

Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

ПОЯСНЮВАЛЬНА ЗАПISKA

до випускної кваліфікаційної роботи магістра
на тему: «Методи та технології побудови надійної
енергонезалежної мережної інфраструктури
організації»

Виконав студент 2-го курсу групи КІм-23-1
спеціальності 123 «Комп'ютерна інженерія»

 В. С. Окрушко

Керівник роботи доцент кафедри комп'ютерної інженерії
та кібербезпеки, кандидат педагогічних наук, доцент

 О. С. Головня

Рецензент доцент кафедри комп'ютерної інженерії та
кібербезпеки, кандидат технічних наук, доцент

 Ю. Б. Бродський

Житомир – 2024

Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки
Спеціальність 123 «Комп'ютерна інженерія»
Освітня програма «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач

кафедри комп'ютерної
інженерії та кібербезпеки

 Андрій Сфіменко
17 вересня 2024 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи магістра

Студент: Окрушко Віталій Сергійович

Тема роботи: «Методи та технології побудови надійної енергонезалежної
мережної інфраструктури організації»

затверджена Наказом університету від 17 вересня 2024 р. №478/с

Термін здачі студентом закінченої роботи 20 грудня 2024 р.

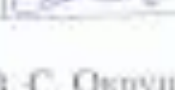
Вихідні дані роботи (зазначається предмет і об'єкт дослідження):

1. Інфраструктура енергонезалежної мережі для інтернет-провайлера.
2. Проект мережевої топології з використанням оптоволоконних технологій GPON та DWDM.
3. Використання алгоритмів машинного навчання для прогнозування стану мережі.

Обладнання та програмне забезпечення:

1. Windows Server 2019 Standard (v.1809);
2. Windows 10 (v.1909);
3. Платформа для машинного навчання ML.NET (v.7.5.0);
4. Програмне забезпечення для моделювання мереж Cisco Packet Tracer (v.8.2.1);
5. IDE Visual Studio 2022 (v.17.6.4);
6. SQL Server Management Studio (SSMS) (v.18.12.1).

Консультанти з випускної кваліфікаційної роботи із зазначеним розділом, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Задавши питання	Задавши відповідь
1	О. С. Голованюк		
2	О. С. Голованюк		
3	О. С. Голованюк		

Студент  В. С. Окрушко
Керівник  О. С. Голованюк

Календарний план

№ з/п	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка завдання	17.09.2024 – 23.09.2024	Виконано
2	Розробка плану роботи	24.09.2024 – 30.09.2024	Виконано
3	Аналіз інфраструктурних та технологічних рішень для енергонезалежної мережі	01.10.2024 – 07.10.2024	Виконано
4	Аналіз альтернативних джерел живлення та систем зберігання енергії	08.10.2024 – 14.10.2024	Виконано
5	Дослідження автоматизованих систем управління енерготорговлею	15.10.2024 – 21.10.2024	Виконано
6	Моделювання мережевої інфраструктури	22.10.2024 – 28.10.2024	Виконано
7	Налаштування конфігурацій VLAN, DHCP, NAT та маршрутизації	29.10.2024 – 04.11.2024	Виконано
8	Тестування працездатності мережевої інфраструктури	05.11.2024 – 11.11.2024	Виконано
9	Вибір алгоритмів машинного навчання для прогнозування стану мережі	12.11.2024 – 18.11.2024	Виконано
10	Розробка та тестування ПЗ для прогнозування стану обладнання мережі	19.11.2024 – 25.11.2024	Виконано
11	Розрахунок економічної ефективності та оцінка терміну окупності проекту	26.11.2024 – 02.12.2024	Виконано
12	Написання пояснювальної записки	03.12.2024 – 13.12.2024	Виконано

Студент  В. С. Оврутин
Керівник  О. С. Голосенко.

