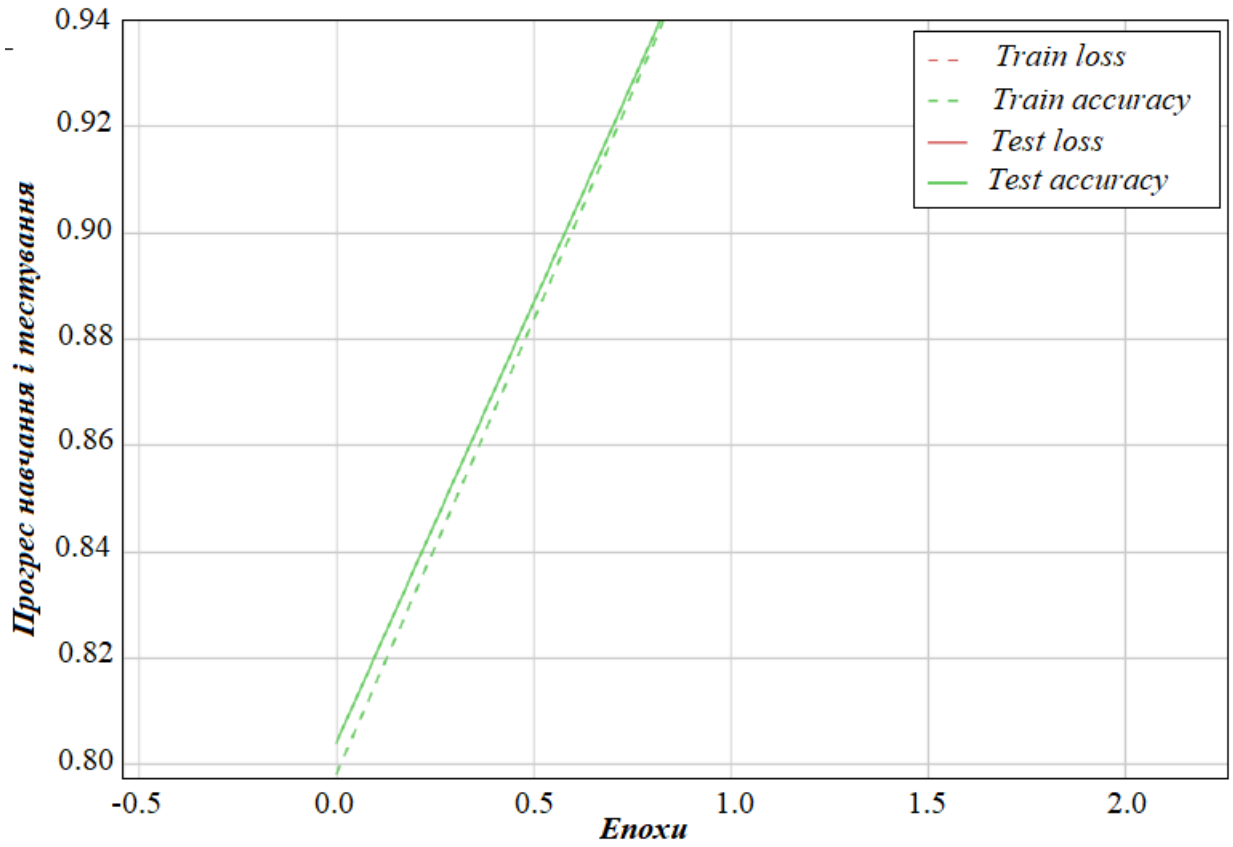


Рисунок 2.9 Графічне зображення кривих навчання та тесту ШНМ

Окрім того що приведені вище графіків на рис. 2.8 та рис.2.9, було виконано розрахунок матриці протиріччя ШНМ навчань штучних нейронних мереж. Результат такої матриці представлено у рис.2.10.

Рис. 2.8 і рис.2.9 відображають дії навчання мережі, результат показує, для вказаної задачі ШНМ яка аналізується, виконала дію навчання успішно. Результатом дій навчання НМ а також перевірки її функціональної можливості з використанням тестових наборів інформації у стані після навчання застосована нейронна мережа яка має можливість виконати ідентифікацію потоків IoT, які генеруються, ймовірність 98,6 %.

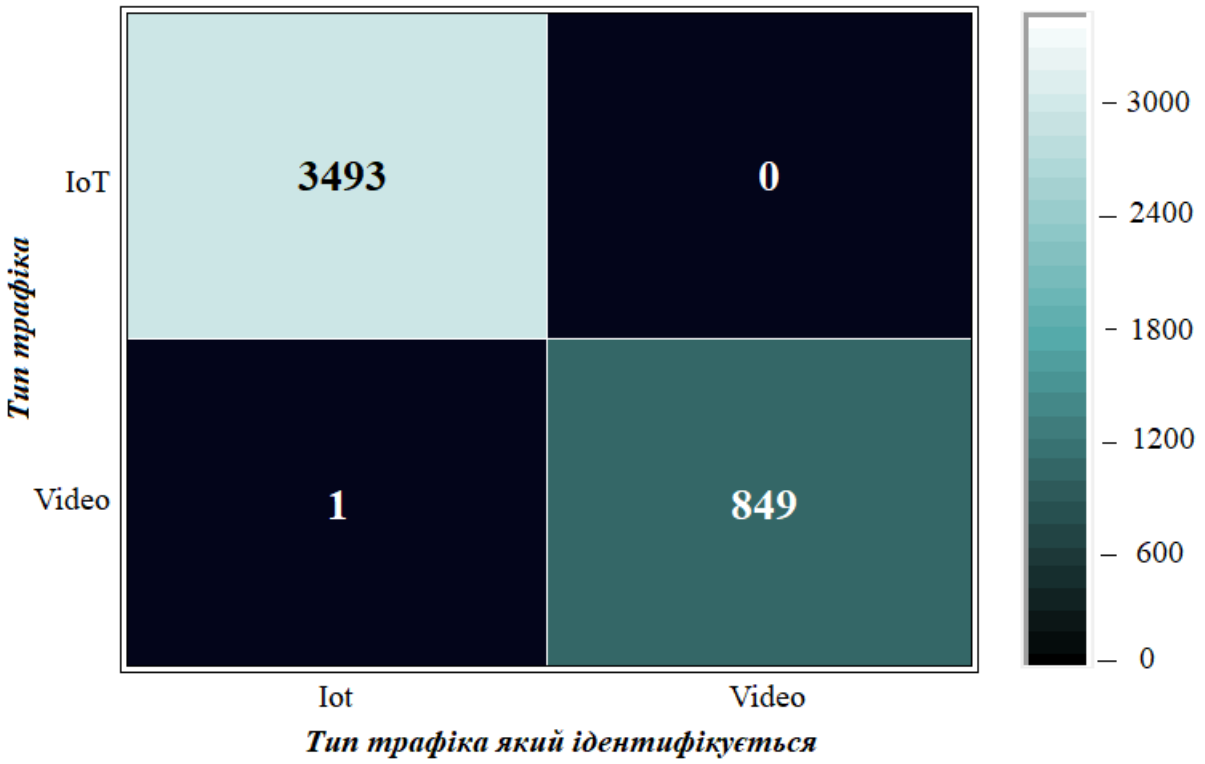


Рисунок 2.10 - Відображення матриці протиріч

У відповідності також потік, який генерує відео. Із використанням матриць протиріччя, рис.2.10, можливо побачти, що мережа одного разу припустилась помилки, виконавши ідентифікацію відео потоку, як трафік IoT. Слід вказати, досліджувана архітектура є доволі ефективна у використанні вирішення задач яке було поставлено. У випадку меншого значення нейронів у рівнях, мережа функціонує не досить стійко, здійснюється чимало похибок, також під час дій навчання, а не лише по результатам функціонування. У випадку більшого значення нейронів – досліджувана мережа неефективна, бо вимагає значно більшої ваги обчислювальних ресурсів. Необхідно звернути увагу, у випадку з перебільшенням нейронів у шарі –- мережі перетворюються у режими «перенавчання».

На сам кінець слід відзначити, що ймовірність розпізнання потоку інформації, яку ми отримали, є достатньо великою. У разі зросту кількості типів потоків інформації (якщо є збільшення, в такому разі виконується формування нового, розширеного набору даних щоб виконувати навчання мережі) така можливість може бути знижена, але є допустимими рамками межах (75-85%).

РОЗДІЛ 3 КОНТРОЛЬ ТА ПРОГНОЗУВАННЯ НАВАНТАЖЕННЯ СКЛАДОВИХ МЕРЕЖІ В УМОВАХ ОБМЕЖЕННЯ ЯКОСТІ ОБСЛУГОВУВАННЯ ПОСЛУГ

3.1. Огляд складності контролю та моніторингу складових мережі SDN

Для забезпечення високих вимог до якості послуг, зокрема в мережах класу з ультрамалими затримками та підвищеною надійністю, необхідно забезпечити стійкість систем управління в мережах IMT-2020 та наступних поколінь. У системах SDN ключову роль у керуванні відіграють контролери, які здійснюють логічну та протокольну взаємодію. Вони також працюють із протоколами OF-Config і OpenFlow для керування SDN-комутаторами. У віртуалізованих мережевих функціях аналогічну роль виконують оркестратори NFV. У цій дисертації приділено особливу увагу питанням моніторингу та прогнозування навантаження на контролери SDN.

За час розвитку програмно-конфігурованих мереж було проведено значну кількість досліджень, включно з комерційними компаніями, що займаються розробкою та впровадженням SDN-рішень, зокрема стрес-тестуванням. Основною метою таких досліджень зазвичай є визначення меж роботи SDN-контролерів за кількістю одночасно обслуговуваних потоків OpenFlow, порівняння переваг різних архітектур контролерів, а також визначення найбільш ефективного контролера з розширеним функціоналом. Таким чином, знання про навантаження на контролер SDN дає можливість оцінити загальну працездатність мережі.

Прогнозування роботи контролера програмно-конфігурованої мережі є одним із ключових завдань, зважаючи на його важливість. Один із методів дослідження цього питання полягає у створенні спеціалізованого програмного забезпечення для операційної системи, на якій працює контролер SDN, наприклад, Linux Debian або Ubuntu. Це програмне забезпечення взаємодіє з ОС через системні виклики, щоб отримувати інформацію про параметри апаратних ресурсів (такі як процесорні потоки, їхнє навантаження, оперативну пам'ять, постійну пам'ять,

активність ядер процесора тощо). Зібрані дані можна обробити та передати на аналітичну платформу через мережевий API, яка може бути віддалено розміщена щодо ОС із розгорнутим SDN-контролером. Проте, така архітектура має кілька недоліків:

Залежність від апаратних компонентів контролерів SDN.

Залежність від операційної системи, що може створювати так звану «вендорську прив'язаність» до постачальників ПЗ і бібліотек.

Постійні оновлення системних утиліт, які можуть впливати на стабільність та безпеку системи.

Додаткове навантаження на процесор контролера SDN через нові процеси.

Складнощі з перенесенням та швидким впровадженням цього моніторингового ПЗ на інші рішення.

Використання стороннього ПЗ, такого як Zabbix, може обмежувати функціонал і не враховувати специфіку роботи SDN-контролера.

З огляду на розвиток технологій штучного інтелекту в системах управління мережею, моніторингове ПЗ повинно відповідати вимогам для цього класу службових додатків. Крім того, є ймовірність, що некоректна робота моніторингового ПЗ може вивести з ладу контролер SDN, якщо його розмістити на одному рівні з ОС контролера.

У зв'язку з вищезазначеними проблемами, постає необхідність застосування нового методу аналізу та прогнозування навантаження на SDN-контролери. У цій роботі досліджується метод, що базується на аналізі метаданих службових потоків, які надходять від OpenFlow-комутаторів до SDN-контролера. Цей підхід дозволяє вирішити більшість проблем, зокрема залежність від апаратної частини та операційної системи.

Враховуючи специфіку даних протоколу OpenFlow, було виконано огляд існуючих математичних методів прогнозування, проведено багатопараметричний кореляційний аналіз, який виявив зв’язки між активністю службових потоків OpenFlow та навантаженням на апаратні ресурси SDN-контролерів. Також було доілідено додатковий програмний модуль для сервера моніторингу та

прогнозування службових потоків SDN з використанням мови Python і бібліотек Pandas, NumPy та TensorFlow. Результати дослідження підтвердили можливість успішного застосування цього методу прогнозування навантаження на SDN-контролери в мережах зв'язку IMT-2020 та майбутніх поколінь.

3.2 Застосування методу прогнозування навантаження на контролер мереж SDN із застосуванням ШІ

Аналіз методу та моделі, що досліджуються.