АНОТАЦІЯ

Випускна робота магістра присвячена дослідженню методів та технологій побудови надійної енергонезалежної мережної інфраструктури організації. Пояснювальна записка до випускної роботи містить 119 сторінок, 60 ілюстрацій, 15 таблиць, 1 додаток та 32 літературних джерел.

Метою роботи є створення та обґрунтування комплексної енергонезалежної мережної інфраструктури з оптоволоконним підключенням абонентів для забезпечення безперервності надання інтернет-послуг і зменшення енергозалежності організації за рахунок використання альтернативних джерел живлення та автоматизованих систем моніторингу.

У роботі визначені основні завдання проекту, зокрема розробка інфраструктурного рішення з використанням оптоволоконних технологій GPON і DWDM, впровадження програмного забезпечення для моніторингу з функціями аналізу та прогнозування несправностей обладнання мережі, а також оцінювання економічної доцільності проекту.

Основна увага приділяється аналізу сучасних рішень для забезпечення автономності, стабільності та безперебійності роботи мережі, включаючи використання альтернативних джерел живлення, оптоволоконних технологій і автоматизованих систем моніторингу.

Оцінка економічної доцільності показала, що запропоноване рішення є рентабельним, із терміном окупності 1,8 року та рівнем економічної ефективності 0,56.

КЛЮЧОВІ СЛОВА: ЕНЕРГОНЕЗАЛЕЖНА МЕРЕЖА, ОПТОВОЛОКОННІ ТЕХНОЛОГІЇ, МАШИННЕ НАВЧАННЯ, ПРОГНОЗУВАННЯ НЕСПРАВНОСТЕЙ, ШВИДКЕ ДЕРЕВО.

The master's thesis is devoted to the study of methods and technologies for building a reliable energy-independent network infrastructure of the organization. The explanatory note to the final thesis contains 119 pages, 60 illustrations, 15 tables, 1 appendix and 32 literary sources.

The purpose of the work is to create and substantiate a complex energy-independent network infrastructure with fiber-optic connection of subscribers to ensure the continuity of Internet services and reduce the organization's energy dependence through the use of alternative power sources and automated monitoring systems.

The work defines the main tasks of the project, in particular, the development of an infrastructural solution using GPON and DWDM fiber optic technologies, the implementation of monitoring software with the functions of analyzing and forecasting network equipment malfunctions, as well as evaluating the economic feasibility of the project.

The main focus is on the analysis of modern solutions to ensure autonomy, stability and uninterrupted operation of the network, including the use of alternative power sources, fiber optic technologies and automated monitoring systems.

The economic feasibility assessment showed that the proposed solution is profitable, with a payback period of 1.8 years and an economic efficiency level of 0.56.

KEYWORDS: ENERGY-INDEPENDENT NETWORK, FIBER TECHNOLOGIES, MACHINE LEARNING, FAULT PREDICTION, FAST TREE.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	10
ВСТУП.....	11
РОЗДІЛ 1. ІНФРАСТРУКТУРНІ ТА ТЕХНОЛОГІЧНІ РІШЕННЯ ДЛЯ ЕНЕРГОНЕЗАЛЕЖНОЇ МЕРЕЖІ З ОПТОВОЛОКОННИМ ПІДКЛЮЧЕННЯМ АБОНЕНТІВ.....	14
1.1. Особливості оптоволоконного підключення для абонентів інтернет- провайдера.....	14
1.2. Енергоефективність та стабільність оптоволоконних мереж	16
1.3. Альтернативні джерела живлення для енергонебезпечності мережі	19
1.3.1. Сонячні панелі для енергозабезпечення вузлів мережі.....	19
1.3.2. Резервне живлення за допомогою вітрових турбін	22
1.4. Системи зберігання та розподілу енергії	24
1.4.1. Підтримка абонентських підключень за допомогою систем зберігання енергії.....	25
1.4.2. Power over Ethernet для забезпечення живлення абонентського обладнання	27
1.5. Автоматизоване управління енергорозподілом у вузлах	29
1.6. Дослідження діяльності організації та постановка задачі на розробку комплексного рішення	36
Висновок до першого розділу	40
РОЗДІЛ 2. МОДЕЛЮВАННЯ ІНФРАСТРУКТУРИ З АБОНЕНТСЬКИМ ПІДКЛЮЧЕННЯМ	41
2.1. Аналіз технічного завдання	41
2.2. Планування структури мережі	42
2.3. Конфігурація та налаштування роботи проекту мережі	49
2.3.1. Налаштування основних параметрів та безпеки пристроїв	51
2.3.2. Конфігурація VLAN та DHCP маршрутизаторів.....	52
2.3.3. Конфігурація VLAN на комутаторах точок доступу.....	55
2.3.4. Налаштування гіперканалу та VLAN комутатора виносу	56
2.3.5. Налаштування адресації хостів клієнтів мережі (DHCP).....	59

2.3.6. Налаштування адресації Wi-Fi вільної точки доступу	60
2.3.7. Контроль коректності налаштувань	61
2.3.8. Налаштування DNS-серверу	62
2.3.9. Налаштування маршрутизації по EIGRP	64
2.3.10. Налаштування NAT на головному маршрутизаторі	68
2.4. Тестування працездатності мережі	69
Висновок до другого розділу	73
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ОЦІНКА	
ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ ВПРОВАДЖЕННЯ ЕНЕРГОНЕЗАЛЕЖНОЇ	
МЕРЕЖНОЇ ІНФРАСТРУКТУРИ	
3.1. Аналіз вимог до програмного забезпечення для моніторингу мережі	75
3.2. Вибір алгоритмів машинного навчання для аналізу та прогнозування	
стану мережі	76
3.3. Попередня обробка даних та особливості роботи з датасетом	78
3.4. Розробка коду для тренування моделей для прогнозування стану	
мережі	81
3.5. Інтеграція моделі машинного навчання у програмне забезпечення	
моніторингу	88
3.6. Тестування програмного забезпечення та оцінювання точності	
прогнозів	91
3.7. Оцінка економічної доцільності впровадження запропонованого	
енергонеалежного рішень	97
3.7.1 Визначення стадій техпроцесу для побудови енергонеалежного	
мережної інфраструктури організації	98
3.7.2 Визначення загальних витрат на оплату праці та відрахувань	99
3.7.3 Розрахунок матеріальних витрат	101
3.7.4 Розрахунок витрат на електроенергію	103
3.7.5 Визначення транспортних затрат	104
3.7.6 Розрахунок суми амортизаційних відрахувань	104
3.7.7 Обчислення накладних витрат	105

3.7.8 Складання кошторису витрат та визначення собівартості впровадження	106
3.7.9 Розрахунок ціни запропонованого рішення.....	107
3.7.10 Визначення економічної ефективності і терміну окупності капітальних вкладень	108
Висновок до третього розділу	109
ВИСНОВКИ	111
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	113
ДОДАТКИ	117
Додаток А	118

ВЛ – Випадковий ліс.

ГБДР – Градієнтний бустинг дерев рішень.

МН – Машинне навчання.

ШД – Швидке дерево.

DNS – Domain Name System.

DWDM – Dense Wavelength Division Multiplexing.

EIGRP – Enhanced Interior Gateway Routing Protocol.

ENTSO-E – European Network of Transmission System Operators for Electricity.

GPON – Gigabit Passive Optical Network.

MAN – Metropolitan Area Network.

NAT – Network Address Translation.

PLC – Power Line Communications.

PoE – Power over Ethernet.

WAN – Wide Area Network.

Сучасні організації усе більше залежать від стабільної роботи інформаційних мереж, що забезпечують зв'язок, обмін даними та критичні бізнес-процеси. В умовах зростання потреб у високонадійних мережах, здатних працювати навіть за відсутності живлення, питання побудови енергонезалежної інфраструктури набуває особливої актуальності. Збої в мережах спричиняють економічні втрати, знижуючи ефективність і конкурентоздатність компаній, тому створення енергонезалежних систем стає важливим.