На сьогоднішній день в розвитку технологій програмно-конфігурованих мереж існує безліч досліджень, спрямованих на вивчення та реалізацію різних методів для тестування контролерів, включаючи стрес-тести. Згідно з проведеним аналізом, у цій роботі розглядається можливість контролю навантаження на SDN-контролер шляхом моніторингу та інтелектуального аналізу метаданих лише певних груп службових потоків OpenFlow. Для перевірки ефективності цього підходу була використана модельна програмно-конфігурована мережа з сервісом «Video-on-Demand» на рівні передачі даних у SDN.

Аналітичні системи (далі – службові програми мереж), які також включають модулі прогнозування активності службових потоків OpenFlow, реалізовані на мові програмування Python у вигляді веб-сервера, побудованого за моделлю проектування MVC (Model-View-Controller). Це програмне забезпечення функціонує поверх північних API контролерів SDN модельних мереж, які підлягають дослідженню.

У якості контролера SDN у цій модельній мережі використано OpenDaylight Beryllium SR4. Рівні передачі реалізовані на базі комутатора Mikrotik, який підтримує протокол OpenFlow версії 1.0. На рисунку 3.1 представлена загальна архітектура сегменту. Крім основних елементів модельної мережі, на цьому малюнку також відображені об'єкти дослідження та їх параметри.

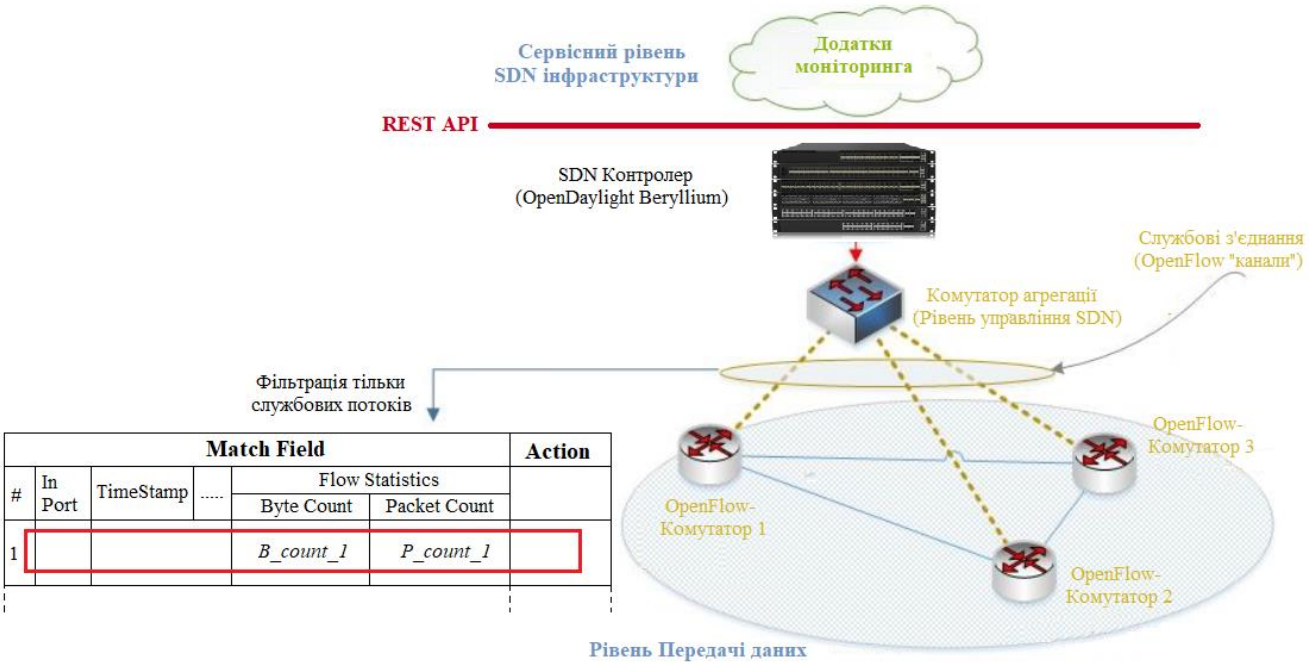


Рисунок 3.1 - Архітектура сегменту з об'єктом дослідження

Для створення досліджуваних наборів даних додаток моніторингу надсилав REST-запити кожну секунду до контролера програмно-конфігурованої мережі через північний API. Наступним етапом була додаткова фільтрація отриманих таблиць потоків, щоб надсилати на подальшу обробку лише дані по службових потоках OpenFlow. Оброблений набір даних проходив додаткову обробку, необхідну для формування набору даних, що дозволяє створити аналітичну модель, загальна характеристика якої буде розглянута в наступному підрозділі.

Раніше було виявлено можливість складання метаданих потоків на основі двох ключових частин таблиці потоків SDN-комутатора: Match Field та Actions. Важливою особливістю таких даних є те, що використання лічильників "Byte Count" та "Packet Count" не дозволяє точно визначити довжину пакетів у потоках. Наприклад, в один момент часу лічильник може показувати: «Byte Count» - 1500, «Packet Count» - 3. Отже, на основі таких даних неможливо точно визначити довжину кожного пакета, зареєстрованого в потоці за певний проміжок часу.

$$\Delta T = 1 \text{ [c]}.$$

Слід зазначити, що лічильники відображають загальні значення параметрів «Byte_Count» та «Packet_Count». Проте, окрім даних, які надають ці лічильники, в

таблицях потоків SDN-комутатора є важливий параметр «Time_Stamp», який дозволяє оцінити миттєві значення «ByteCount_delta» та «PacketCount_delta» в необхідний проміжок часу. Таким чином, в будь-який момент часу T, за наявності значень «Byte_Count», «Packet_Count» і «Time_Stamp», можна сформувати інформаційний набір зі встановленою структурою, де кожен відлік відображає миттєві значення «ByteCount_delta» та «PacketCount_delta». Значення «Byte_Count», «Packet_Count» і «Time_Stamp» формуються шляхом миттєвого запиту до контролерів SDN через REST API. Структура формованого на основі запитів DataSetRQ із «сирими» даними (3.1) та формула (3.2) для його перетворення в потрібний формат DataSetML з миттєвими значеннями (3.3) наведені нижче.

Нехай, Packet_Count_delta - PC_{delta},

$Byte\_Count_{delta} - BC_{delta}, a$

$Time\_Stamp_{deltas} - TS = 1[sec.] = const$, тоді:

$$DataSet_{RQ} = \begin{matrix} [TimeStamp] & [ByteCount] & [PacketCount] \\ TimeStamp_{11} & ByteCount_{12} & PacketCount_{13} \\ TimeStamp_{21} & ByteCount_{22} & PacketCount_{23} \\ \cdots & \cdots & \cdots \\ TimeStamp_{N1} & ByteCount_{N2} & PacketCount_{N3} \end{matrix} \quad (3.1)$$

$$\begin{cases} BC_{delta_{N2}} = ByteCount_{N2} - ByteCount_{(N-1)2}, & if\ N \geq 1 \\ PC_{delta_{N2}} = PacketCount_{N2} - PacketCount_{(N-1)2}, & if\ N \geq 1 \end{cases} \quad (3.2)$$

$$DataSet_{RQ} = \begin{matrix} [TimeStamp] & [ByteCount] & [PacketCount] \\ TS & BC\_delta_{12} & PC\_delta_{13} \\ TS & BC\_delta_{22} & PC\_delta_{23} \\ \cdots & \cdots & \cdots \\ TS & BC\_delta_{N2} & PC\_delta_{N3} \end{matrix} \quad (3.3)$$

В такому випадку розрахунки сумарного значення параметру у встановленому проміжку часу виконується за наступними формулами (3.4, 3.5):

$$ByteCount_{\Delta T} = \sum_{N=1}^{N=\Delta T/TS} BC\_delta_{N2} \quad (3.4)$$

$$PacketCount_{\Delta T} = \sum_{N=1}^{N=\Delta T/TS} PC\_delta_{N2} \quad (3.5)$$

Частково дані структури даних, відображені вище, схожі з тими, які були наведені в рамках методу ідентифікації трафіку упопередньому розділі цієї роботи, проте поточні структури описують потоки рівня управління в мережах, що програмно-конфігуруються. У той час, як у другому розділі досліджувалися потоки на рівні передачі даних користувача.

3.3 Дослідження працездатності методу

У ході моїх досліджень першочергово потрібно було перевірити наявність взаємозв'язку між активністю потоку OpenFlow та змінами в навантаженні апаратної частини контролера SDN. Принципова схема експериментальної установки показана на рисунку 3.2.

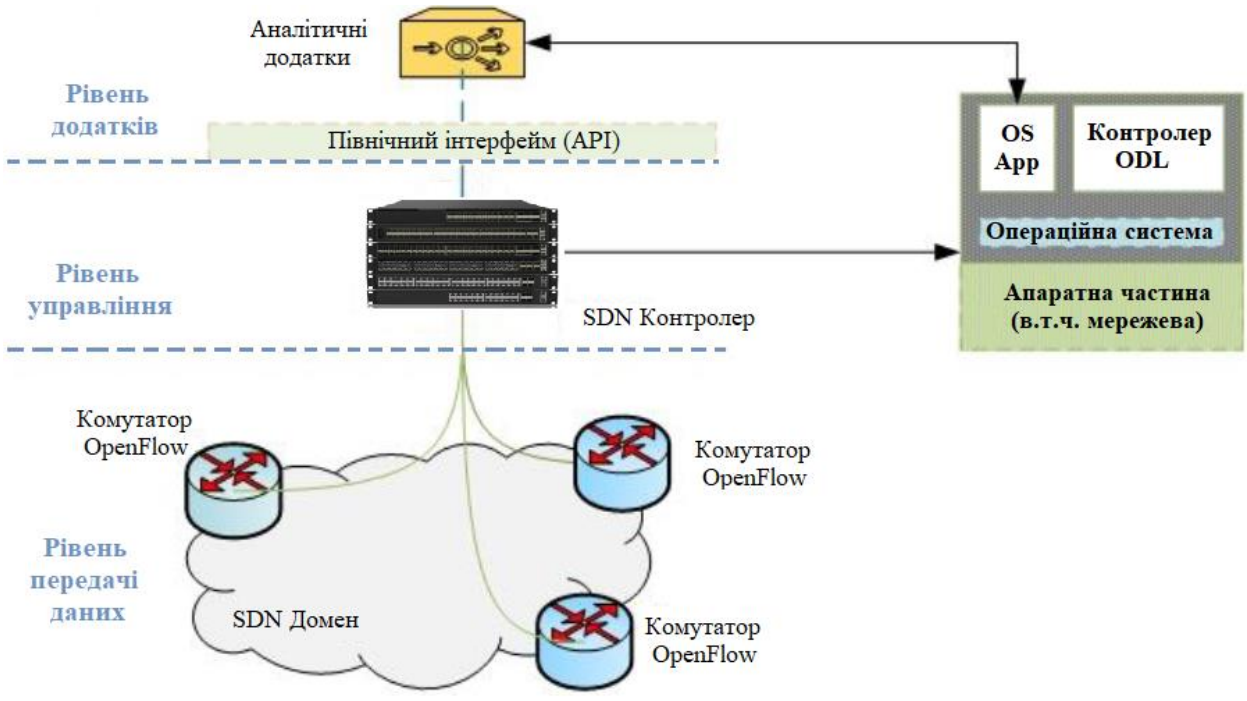


Рисунок 3.2 - Загальна схема досліджуваного стенду

Для дослідження зв'язку між змінами в активності службових потоків OpenFlow і навантаженням апаратних компонентів SDN-контролерів було

проаналізовано дві програми: «OS App» і «Аналітичні додатки». «OS App» — це серверна програма для операційної системи Linux, на основі якої функціонують SDN-контролери OpenDaylight Beryllium SR4. Вона запитує значення параметрів апаратного забезпечення в Linux Ubuntu за допомогою системних утиліт. Дані доступні через REST API цієї програми для інших застосунків через службову мережу. «Аналітичний додаток» також виконує функцію сервера, проте працює на північному інтерфейсі контролера REST API. Ця програма запитує значення параметрів у «OS App» через його API в службових мережах на управлінському рівні та в кінцевому підсумку формує набір даних для оцінки їх залежності.

Після вивчення доступних математичних методів, що оцінюють залежності між різними параметрами, було обрано метод багатопараметричного кореляційного аналізу.