Часті перебої електропостачання, зокрема в Україні, підвищують актуальність впровадження таких технологій. Енергонезалежні мережі дозволяють забезпечити зв'язок навіть під час відключень, що є критичним для організацій, які надають життєво важливі послуги, та сприяють сталому розвитку цифрової економіки.

В умовах активного розвитку технологій, зокрема мережевого обладнання та джерел альтернативної енергії, такі як сонячні панелі, вітрові генератори та акумуляторні батареї, з'являються нові можливості для побудови гібридних систем, що здатні забезпечувати автономну роботу мереж. Використання альтернативних джерел енергії у поєднанні з традиційними підходами до управління мережевою інфраструктурою створює умови для значного підвищення стійкості мереж. Крім того, розробка резервних маршрутів та оптимізація рівня IP-маршрутизації забезпечує можливість безперервного доступу до ресурсів мережі навіть у разі виходу з ладу окремих компонентів системи [1].

У багатьох країнах світу уже успішно реалізовано проекти, які спрямовані на енергетичну автономність мережевих систем, особливо у віддалених та важкодоступних регіонах, де централізоване енергопостачання є нестабільним або відсутнім. Україна, як держава, що постійно працює над підвищенням власної енергетичної незалежності, може отримати значні переваги від впровадження енергонезалежних мережевих технологій на національному рівні. Такий підхід дозволить знизити залежність від зовнішніх джерел енергії та

забезпечити безперервність роботи критично важливих інфраструктур, особливо в умовах зростаючих кібератак та загроз стабільності енергомереж.

Попри очевидні переваги, впровадження енергонезалежної мережної інфраструктури вимагає ретельного планування, значних інвестицій та застосування сучасних підходів до управління мережею. Сучасні дослідження зосереджені на інтеграції інтелектуальних систем управління, які здатні автоматично адаптувати режим роботи мережі відповідно до рівня доступної енергії та поточного навантаження [2]. Важливим аспектом є також застосування методів машинного навчання для автоматичного моніторингу та прогнозування стану мережі, що дозволяє мінімізувати ризики відмови мережевих елементів та оптимізувати витрати на технічне обслуговування.

Об'єкт дослідження – процес побудови та підтримання енергонезалежної мережної інфраструктури організації з використанням оптоволоконних технологій та альтернативних джерел живлення.

Предмет дослідження – технологічні рішення та методи енергозабезпечення, що забезпечують автономність, стабільність і безперебійність роботи мережної інфраструктури.

Мета дослідження – створення та обґрунтування комплексної енергонезалежної мережної інфраструктури з оптоволоконним підключенням абонентів, для забезпечення безперервності надання інтернет-послуг та зменшення енергозалежності організації за рахунок використання альтернативних джерел живлення та автоматизованих систем моніторингу.

Для досягнення поставленої мети, необхідно вирішити наступні задачі:

- здійснити аналіз інфраструктурних та технологічних рішень для забезпечення енергонезалежності мережі з акцентом на оптоволоконні підключення абонентів;
- провести моделювання мережної інфраструктури для підтримання безперебійної роботи мережі;
- розробити алгоритми машинного навчання для прогнозування стану мережі та впровадження їх у програмне забезпечення для моніторингу;

- здійснити оцінювання економічної доцільності застосування енергонезалежних рішень, зокрема, ефективності використання альтернативних джерел живлення для ТОВ «Net City».

Основні методи дослідження:

- аналіз літературних джерел для вивчення наявних рішень та технологій;
- системний аналіз для визначення вимог до енергонезалежної мережі;
- моделювання мережної інфраструктури з використанням програмних засобів;
- методи машинного навчання для прогнозування стану мережі;
- економічний аналіз для оцінки доцільності впровадження рішень.