У відповідності до теорії багатопараметричного кореляційного аналізу визначається ряд даних для порівняння і формується багатовимірна матриця даних (X), яка в подальшому наводиться до типової U-матриці. Варто зазначити, що рядок даної матриці (не наведеної), відповідні результату реєстрації параметру об'єктів (потік OpenFlow і апаратна частина SDN-контролера) ву одному відповідному експерименті, а стовбець містить результат спостерігання за певним параметром в усіх експериментальних дослідженнях. Для визначення рядів виконаємо позичання кількості параметрів з m де (m>1), ну а чисельність спостереженя відповідно n. У матрицях такий елемент як xij має значення j-ого параметра у і-му експеременті. При цьому допускається наявність порожніх значень певного елементу, як приклад, які можуть виникнути в резуьтаті пропуску виконання реєстрації значення параметру. В контексті даної дослідницької задачі реєстрація параметру відбувається кожну секунду. Однак у багатовимірному аналізі варто виконати усунення значень, які пропущено.

В такому випадку існує два підходи: викреслення певних рядів матриці чи внесення середнього значення на заміну відсутніх. У цій роботі вхідні ряди даних піддаються нормуванню перед безпосереднім кореляційним аналізом на основі механізму занесення середніх значень замість відсутніх.

Подальші методи обробки матриці X засновані на наступному припущенні: якщо об'єкт дослідження піддати новому обстеженню та отримати іншу матрицю даних, то після її обробки за допомогою тих самих методів будуть отримані результати, близькі до результатів першої матриці. Дане припущення ґрунтується на статистичній гіпотезі формування матриці.

Таким чином, об'єктом дослідження в рамках багатовимірного аналізу є багатовимірна випадкова величина, представлена вибіркою з кінцевим обсягом. Варто також підкреслити, що параметри, які характеризують об'єкт аналізу з різними фізичними значеннями, змінюють матрицю даних в залежності від шкали, за якою проводяться вимірювання обраних параметрів. Відповідно, матриця даних приводиться до стандартного формату, тобто значення параметрів стандартизуються (варіант). Як було зазначено раніше, стандартизовані матриці позначатимуться буквою U. Матриці X, відповідно до рядів параметрів, які будуть побудовані на основі матриці $DataSet_{ML}$, виглядають наступним чином:

$$X = \begin{matrix} ByteCount & PacketCount & CPU & RAM \\ x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{i1} & x_{i2} & x_{i3} & x_{i4} \end{matrix} \quad (3.6)$$

Перетворення матриці X, яка була сформована на основі обраних параметрів (3.6) потоку та додаткових даних про навантаження контролера в процентному співвідношенні, у стандартизовану матрицю U виконується наступним чином: для кожного досліджуваного параметра j = 1, 2, ... , 4 обчислюються зважені оцінки за такою формулою:

$$u_{ij} = \frac{(x_{ij} - \mu_1(x_j))}{\delta(x_j)}, \text{при i} = 1,2,\dots,n; j = 1,2,\dots,4 \quad (4.2.2.2)$$

У формулі (3.7) застосовується така пара параметрів, а саме математична очікуваність $\mu_1(x_j)$ і дисперсія $\delta^2(xj)$, що можливо розрахувати за наступними формулами:

$$\mu_1(x_j) = \frac{1}{n}\sum_{i=1}^{n} x_{ij} \tag{3.7}$$

$$\mu_2(x_j) = \delta^2(x_j) = \frac{1}{n-1}\sum_{i-1}^{n}\left(x_{ij} - \mu_1(x_j)\right)^2 \tag{3.8}$$

Отже, після розрахунку елементів "u_{ij}" формується матриця U, яка стає наступним об'єктом для аналізу. Слід підкреслити, що вплив загальних факторів і наявність об'єктивних закономірностей у поведінці досліджуваних об'єктів призводять лише до виникнення так званої статичної залежності. Таку залежність можна визначити як ситуацію, коли зміни одних величин викликають зміни в розподілі інших. У цьому випадку, як відомо, певні величини набирають конкретних значень з певними ймовірностями.

Є також окремий випадок статичної залежності, яка більш підходить до моделі, що розглядається в даній дисертаційній роботі, а саме кореляційної залежності, яка в свою чергу може арактеризуваись взаємозв'язком значень певною випадковою величиною із середніми значеннями іншими. Варто зазначити, кореляційні залежності зазвичай описують причинни залежності між значеннями використовуваних параметрів досліджуваної аналітичної моделі.

Таким чином, кореляційні залежності визначені різного роду параметрами, найбільше поширення серед яких набув показник, який виконує характеризування взаємозв'язків кількох непередбачуваних величин (так званий – парний показник):

- кореляційні моменти;

- кореляційні коофіцієнти.

Оцінки кореляційних моментів пар варіантів xj і xi виконують обчислення по вихідній матриці-X:

$$\xi_{jk} = \frac{1}{n}\sum_{i-1}^{n}\left(x_{ij} - \mu_1(x_j)\right)\left(x_{ik} - \mu_1(x_k)\right) \tag{3.9}$$

Коефіцієнти кореляції ρ_{jk} нормовано випадкової величини можуть називати коефіцієнтами кореляції, та його оцінки можливо розрахувати за наступною формулою:

$$\rho_{jk} = \frac{1}{n}\sum_{i-1}^{n}\left(u_{ij}x_{ik}\right) \quad (3.10)$$

У цьому, коефіцієнт кореляції залежить немає від значень випадкових величин, як від їх варіацій, тож значення величин збільшується на порядок, в такому разі коефіцієнти не зміниться. При цьому значення коефіцієнтів кореляції знаходяться у відліках від - 1 і до +1. У тому випадку коли випадкова величина U_j та U_k є незалежними, але коефіцієнт j_k має дорівнювати нулю, проте зворотні твердження є неправильним. Коефіцієнти кореляції j_k характеризуються значущістю лінійних зв'язків між випадковими величинами (параметрами):

- у разі якщо ρ_{jk} = 1 значеннь u_{ij} і u_{ik} збігаютьсяу повній мірі,а саме, значення параметру набуває однакового значення. Виловлюючись по інакшому, присутні функціональні залежності, якщо відоме значення певних параметрів, можливо з точністю вказувати значення інших параметрів;

- у разі якщо ρ_{jk} = –1 величина u_{ij} і u_{ik} мають протилежне значення.В такому разі також є функціональні залежності;

- у випадку ρ_{jk}=0 величина u_{ij} а u_{ik} майже не пов'язана одна з одним за лінійними співвідношеннями. Однак у такому разі не вказує на те що відсутні якісь інші (як приклад нелінійні) зв'язки між характеристиками;

- у випадку $|\rho_{jk}|>0$ і $|\rho_{jk}|<1$ чітких лінійних зв'язків uij та uik не існує. У відповідності якщо менші абсолютні величини кооффіцієнтів кореляції, в такому разі меншими мірами за значення одного з параметрів можливо передбачувати значення інших.

Раніше була згадана так звана функціональна залежність, яка, по суті, показує зв'язоки аналізованих величин які мають одну чи множину величин, у разі якщо дані величини залежать лише від цих факторів. Проте в даному випадку варто

враховувати те, що функціональні зв'язки є більш математичними абстракціями, тому що в реальних ситуаціях існує безліч властивостей самого об'єкта і зовнішнього середовища, що впливають один на одного. Тому, говорячи про можливу функціональну залежність, варто враховувати те, що обраний ряд параметрів, на основі яких проводиться безпосередній кореляційний аналіз, є обмеженим. При цьому ряд параметрів визначається в кожній конкретній практичній задачі, наприклад, як у поточній дослідницькій задачі, обґрунтований у цій статті ряд параметрів, вибраних для проведення аналізу, обмежений лише чотирма, що описують метадані суми службових потоків та навантаження контролера SDN.

Таким чином, у цій задачі (вивчення залежності між активністю службових потоків та навантаження контролера SDN), аналітичним додатком проводиться розрахунок наступних показників.

Приймемо такі умовні позначення:

- Byte_Count -BtCt;
- Packet_Count -PcCt;
- CPU value -Cpu;
- RAM value – Ram.

$$\xi_{(1\times4)x} = \begin{vmatrix} \mathrm{BtCt/PcCt} & \mathrm{BtCp/Cpu} & \mathrm{BtCp/Ram} & PcCt/Ram \\ \xi_{1\times1} & \xi_{1\times2} & \xi_{1\times3} & \xi_{1\times4} \end{vmatrix} \quad (3.11)$$

$$\rho_{(1\times4)U} = \begin{vmatrix} \mathrm{BtCt/PcCt} & \mathrm{BtCp/Cpu} & \mathrm{BtCp/Ram} & PcCt/Ram \\ \rho_{1\times1} & \rho_{1\times2} & \rho_{1\times3} & \rho_{1\times4} \end{vmatrix} \quad (3.12)$$

У представлених матрицях (3.11, 3.12), перший рядок відображає співвідношення параметру, між якими виконано розрахунок відповідних показників.

3.4 Прогнозування та практичні тестування навантаження на складові мережі SDN із використанням ШНМ

Аналіз доцільності застосування інструментів штучного інтелекту в сучасних та перспективних комунікаційних мережах було раніше детально висвітлено. У цьому контексті штучні нейронні мережі (ШНМ) були інтегровані в методику ідентифікації трафіку в комунікаційних мережах на основі метаданих потоків на етапах передачі інформації. Зараз ШНМ все частіше використовуються в різних сферах людської діяльності для вирішення багатьох задач. Наприклад, такі задачі, як розпізнавання текстів і прогнозування складних моделей, дедалі частіше реалізуються з використанням ШНМ, що дозволяє досягати кращих результатів. У повсякденному житті також активно застосовуються голосові помічники, які створюють нові типи інтерфейсів між людьми та інформаційними системами. В даний час існує широкий спектр нейронних мереж. У цій задачі нейронна мережа використовується для прогнозування навантаження. Архітектура досліджуваної ШНМ представлена на рисунку 3.3.

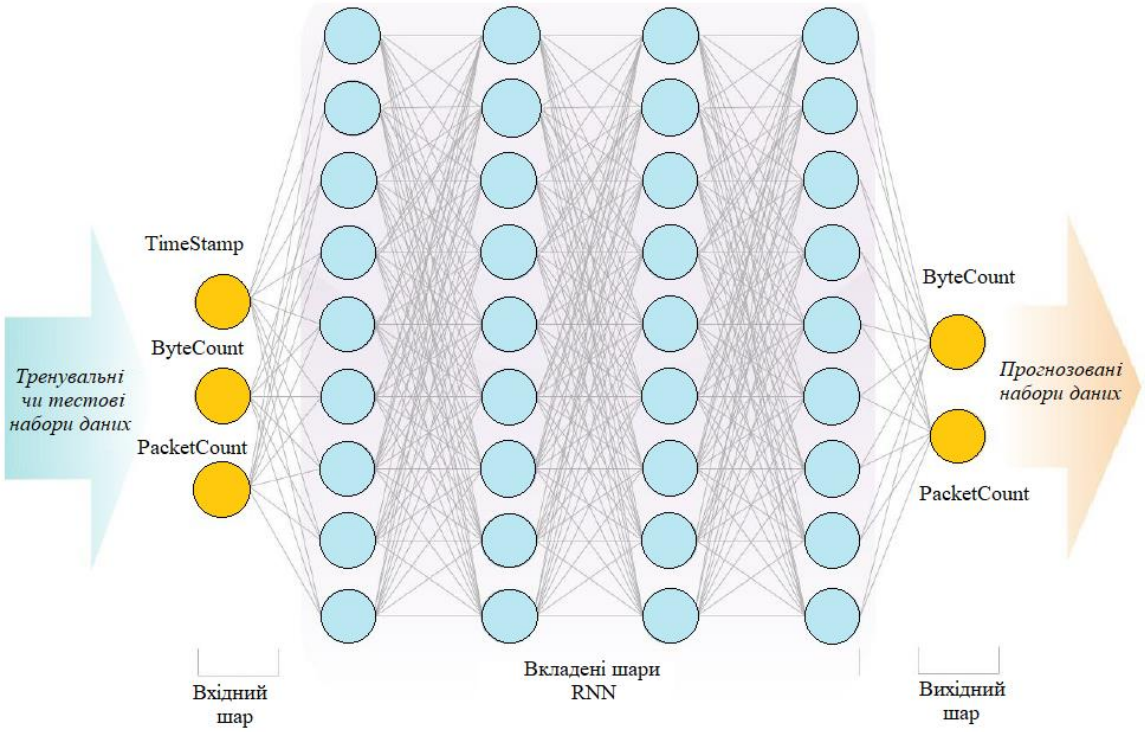


Рисунок 3.3 - Архітектура ШНМ

Функціонування ШНМ: на вхідний шар нейронів (відомий як плейсхолдери) надходять дані з підготовленого набору даних DataSetML. Плейсхолдери, які пов'язані з першим шаром нейронної мережі, з'єднуються зі структурними елементами. На виході розташовані два нейрони, які формують набір прогнозованих значень.

Нейронна мережа обробляє дані фіксованої довжини, причому початковий набір даних розбивається на сегменти по 200 рядків. Вихідні дані також діляться на дві частини: навчальну та тестову, в співвідношенні 80% до 20%. Архітектура цієї нейронної мережі є повністю рекурентною і складається з чотирьох рівнів нейронів, кожен з яких включає приблизно десять нейронів.

Параметри навчання ШНМ:

- оптимізатор: Adam;
- кількість епох: 20;
- розмір вибірки на ітерацію: 1024;
- швидкість навчання: 0.0025.

Отримані результати тестування. Практичні випробування запропонованого методу моніторингу та прогнозування навантаження на контролер програмно-конфігурованої мережі поділяються на два етапи. Перший етап передбачає проведення дослідження для підтвердження гіпотез про прямий вплив активностей службових потоків OpenFlow на контролери, які підключені до SDN-комутаторів, та їхнє апаратне забезпечення. Другий етап, що базується на позитивних результатах першого, полягає в реалізації та навчанні ШНМ для контролю та прогнозування активності службових потоків OpenFlow.

Для проведення практичних випробувань було використано стенд, структура та опис якого були наведені в попередньому розділі.

Для визначення показників залежності між вивченими параметрами було застосовано програмне забезпечення, розроблене на Python. Спочатку це ПЗ формує набір даних, а потім проводить обчислення відповідних показників X та U.

Як уже зазначалося при виборі цього математичного методу, кореляційний аспект має певні особливості: його значення залежить від одиниць виміру

незалежних величин, що ускладнює повну оцінку кореляційного зв'язку між двома величинами.

У цьому випадку початкові матриці нормалізуються відповідно до наведених раніше формул. На основі цієї нормалізованої матриці були створені точкові графіки, які демонструють розподіл зважених оцінок один відносно одного. Отримані графіки формувалися за вибірками, що містять 100 вимірювань з загального набору даних, і представлені на рисунку 3.4.

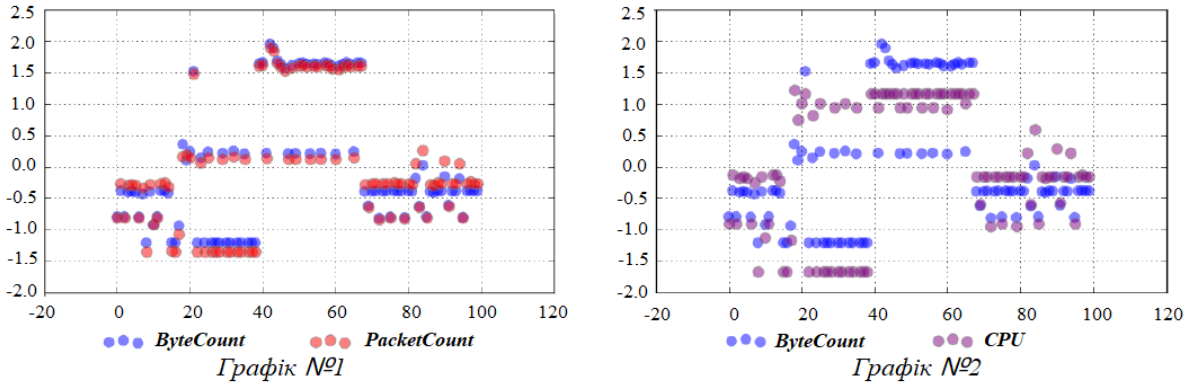


Рисунок 3.4 Точкові графіки розподілів залежних оцінок

На рисунку 3.4 наведені найцікавіші результати, що ілюструють зв'язок між ByteCount та навантаженням на центральний процесор. Графік 1 відображає природні залежності між цими двома параметрами, які мають безпосередню кореляцію, що слугує прикладом для оцінки. Графік 2 на цьому ж рисунку демонструє розподіл зважених оцінок різних параметрів, які потрібно проаналізувати з точки зору їх змін. Відповідно до графіка 2, можна стверджувати, що між варіаціями параметра ByteCount для загальних службових потоків та навантаженням на ЦП контролера програмно-конфігурованої мережі існує певний зв'язок.

Для оцінки рівня кореляції між досліджуваними величинами обчислюється коефіцієнт кореляції. У табл.3.1 наведені розраховані показники кореляції U між величинами, які відображені на графіках рисунка 3.4

Таблиця 3.1 – Матриця коефіцієнтів кореляції

ρ	Bt/Ct|PcCt	BtCp|Cpu
значення	0.98	0.89

Для аналізу отриманих даних, відображених у табл.3.1, звернемося до властивостей показника коефіцієнта кореляції, які були відображені раніше при описі методу. Всі отримані значення коефіцієнтів кореляції задовольняють наступній властивості $|\rho_{jk}| > 0\ i\ |\rho_{jk}| < 0$. На основі цієї властивості можна зробити наступний висновок: існує зв'язок між досліджуваними параметрами, і при цьому чим більша величина коефіцієнта кореляції, тим більше за значеннями одного параметра можна побудувати прогноз значення іншого. З отриманих значень у роботі цікавить значення коефіцієнта кореляції між $BtCt|Cpu$ рівний 0,89 відповідно. Значення цього показника дозволяє зробити висновок про наявність залежності між активністю службових потоків OpenFlow та навантаженням апаратного забезпечення SDN-контролера. Однак слід зазначити, що явної лінійної залежності між цими параметрами не виявлено. Отримані результати вказують на те, що запропонований метод моніторингу та прогнозування навантаження на контролер може бути ефективним. Таким чином, для оцінки роботи та навантаження контролера програмно-конфігурованої мережі може бути достатньо створення прогнозної аналітичної моделі для суми службових потоків OpenFlow, де сумарні лічильники групи службових потоків (метадані) виступають у ролі прогнозованого параметра.

Для проведення другої частини практичного дослідження стенд був модернізований щодо першої частини практичних випробувань. Друга частина практичних випробувань полягала у навчанні досліджуваної ШНМ.

На основі отриманого набору даних DataSetML була створена діаграма розсіювання значень, що представлена на рис.3.5. На цій діаграмі можна візуально виділити кілька зон (кластерів) розподілу точок, в яких домінують відповідні середні значення довжини пакета в потоці (окреслені овалами).

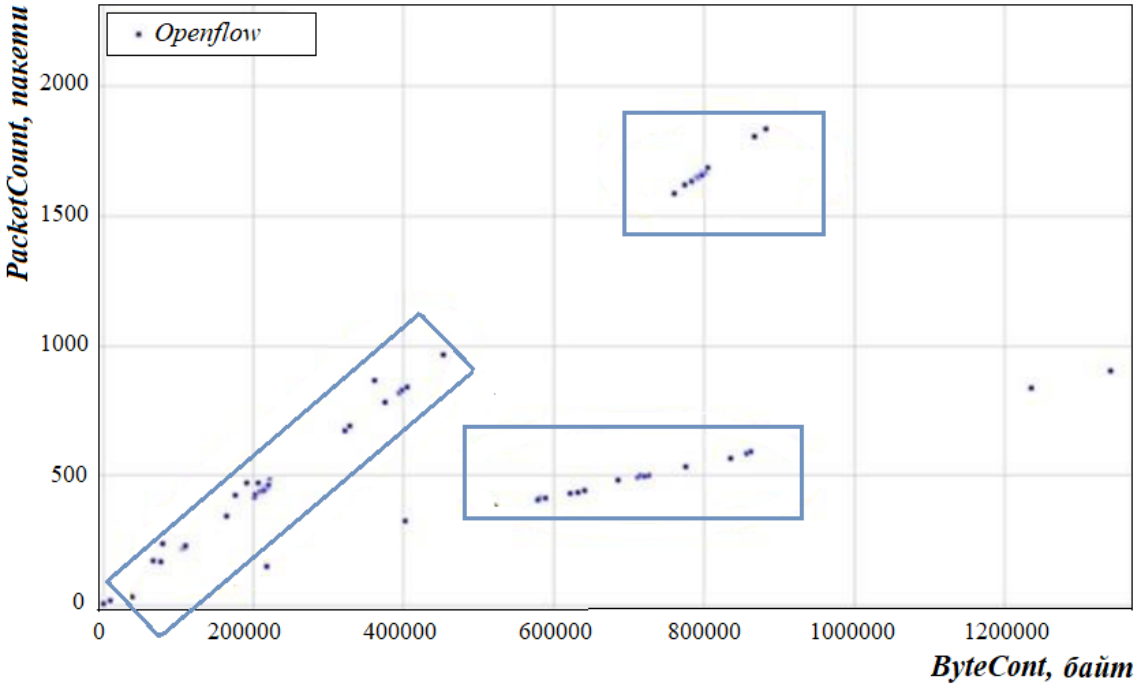


Рисунок 3.5 - Точкова діаграм розкиду значень набору даних DataSetML

Протягом періоду навчання ШНМ (оцінка точності прогнозів нейронних мереж), що є показником ефективності їх функціонування, спостерігалися параметри MSE - середня квадратична помилка. Зміни значення параметра MSE представлені на рис. 3.6.

При цьому MSE оцінюється за такою формулою:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \qquad (3.13)$$

де Y-вектор значень прогнозованої змінної, що спостерігаються, $\hat{Y}_i$ – вектор спрогнозованих значень. Іншими словами, MSE це середнє квадратів помилок прогнозу. Ця оцінка є простою і досить часто застосовуваною у подібному виді задач.

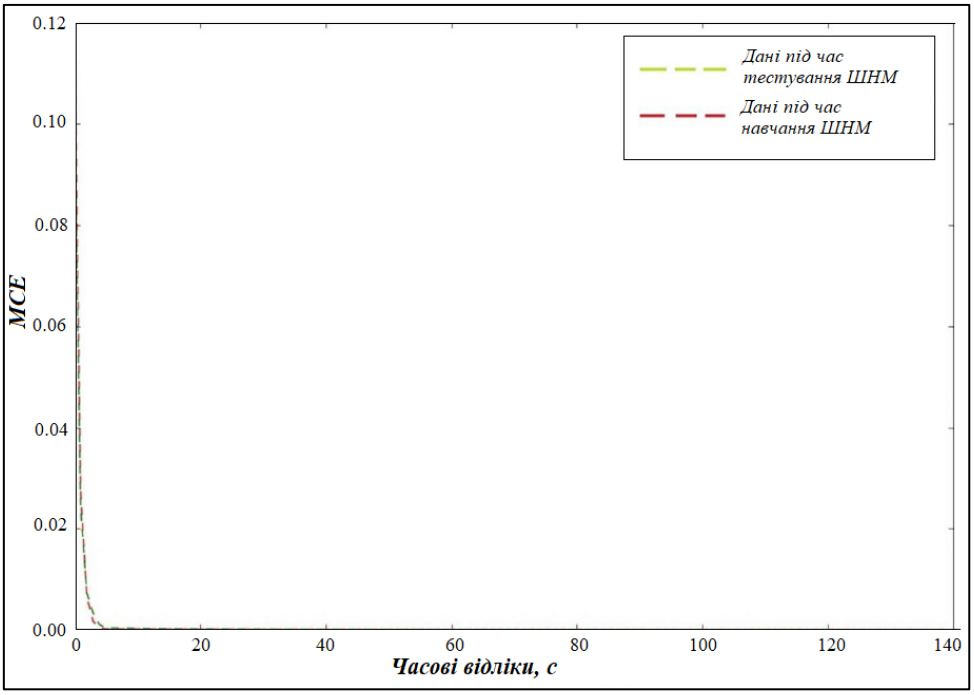


Рисунок 3.6 - Процеси навчаннь та тест ШНМ

З рис.3.6 ми можемо бачити пару графіків, що прямують один біля одного дуже близько. Червона пунктирна лінія відображає зміни параметрів MSE у момент роботи мереж на наборі даних який навчений, зелений колір відображає зміну параметрів MSE у робочих наборах інформації.

Графіки відображають, що архітектура яка була використана і параметр нейронних мереж у певній мірі відповідають задоволенню у формуванні прогнозу навантажень. Зрештою, вийшов наступний параметр:

$$MSE_{Train} = 4.54 * 10^{-6}, \quad (3.14)$$

$$MSE_{Train} = 1.5 * 10^{-5} \quad (3.15)$$

Під час навчання нейронної мережі було зафіксовано процес створення прогнозів. Для кращого розуміння цього процесу був побудований графік, який порівнює фактичні та прогнозовані дані нейронної мережі на інтервалі в 20 тисяч відліків. На рис. 3.7 можна побачити кілька знімків екрана, які демонструють прогрес навчання нейронної мережі.