Практичне значення отриманих результатів полягає у можливості застосування розробленого комплексного рішення для створення енергонезалежної мережної інфраструктури в практичній діяльності підприємств, що надають інтернет-послуги. Розробка забезпечує безперебійність функціонування мережі завдяки використанню альтернативних джерел живлення, автоматизованих систем моніторингу та інтелектуальних алгоритмів прогнозування стану мережі, що дозволяє значно підвищити стабільність і надійність послуг. Зокрема, запропоноване рішення планується до впровадження у ТОВ «Net City», що дозволить цьому підприємству знизити залежність від основного енергопостачання та оптимізувати витрати на обслуговування мережі.

РОЗДІЛ 1. ІНФРАСТРУКТУРНІ ТА ТЕХНОЛОГІЧНІ РІШЕННЯ ДЛЯ ЕНЕРГОНЕЗАЛЕЖНОЇ МЕРЕЖІ З ОПТОВОЛОКОННИМ ПІДКЛЮЧЕННЯМ АБОНЕНТІВ

1.1. Особливості оптоволоконного підключення для абонентів інтернет-провайдера

Комп'ютерні мережі, включаючи глобальну мережу Internet, зазнали суттєвого розвитку в останні роки та демонструють високі показники продуктивності і пропускної здатності каналів передачі даних, що обумовлено значним зростанням обсягів інформаційних потоків. Історія розвитку інформаційних мереж розпочалася з 1980-х років, коли використовувались перші радіорелейні та каналні пристрої передачі з пропускною здатністю близько 0,5-1 кбіт/с та продуктивністю 1-5 тисяч оп./с. Наразі ці мережі трансформувались у найсучасніші комунікаційні системи та мультисервісні станції на основі волоконно-оптичних ліній зв'язку (ВОЛЗ), що досягають пропускної здатності 120-200 Гбіт/с на канал та продуктивності на рівні 10^{12} – 10^{18} оп./с. Такий значний приріст обумовлений високими частотами оптичних хвиль (до $2 \cdot 10^{14}$ Гц) та активним використанням методів паралельної обробки обчислень і потоків даних.

Сучасні мережі, що використовують волоконно-оптичні канали, включають у свою структуру технології ущільнення каналів, такі як TDM, FDM і WDM, а також протоколи TCP/IP, FDDI, FiberChannel, IEEE 802.2XX, IEEE 802.1XX. Поєднання цих технологій із ефективними апаратними засобами на основі ВОЛЗ дозволяє створювати високошвидкісні інформаційні мережі, що відповідають стандартам SDH (SONET) та ATM, і впроваджувати мультисервісні мережеві платформи [3-4].

Цифрова мультисервісна платформа мереж ВОЛЗ являє собою комплексну технічну інфраструктуру, що включає приймально-передавальні пристрої, оптичні мультиплексори та демультиплексори, волоконно-оптичні компоненти введення-виведення, систему автоматичного резервування каналів, апаратну агрегацію інформаційних потоків, системи віддаленого управління і сигналізації,

контрольно-вимірювальні прилади, пристрої службового зв'язку та систему живлення.

Підключення квартири до Інтернету через оптоволоконну лінію надає абоненту можливість залишатися онлайн навіть під час повного знеструмлення будинку. На відміну від рішень з використанням акумуляторів, що встановлюються на обладнанні провайдера, при оптоволоконному підключенні час автономної роботи не обмежується провайдером і не залежить від параметрів, як-от ємність, температура, кількість циклів заряду-розряду акумуляторів. Тривалість роботи визначається виключно ємністю та потужністю блока безперебійного живлення у абонента. Необхідною умовою для такого підключення є наявність у абонента власного блока безперебійного живлення, який забезпечуватиме живлення Wi-Fi роутера та оптичного конвертора. Окрім того, має бути можливість прокласти оптоволоконний кабель від поверхового електрощита до квартири абонента через наявний кабельний ввід без заламів або готовий кабельний короб (рис. 1.1).