У кожній з секцій (графіки №1, 2, 3, 4), представлених на рис. 3.7, розміщені два графіки — синій та зелений, що ілюструють реальні та прогнозовані значення.

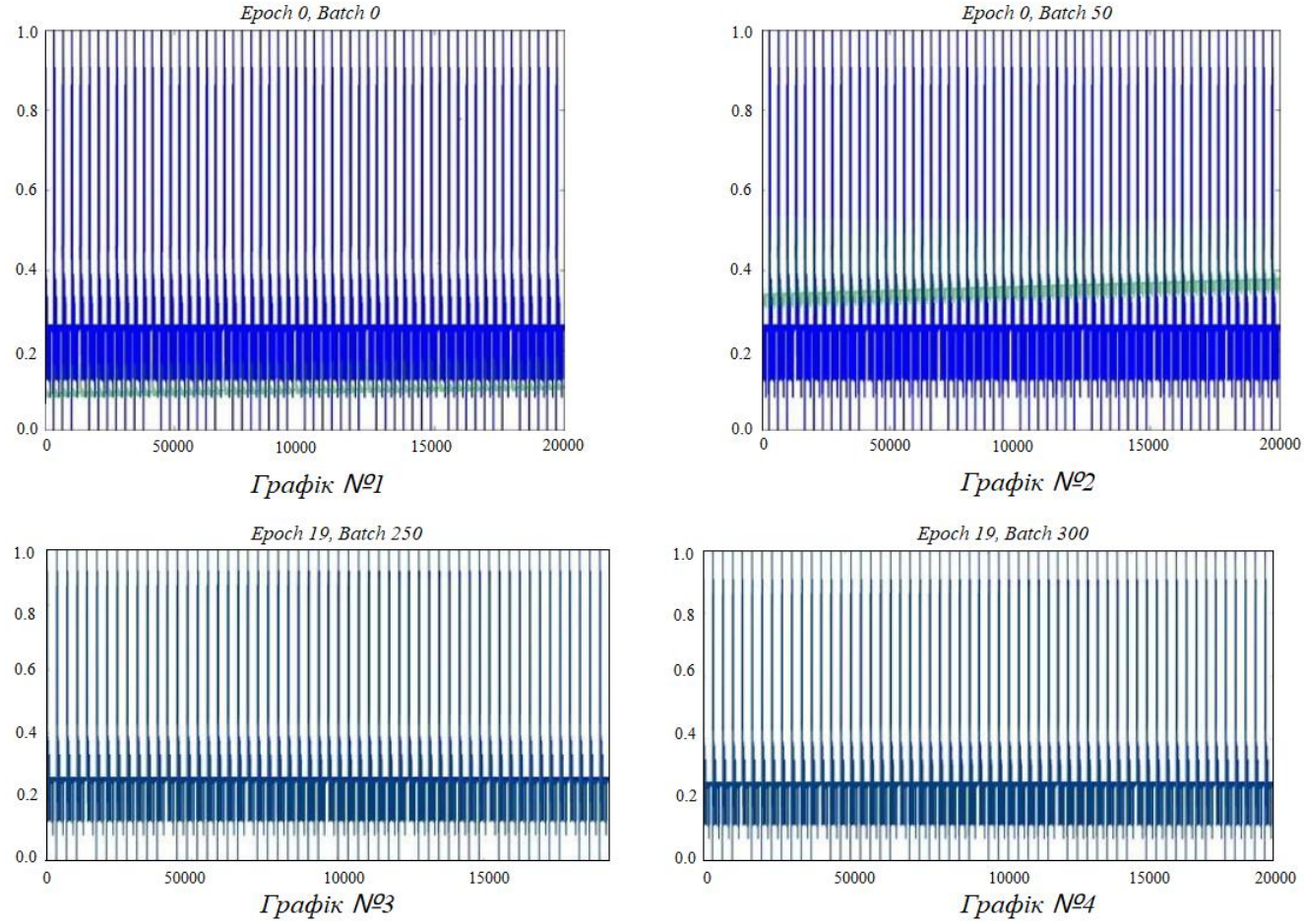


Рисунок 3.7 - Процеси навчань ШНМ

Графік №1 на першій лінії рис. 3.7, позначений як (Epoch 0, Batch 0), демонструє реальні значення та перший прогноз. На цьому графіку видно, що спрогнозовані значення істотно відрізняються від фактичних: зелений графік розташований значно нижче синього.

Праворуч, графік №2 на тій же лінії рисунка 3.7 з підписом (Epoch 0, Batch 50) відображає наступний прогноз нейронної мережі, що слідує за першим (графік №1). Тут помітно, що спрогнозовані значення (зелений графік) тепер перевищують реальні, але виглядають значно завищеними.

Нижні два графіки (графік №3 і графік №4), з підписами (Epoch 19, Batch 250) та (Epoch 19, Batch 300), показують завершальні етапи навчання мережі. На цих

графіках можна спостерігати, що нейронна мережа навчилася і тепер може робити точні прогнози активності службових потоків OpenFlow у справжній програмно-конфігурованій мережі на управлінському рівні. На останньому (правому нижньому) графіку №4 реальний графік візуально збігається з прогнозованим.

Після завершення навчання ШНМ програмне забезпечення фіксує її архітектурні стани та параметри в такому форматі, щоб інший модуль «Аналітичної програми» міг їх завантажити. Наступним кроком є завантаження стану навченої нейронної мережі, після чого програмний модуль генерує прогнозні дані на основі тестової інформації, що подається на вхід мережі.

ВИСНОВКИ

У магістерській роботі проведено комплексний аналіз сучасного стану мереж зв’язку та визначено перспективи їхнього розвитку до 2030 року. Розглянуто ключові характеристики мереж IMT-2020 та мереж наступних поколінь, які включають «ультрамалу затримку», «високу надійність зв’язку» та «надщільність мереж». Встановлено, що реалізація таких характеристик вимагає впровадження інноваційних методів і підходів, що забезпечують підвищення ефективності та адаптивності мережевої інфраструктури до сучасних потреб.

Для забезпечення ключових властивостей сучасних мереж зв’язку, зокрема «щільності» та «ультрамалих затримок», було досліджено сучасні технології та методи реалізації мережевої інфраструктури й систем надання послуг. До основних розглянутих технологій належать програмно-конфігуровані мережі (SDN), віртуалізація мережевих функцій (NFV), граничні обчислення з багатоканальним доступом, хмарні обчислення та мікросервісні підходи до проєктування програмного забезпечення. Також проаналізовано перспективи побудови мереж із використанням сучасних принципів управління та впровадження штучного інтелекту (ШІ). Особливу увагу приділено питанням ідентифікації трафіку, прогнозування його характеристик та розподілу навантаження на SDN-контролери.

У межах роботи ідентифікація трафіку базувалася на аналізі метаданих потоків, сформованих на основі полів MatchField протоколу OpenFlow. Метадані також використовувалися для прогнозування навантаження на SDN-контролери.

Проаналізувавши технічні аспекти та програмні можливості SDN-мереж, а також функціональність протоколу OpenFlow, було розроблено й реалізовано метод ідентифікації трафіку на основі метаданих потоків із використанням ШНМ.

Під час роботи з модельною мережею було проведено дослідження навчання ШНМ і серію тестувань, результати яких представлені у другому розділі. Отримані результати підтверджують ефективність досліджуваного методу, демонструючи точність ідентифікації трафіку без внесення додаткових затримок або змін у структуру потоків.

У межах виконання третього завдання проведено аналіз складності моніторингу контролера SDN як ключового елемента системи управління в мережах зв'язку IMT-2020 та наступних поколінь. Дослідження підтвердило необхідність реалізації механізмів прогнозування та моніторингу навантажень на контролер SDN для забезпечення стабільності роботи системи в умовах суворих вимог до якості обслуговування (QoS). Контролер SDN визнано найбільш уразливим компонентом мереж такого типу. Аналіз включав розгляд архітектурних підходів до моніторингу серверних систем, зокрема контролерів SDN. Встановлено недоліки стандартних методів моніторингу серверів, особливо при оцінці навантаження в контексті функціонування конфігурованих мереж. Для вирішення цих проблем використано метод контролю та прогнозування навантаження на контролер SDN на основі аналізу метаданих службового протоколу OpenFlow на рівнях управління.

Для реалізації прогнозування навантаження було використано рекурентну нейронну мережу (РНМ) з чотирма вкладеними рівнями. Такий підхід продемонстрував високу точність прогнозів за умов правильного налаштування параметрів навчання та архітектури РНМ. Практичну реалізацію методу виконано за допомогою програмного забезпечення на базі Python із використанням інтерфейсу REST API для інтеграції з контролерами мережі. Отримані результати підтвердили працездатність методу, а тестування моделі продемонструвало середньоквадратичну похибку (MSE) на рівні $MSETEST$=$1.5*10^{-5}$, що є високим показником якості.

Дослідження охоплює актуальні питання створення мереж із ультрамалими затримками, включно з аналізом мультиконтролерного взаємозв'язку та розподілу навантаження між декількома контролерами SDN.

Таким чином, результати дослідження свідчать про потенціал розробки в подальшому єдиного інтелектуального ядра мережі, здатного до самостійної адаптації та оптимізації під специфіку трафіку. Такий підхід, орієнтований на інтеграцію інтелектуальних мереж, може забезпечити високу якість

обслуговування для розумних сервісів, відповідно до жорстких критеріїв надійності та працездатності інфраструктури.

ПЕРЕЛІК ПОСИЛАНЬ

1. Shimodaira H, Tran GK, Sakaguchi K, Araki K. Investigation on Millimeter-wave Spectrum for 5G. 2015 IEEE Conference on Standards for Communications and Networking (CSCN); 2015. p. 143–148.

2. Borodin, A.S. Fifth-generation communication networks as the basis of the digital economy / A.S. Borodin, A.E. Kucheryavy // Electrosvyaz. - 2017. - No. 5. - P. 45-49.

3. K. Antonakoglou, X. Xu, E. Steinbach, T. Mahmoodi, and M. Dohler,"Towards haptic communications over the 5G tactile internet," IEEECommunications Surveys Tutorials, pp. 1–1, 2023.

4. Volkov A. N., Muthanna A. S. A., Kucheryavy A. E. Fifth-generation communication networks: towards 2030 networks// Information technologies and telecommunications. 2020. Vol. 8. No. 2. P. 32–43.

5. Recommendation M.2083-0 IMT-Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond. ITU-R, Geneva. – 2022.

6. R. Yaroshevych, V. Tkachov, A. Kovalenko, and D. Rosinskyi "Modelling the Domain Architecture of the Tactile Inter-net Using a Foggy Infrastructure," IEEE 9th International Conference on Problems of Infocommunications, Science and Tech-nology (PIC S&T). October 2022, pp. 512-516.

7. H.-C. Yu, Z.-L. Lee and H.-Y. Lee, "Revising Taiwan's frequency usage fee regulation," Telecommun. Pol., vol. 28, no. 9–10, pp. 679–695, 2019.

8. Technical Specification. FG-NET2030 – Focus Group on Technologies for Network 2030. Network 2030 Architecture Framework. ITU-T, Geneva. – June 2020.

9. A. Muthanna et al. "Secure and Reliable IoT Networks Using Fog Computing with Software-Defined Networking and Blockchain," Journal of Sensor and Actuator Networks, vol. 8, no. 1, pp. 15, February 2019.

10. Recommendation Y.3172. Architectural framework for machine learning in future networks including IMT-2020. ITU-T, Geneva. – June 2019.

11. Recommendation Y.3170. Requirements for machine learning-based quality of service assurance for the IMT-2020 network. ITU-T, Geneva. – September 2018.

12. Recommendation Y.3104. Architecture of the IMT-2020 network. ITU-T, Geneva. – December 2018.

13. Recommendation Y.3174. Framework for data handling to enable machine learning in future networks including IMT-2020. ITU-T, Geneva. – February 2020.

14. Technical Specification ETSI TS 123 01 v16.6.0 Release 16. 5G. System architecture for the 5G System (5GS). ETSI, France. – October 2020.

15. Recommendation Y.3102. Framework of the IMT-2020 network. ITU-T, Geneva. – May 2018.

16. Koucheryvy, A., Muthanna, A., Volkov, A.: AI/machine learning for ultra-reliable low-latency communication, ITU News Magazine, No.5, December 2020. pp.62-65.

17. Fan, Z., Xiao, Y., Nayak, A., Tan, C.: 'An improved network security situation assessment approach in software defined networks', Peer-to-Peer Networking and Applications, 2017.

18. Volkov A., Khakimov A., Muthanna A., Kirichek R., Vladyko A., and Koucheryavy A. Interaction of the IoT traffic generated by a Smart city segment with SDN core network // WWIC 2017 International Conference on Wired/Wireless Internet Communication, 2021, pp. 115–126.

19. Tran T.X. Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges. Hajisami, A., Pompili, D. IEEE Communications Magazine 55, 2017, 54–61.

20. A. Ateya, A. Muthanna, I. Gudkova, A. Vybornova and A. Koucheryavy, "Intelligent core network for Tactile Internet system," International Conference on Future Networks and Distributed Systems, ACM, P.15, Cambridge, Jul.2020.

21. Banikazemi, M.; Olshefski, D.; Shaikh, A.; Tracey, J.; Wang, G. Meridian: An SDN Platform for Cloud Network Services. IEEE Commun. Mag. 2018, 51, 120–127.

22. Heller, B.; Scott, C.; McKeown, N.; Shenker, S.; Wundsam, A.; Zeng, H.; Whitlock, S.; Jeyakumar, V.; Handigol, N.; McCauley, J.; et al. Leveraging SDN

Layering to Systematically Troubleshoot Networks. In Proceedings of the ACMWorkshop on Hot Topics in Software Defined Networks (HotSDN), Hong Kong, China, 12–16 August 2017, pp. 37–42.

23. X. Sun, and N. Ansari, “Green Cloudlet Network: A Distributed Green Mobile Cloud Network,” IEEE Network, vol. 31, no. 1, pp. 64-70, January 2019.

24. A. Muthanna et al. “Secure and Reliable IoT Networks Using Fog Computing with Software-Defined Networking and Blockchain,” Journal of Sensor and Actuator Networks, vol. 8, no. 1, pp. 15, February 2019.

25. Tran T.X. Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges. Hajisami, A., Pompili, D. IEEE Communications Magazine 55, 2017, 54–61.

26. Global Mobile Suppliers Association, “Asia Pacific – thought leaders in the 5G spectrum domain – key spectrum aspects,” Forum Global the 5th annual Asia Pacific Spectrum Management Conference, Kuala Lumpur, 2019.

27. M. Suryanegara, “5G as disruptive innovation: standard and regulatory challenges at a country level,” Int. J. Tech., vol. 4, pp. 635–642, 2016.

28. Khan, S.S., Ahmad, A.: ‘Cluster center initialization algorithm for K-means clustering’, Pattern recognition letters,2021, 25, (11), pp. 1293-1302.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

КВАЛІФІКАЦІЙНА МАГІСТЕРЬКА РОБОТА

на тему: **«МЕТОДИКА ПОБУДОВИ МЕРЕЖ ЗВ'ЯЗКУ ІЗ ЗАСТОСУВАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ»**

Виконав: здобувач вищої освіти гр.КСДМ-61
Бодорацький Д.А.
Керівник: к.т.н., доцент
Лащевська Н.О.

Київ 2024 р.

МЕТА ТА ЗАВДАННЯ

Мета роботи - реалізація методів побудови мережі та надання послуг із застосуванням штучного інтелекту.

Об'єкт дослідження – процес побудови мереж зв'язку та послуг із застосуванням штучного інтелекту.

Предмет дослідження – сучасні мережі зв'язку.

Завдання магістерської роботи:

1. огляд сучасних мереж зв'язку, стан та перспективи їх подальшого розвитку;
2. дослідження методики розпізнання трафіка послуг в програмно-конфігурованих мережах;
3. контроль та прогнозування навантаження складових мережі в умовах обмеження якості обслуговування послуг.

Концепція мереж зв'язку IMT-2020

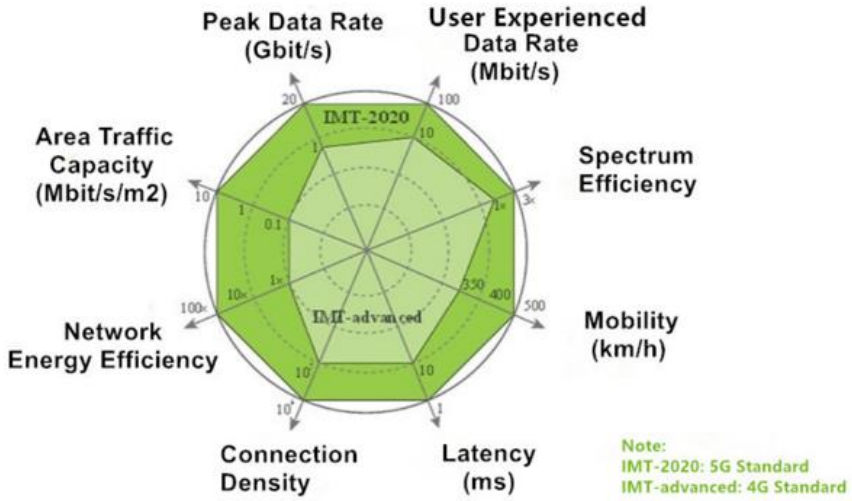


Рис.1.1. Ключові можливості від IMT-advanced до 5G

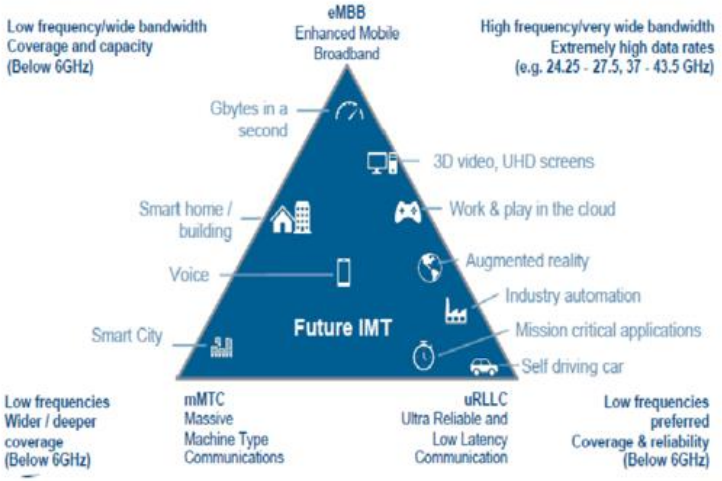


Рис.1.2. Сценарії застосування IMT-2020

Програмно-визначена мережа з використанням OpenFlow

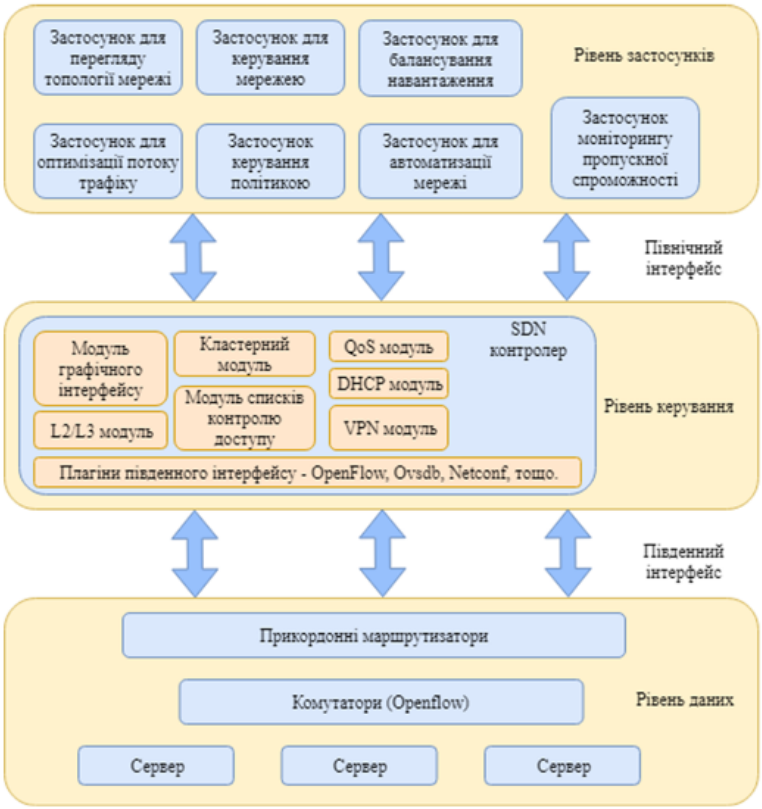


Рис.1.3. Трирівнева програмно-визначена мережева архітектура (SDN)