Рисунок 1.1 – Приклад схеми оптоволоконного підключення квартири [5]

Без блока безперебійного живлення підключення через оптоволокну не надає значних переваг у порівнянні зі звичайним мідним підключенням типу «вита пара». Додатковою перевагою оптоволокну є його стійкість до електромагнітних впливів, зокрема, до розрядів блискавки та електромагнітних імпульсів. Це робить оптоволоконні мережі більш надійними та стійкими до зовнішніх електромагнітних перешкод, що особливо важливо в умовах підвищених вимог до безпеки інформаційних мереж та в місцях з високим рівнем

атмосферних явищ. Окрім того, оптоволокну забезпечує значно вищу швидкість передачі даних та стабільність сигналу порівняно з традиційними мідними кабелями. Завдяки відсутності електричного струму в оптичному волокні, практично виключається можливість втрат сигналу через нагрівання або електромагнітне випромінювання, що характерно для мідних кабелів. Це дозволяє забезпечити стабільне і швидкісне з'єднання незалежно від зовнішніх умов або відстані між вузлом провайдера і абонентом [6].

Загалом, оптоволоконне підключення є ідеальним вибором для користувачів, які потребують високу швидкість передачі даних, надійність з'єднання, а також додаткову стійкість до фізичних та електромагнітних завад.

1.2. Енергоефективність та стабільність оптоволоконних мереж

Оптоволоконні мережі зарекомендували себе як надзвичайно енергоефективна технологія для передачі даних на великі відстані, що забезпечує високу стабільність з'єднання. Завдяки низці технічних особливостей, таких як низьке енергоспоживання і високий рівень захисту від перешкод, вони стали стандартом для сучасних телекомунікаційних мереж. Енергоефективність оптоволоконних мереж полягає у здатності передавати великі обсяги даних з мінімальними втратами енергії та низьким споживанням електроенергії. На відміну від мідних кабелів, оптоволокну використовує світлові сигнали, що проходять через спеціальні скляні або пластикові волокна, практично не втрачаючи інтенсивності на великих відстанях. Це забезпечує стабільне з'єднання навіть у складних умовах і знижує необхідність у підсилювачах сигналу, що додатково зменшує енергоспоживання. Відсутність електромагнітних перешкод у оптоволоконних лініях також сприяє збереженню енергії, адже немає потреби у витратах на боротьбу з шумами та інтерференцією. Завдяки високій пропускну здатності, оптоволокну може замінити декілька мідних кабелів, що оптимізує використання апаратних ресурсів і зменшує споживання енергії всією мережею.

Для кращого розуміння енергоефективності оптоволоконних мереж наведено ключові аспекти у табл. 1.1.

Основні аспекти енергоефективності оптоволоконних мереж

Аспект	Опис	Переваги	Вплив
Низькі втрати сигналу	Світловий сигнал передається без втрат на великі відстані	Стабільність з'єднання	Зниження енергоспоживання
Відсутність електромагнітних перешкод	Нечутливість до електромагнітних полів	Відсутність перешкод	Менше потреби в обробці перешкод
Зменшена потреба в підсилювачах	Мінімальна кількість підсилювачів через низькі втрати сигналу	Економія на обладнанні	Зменшення загальних витрат енергії
Висока пропускна здатність	Можливість передавати великі обсяги даних	Скорочення кількості кабелів	Оптимізація енергоспоживання
Тривалий термін служби	Стійкість до корозії та механічних пошкоджень	Зниження витрат на обслуговування	Ефективне використання ресурсів
Екологічна безпека	Зменшення викидів вуглецю та впливу на навколишнє середовище	Енергоефективне рішення	Зниження енергоспоживання
Мінімальна витрата ресурсів	Використання менше апаратних компонентів	Оптимізація мережевих ресурсів	Підвищення загальної ефективності

Енергоефективність оптоволоконних мереж обумовлена сукупністю технологічних характеристик, що сприяють зменшенню енергоспоживання і підвищенню стабільності передачі даних. Завдяки зниженим втратам сигналу, відсутності потреби в частих підсилювачах, високій пропускній здатності та тривалому терміну служби, оптоволоконні мережі стають оптимальним вибором для сучасних телекомунікаційних систем.