Рис.1.4. Базове пересилання пакетів за допомогою OpenFlow у комутаторі

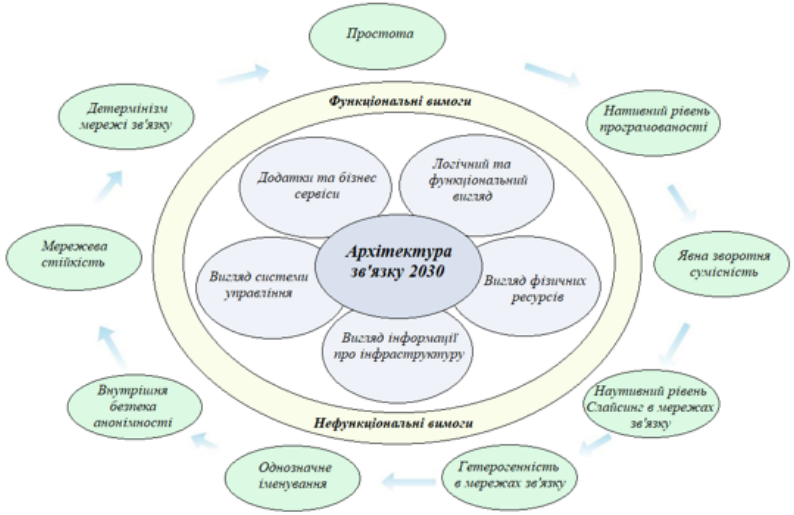


Рис.1.5. Принципи реалізації архітектури мереж зв'язку майбутнього

Особливості застосування штучного інтелекту у мережі

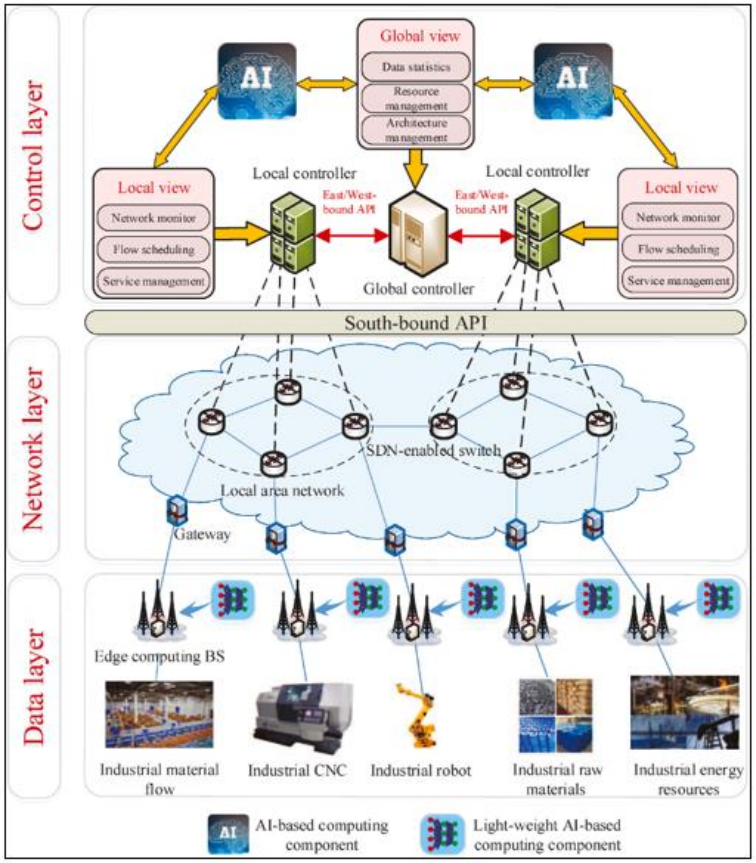


Рис.1.6. Модель мережі із застосуванням ШІ

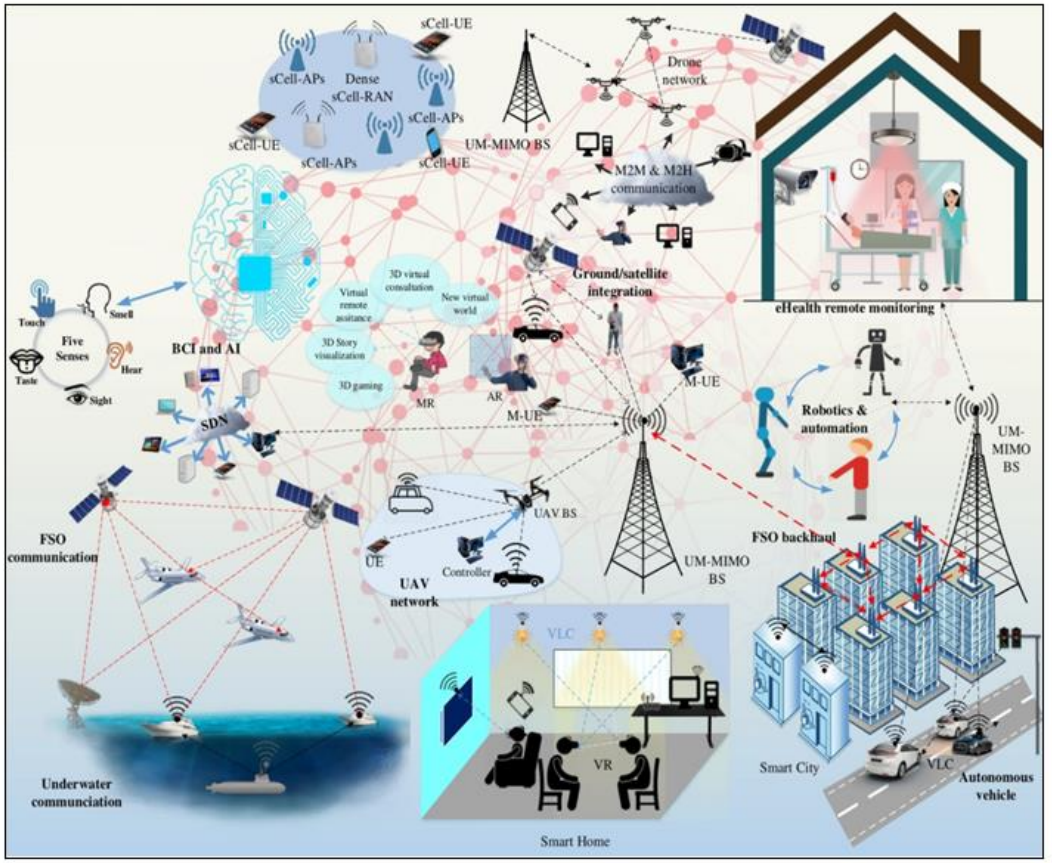


Рис.1.7. Приклад архітектури розподіленого ШІ в мережах та перспектива подальших досліджень

Огляд стандартизації ШІ в мережах зв'язку

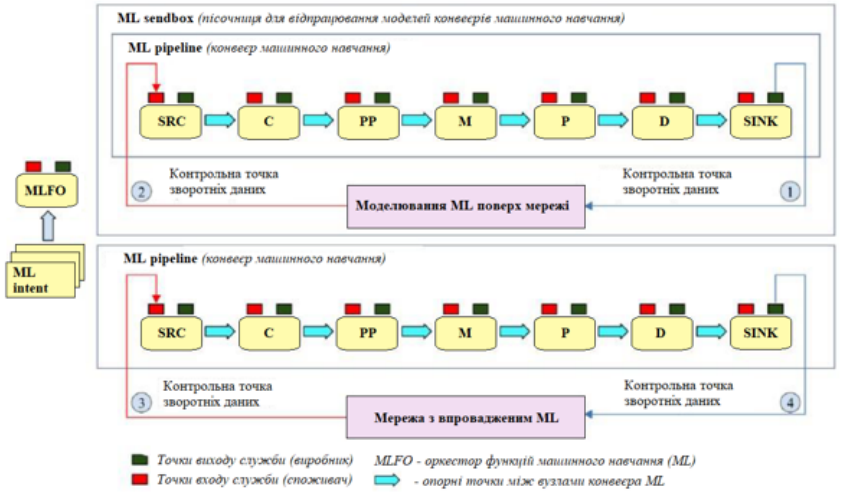


Рис.1.8. Конвеєр машинного навчання для мереж зв'язку

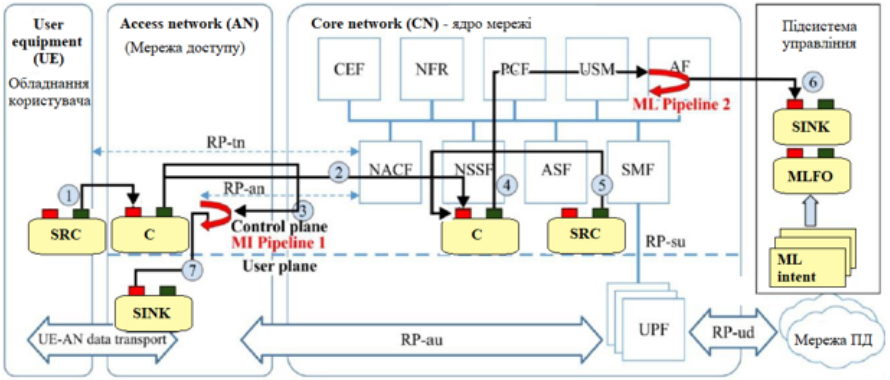


Рис.1.9. Високорівнева архітектура конвеєра машинного навчання та його реалізація в мережі МТ-2020

Огляд архітектури модельної мережі та трафіку досліджуваних сервісів

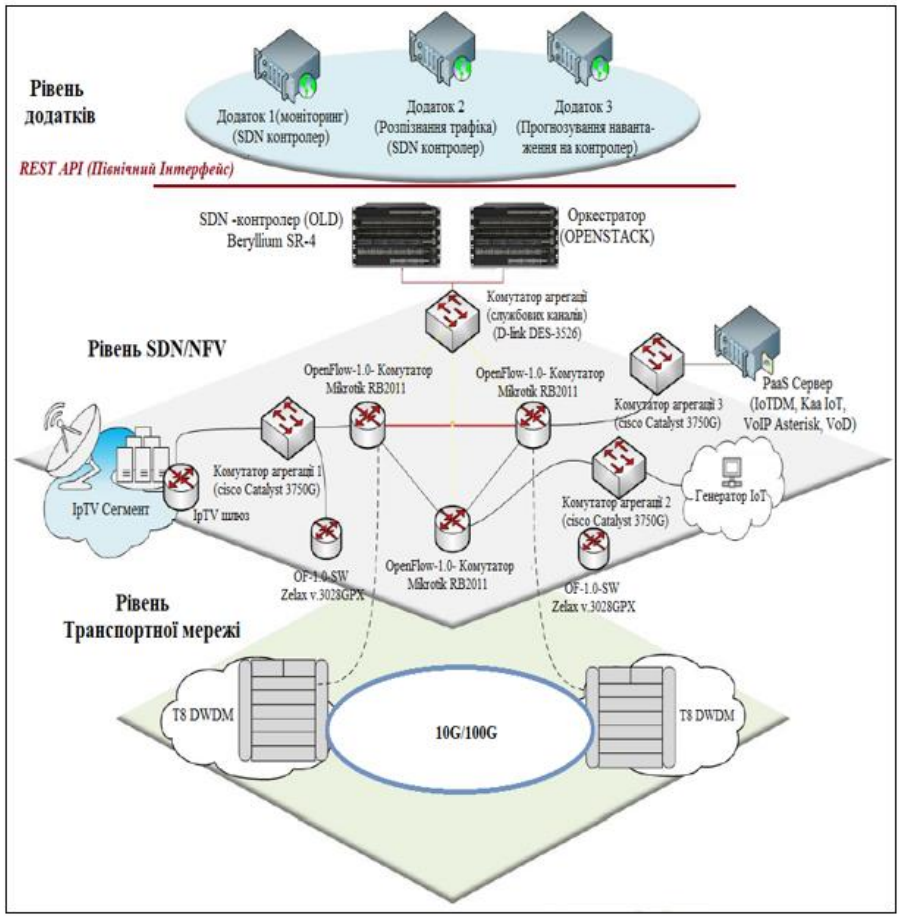


Рис.2.1. Приклад архітектурної моделі мережі

Табл.2.1. Характеристики основного обладнання

№	Назва	Технічний опис
1	SDN-контролер (ODL) Beryllium SR-4	ЦПУ: Intel Xeon® E3-1220V2 ОЗУ: 16 Гб ОС: Linux Ubuntu 14.04 LTS Версія ПЗ: Opendaylight Beryllium SR4
2	OpenFlow 1.0 комутатор Mikrotik RB 201 1UI AS-RM	Mikrotik RB 201 1UI AS-RM Версія ПО: OpenFlow 1.0
3	Комутатор агрегації №1	Cisco Catalyst 3750G series PoE-24
4	Комутатор агрегації №2	D-Link DES 3526
5	Комутатор агрегації №3	Cisco Catalyst 3750G series PoE-24
6	Комутатор агрегації (службових каналів)	D-Link DES 3526
7	PaaS Сервер (IoTDM – сервер)	ЦПУ: Intel Xeon(R) E3-1220V2 ОЗУ: 12 Гб ОС: Linux Ubuntu 14.04 LTS Версія: Opendaylight Boron SR2
8	Д1 (моніторингу)* Д2 (Розпізнавання трафіку) Д3 (Прогнозування навантаження на контролер) *	ЦПУ: Intel Core i5 ОЗУ: 8 Гб ОС: Linux Ubuntu 16.04 LTS