Стабільність оптоволоконних мереж забезпечується завдяки їхній нечутливості до електромагнітних завад, відсутності втрат сигналу на великих відстанях та стійкості до зовнішніх впливів. На відміну від мідних мереж, які можуть страждати від завад, викликаних електричними і магнітними полями,

оптоволокну передає дані за допомогою світлових хвиль, що робить його майже невразливим до зовнішніх факторів. Це дозволяє підтримувати стабільний зв'язок навіть у промислових зонах або поблизу інших джерел електромагнітних впливів. У табл. 1.2 приведено основні фактори стабільності оптоволоконних мереж.

Таблиця 1.2

Основні фактори стабільності оптоволоконних мереж

Фактор	Опис	Переваги	Вплив
Нечутливість до завад	Стійкість до електромагнітних полів	Висока якість сигналу	Стабільний зв'язок незалежно від оточення
Механічна стійкість	Волокно не піддається корозії або зносу	Довговічність	Зменшення потреби в обслуговуванні
Стійкість до кліматичних умов	Не піддається впливу температур і вологості	Стабільність в різних умовах	Висока надійність навіть в екстремальних умовах
Відсутність втрат сигналу	Мінімальні втрати сигналу на великих відстанях	Якість передачі даних	Постійно висока швидкість і якість передачі
Захист від фізичних пошкоджень	Захищена конструкція кабелю	Захист від механічних впливів	Підвищення загальної стабільності

Стабільність оптоволоконних мереж забезпечується їх стійкістю до зовнішніх перешкод, зменшеними втратами сигналу та довговічністю матеріалів. Відсутність електромагнітних завад, механічна надійність та відмінна пропускну здатність роблять оптоволоконні мережі надійним вибором для забезпечення якісного з'єднання. Завдяки цьому оптоволоконні мережі є оптимальним рішенням для корпоративних, державних та індустріальних користувачів, які потребують постійного і стабільного з'єднання без втрат якості [7].

Отже, енергоефективність та стабільність оптоволоконних мереж роблять їх ідеальним вибором для сучасних телекомунікаційних систем, які прагнуть до

надійності, стійкості до зовнішніх впливів та мінімізації витрат на експлуатацію. Завдяки своїм технічним характеристикам оптоволокну забезпечує не лише високу швидкість передачі даних, але й суттєво знижує витрати енергії, сприяючи сталому розвитку та екологічній безпеці.

1.3. Альтернативні джерела живлення для енергонезалежності мережі

Для забезпечення безперервної роботи оптоволоконних мереж у разі перебоїв із електропостачанням важливим є використання альтернативних джерел живлення, які дозволяють досягти енергонезалежності [30]. Використання резервного живлення забезпечує можливість підтримання мережевої інфраструктури в робочому стані навіть за умов надзвичайних ситуацій, зокрема відключення електроенергії, стихійних лих чи техногенних аварій. У роботі розглянуто основні альтернативні джерела живлення, які можуть бути використані для енергонезалежності мережевих систем.

1.3.1. Сонячні панелі для енергозабезпечення вузлів мережі

Сонячні панелі є одним з найефективніших рішень для забезпечення автономного енергопостачання вузлів мережевої інфраструктури. Використання сонячної енергії дозволяє підтримувати роботу мереж навіть у віддалених регіонах або в умовах, коли доступ до традиційного електропостачання є обмеженим або ненадійним. Сонячні панелі перетворюють сонячне випромінювання на електричну енергію, яка може бути безпосередньо використана для живлення комунікаційного обладнання або збережена у спеціальних акумуляторах для подальшого використання в нічний час або в похмуру погоду. Завдяки низьким витратам на обслуговування та довгому терміну служби, сонячні панелі є економічно доцільним рішенням для багатьох організацій.