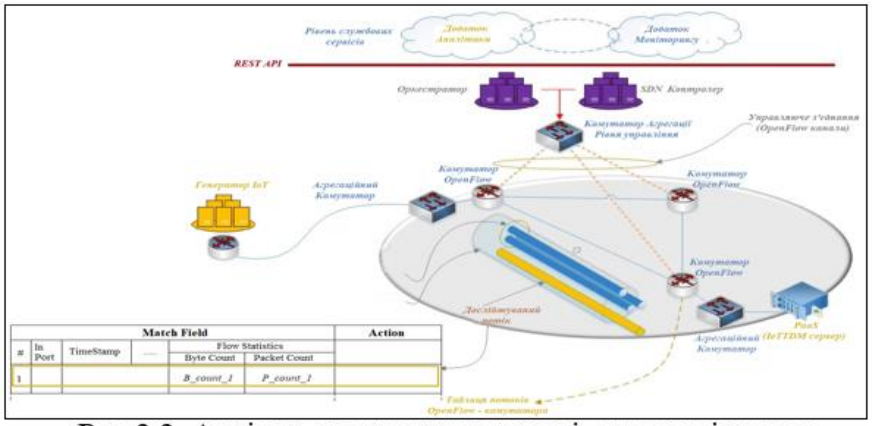


Рис.2.2. Архітектура частини мережі, яка аналізується по взятому сценарію

Результати досліджень

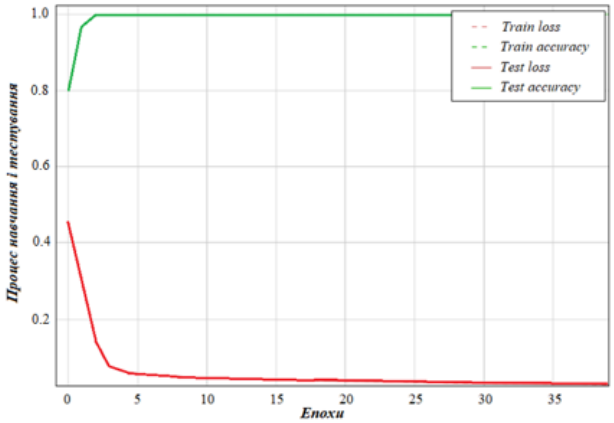


Рис.2.3. Криві періоду навчання і тесту ШНМ

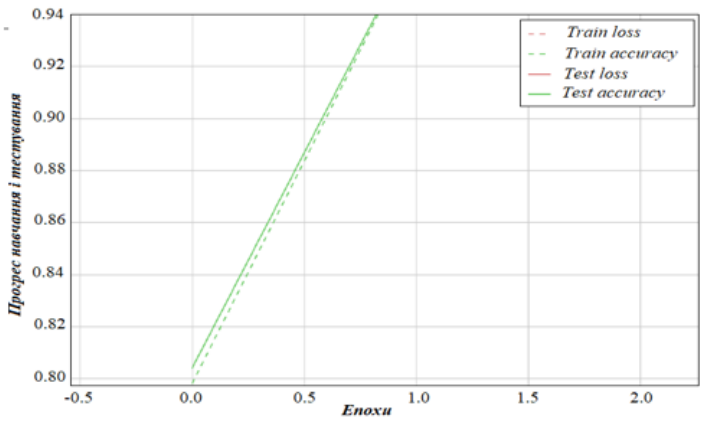


Рис.2.4. Графічне зображення кривих навчання та тесту ШНМ

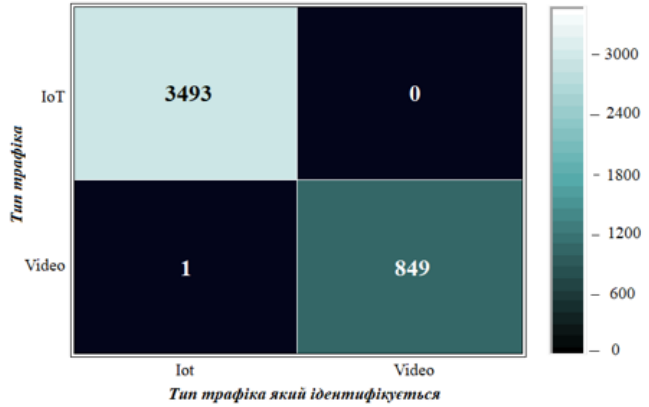


Рис.2.5. Відображення матриці протиріч

Прогнозування навантаження на контролер мереж SDN із застосуванням ШІ

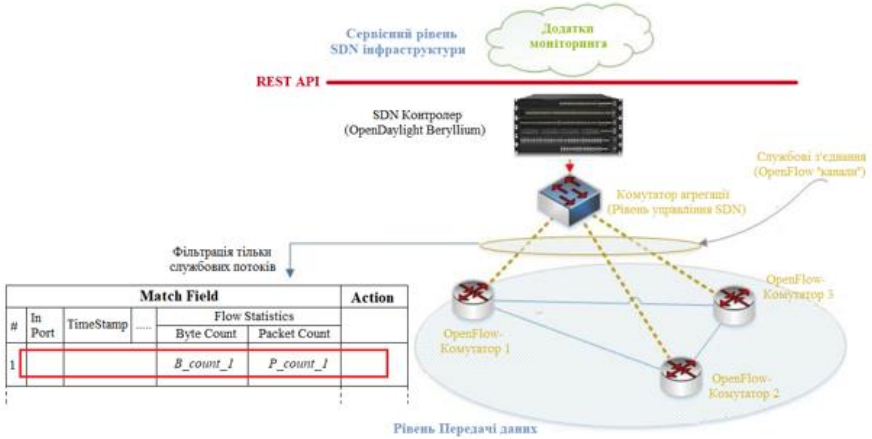


Рис.3.1. Архітектура сегменту з об'єктом дослідження

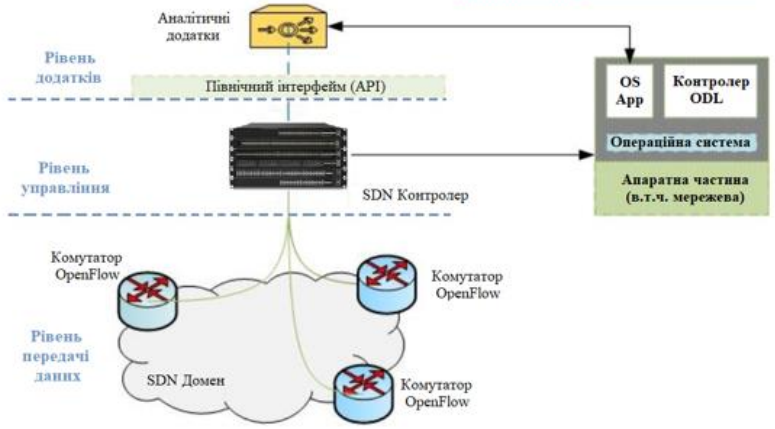


Рис.3.2. Загальна схема досліджуваного стенду

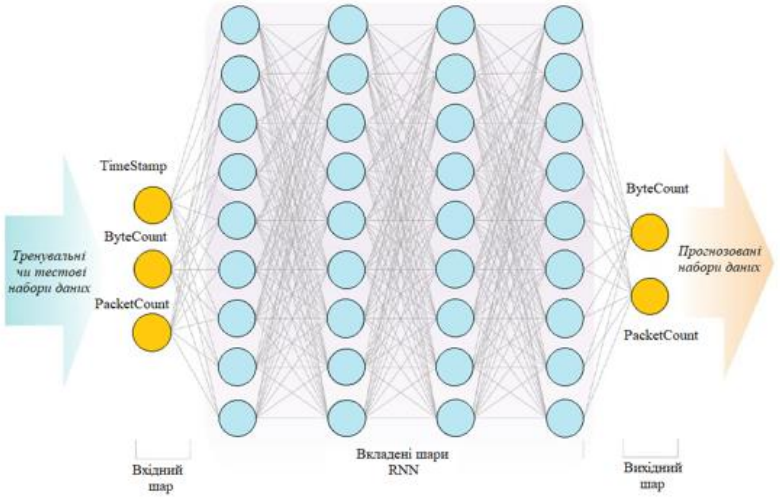


Рис.3.3. Архітектура ШНМ

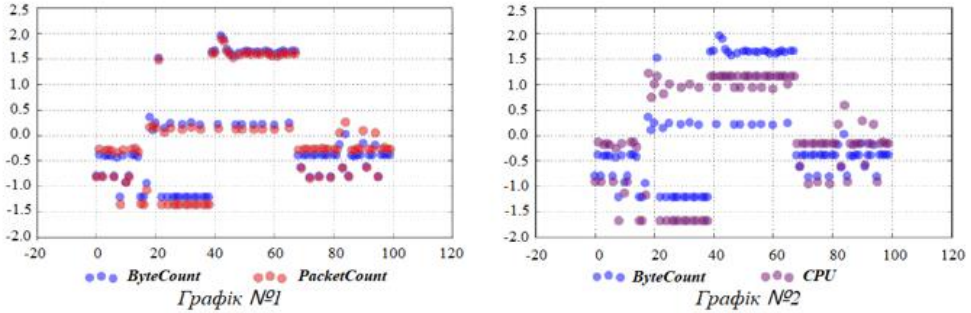


Рис.3.4. Точкові графіки розподілів залежних оцінок

Практичні дослідження

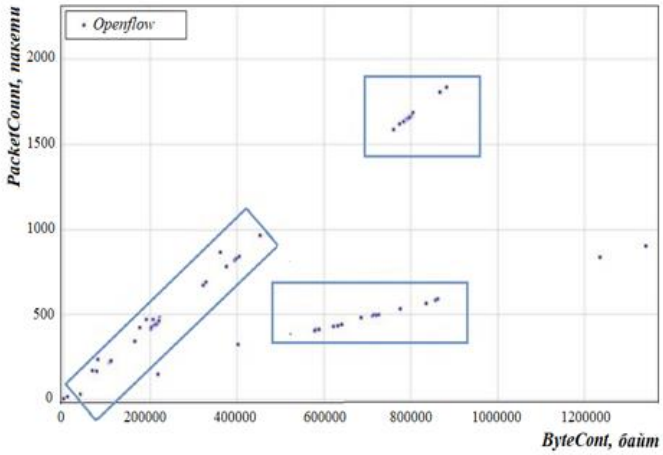


Рис.3.5.Точкова діаграм розкиду значень набору даних DataSetML

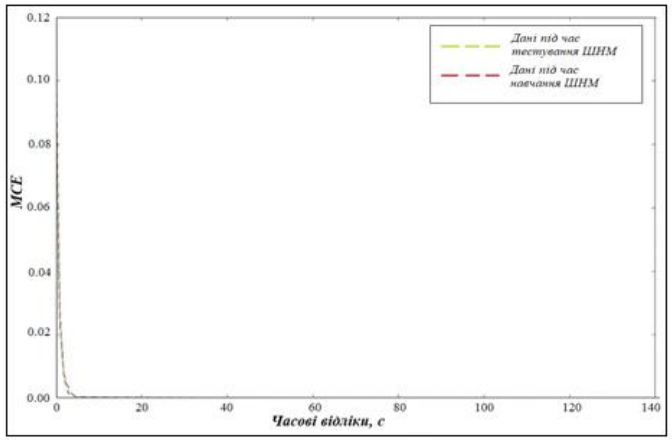
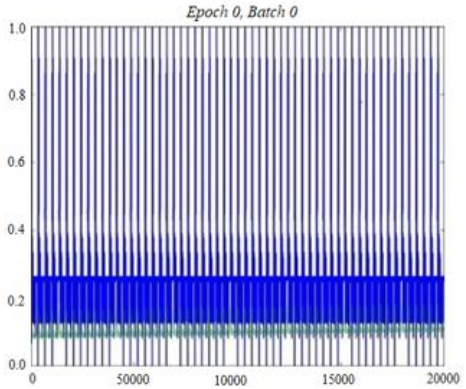
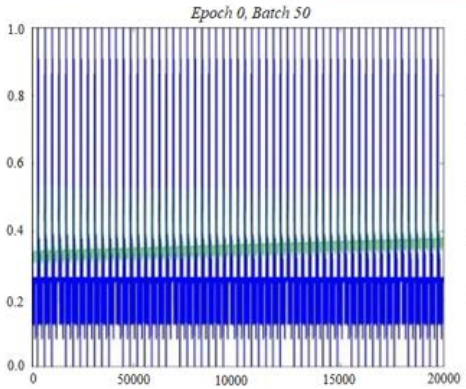


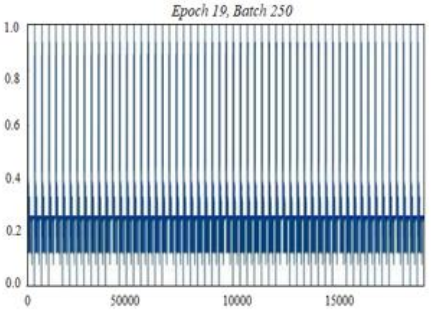
Рис.3.6. Процеси навчаннь та тест ШНМ



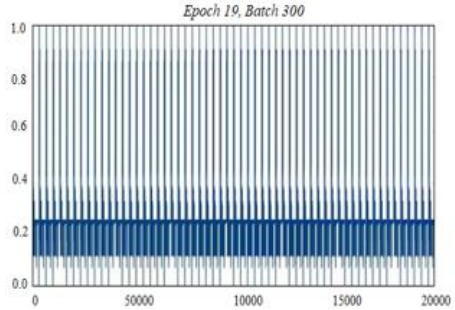
Графік №1



Графік №2



Графік №3



Графік №4

Рис.3.7. Процеси навчань ШНМ

Висновки

Отже, у магістерській роботі проведено аналіз сучасного стану мереж зв'язку та визначено перспективи їхнього розвитку.

Розглянуто ключові характеристики мереж IMT-2020 та мереж наступних поколінь, які включають «ультрамалу затримку», «високу надійність зв'язку» та «надщільність мереж». Встановлено, що реалізація таких характеристик вимагає впровадження інноваційних методів і підходів, що забезпечують підвищення ефективності та адаптивності мережевої інфраструктури до сучасних потреб.

Проаналізовано перспективи побудови мереж із використанням сучасних принципів управління та впровадження штучного інтелекту. Особливу увагу приділено питанням ідентифікації трафіку, прогнозування його характеристик та розподілу навантаження на SDN-контролери.

Під час роботи з модельною мережею було проведено дослідження навчання ШНМ і серію тестувань. Отримані результати підтверджують ефективність досліджуваного методу, демонструючи точність ідентифікації трафіку.

У межах виконання третього завдання проведено аналіз складності моніторингу контролера SDN як ключового елемента системи управління в мережах зв'язку IMT-2020 та наступних поколінь. Дослідження підтвердило необхідність реалізації механізмів прогнозування та моніторингу навантажень на контролер SDN для забезпечення стабільності роботи системи в умовах суворих вимог до якості обслуговування.

Отримані результати підтвердили працездатність методу, а тестування моделі продемонструвало середньоквадратичну похибку (MSE) на рівні $MSETEST = 1.5 * 10^{-5}$, що є високим показником якості.

Таким чином, результати дослідження свідчать про потенціал розробки в подальшому єдиного інтелектуального ядра мережі, здатного до самостійної адаптації та оптимізації під специфіку трафіку. Такий підхід, орієнтований на інтеграцію інтелектуальних мереж, може забезпечити високу якість обслуговування для розумних сервісів, відповідно до жорстких критеріїв надійності та працездатності інфраструктури.

Апробація результатів

Результати магістерської роботи доповідались на двох науково-технічних конференціях:

1. Бодорацький Д.А. Дослідження складнощів застосування штучного інтелекту. II всеукраїнську науково-технічну конференцію «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу» 18.11. 2024.
2. Бодорацький Д.А. Аналіз концепції сучасних мереж зв'язку. Науково-практична конференція «Проблеми комп'ютерної інженерії» 03.12.2024.

По тематиці роботи подано на публікацію одну статтю в науковий журнал «Зв'язок», на тему «Штучний інтелект у мережах зв'язку».

ДЯКУЮ ЗА УВАГУ, ДОПОВІДЬ ЗАВЕРШЕНО