Однією з важливих складових такої системи є контролер заряду, який регулює процес заряджання акумуляторів, захищаючи їх від надмірного або

недостатнього заряду. Це забезпечує ефективне зберігання енергії та продовжує термін експлуатації акумуляторних систем. Для забезпечення стабільного енергопостачання також важливим є комутаційне обладнання, яке забезпечує безперервне з'єднання між сонячними панелями, акумуляторами та іншими компонентами системи. Таке обладнання дозволяє адаптуватися до різних умов навколишнього середовища і забезпечує надійність мережі.

Як показано на рис. 1.2, стандартна система енергозабезпечення для вузлів мережі за допомогою сонячних панелей включає кілька ключових компонентів.

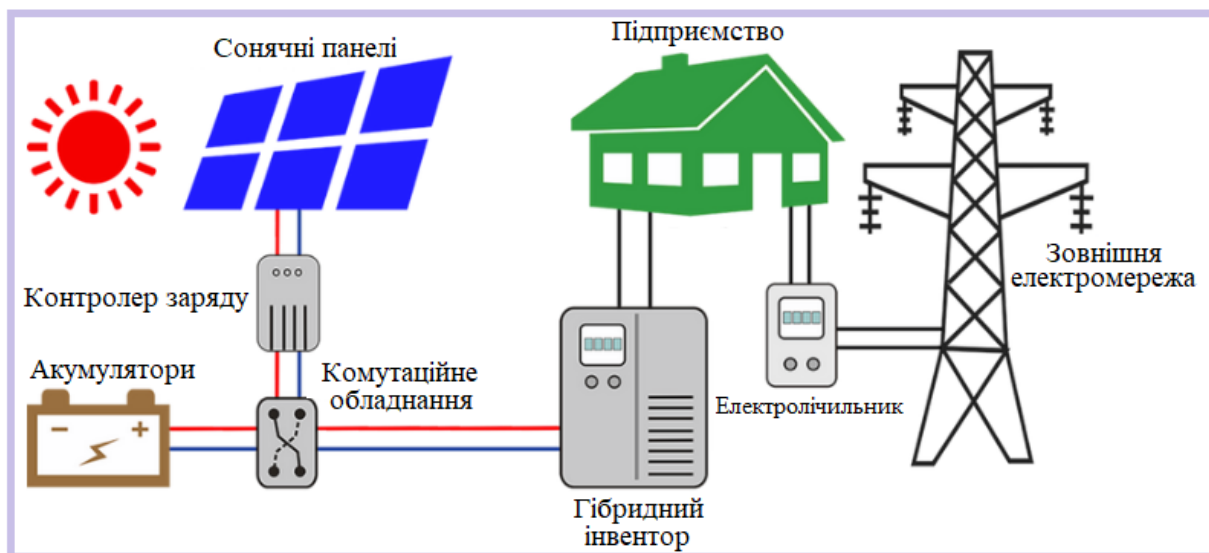


Рисунок 1.2 – Енергозабезпечення за допомогою сонячних панелей

Сонячні панелі генерують енергію, яка подається на контролер заряду, що регулює заряджання акумуляторів, запобігаючи їх пошкодженню. Комутаційне обладнання здійснює передачу енергії від сонячних панелей та акумуляторів до гібридного інвертора. Гібридний інвертор виконує перетворення постійного струму від панелей та акумуляторів на змінний струм, придатний для живлення обладнання у мережі. Окрім того, він може використовувати зовнішню електромережу як резервне джерело живлення, підключаючи його у разі нестачі енергії, накопиченої сонячними панелями. Електролічильник відображає кількість спожитої енергії з зовнішньої електромережі, якщо така потреба виникає [8].

Така система є надійним та екологічно чистим варіантом для забезпечення безперебійного живлення мережевих вузлів, що дозволяє значно підвищити енергонезалежність організації та знизити витрати на електроенергію.

До переваг енергозабезпечення за допомогою сонячних панелей для енергонезалежності мережі підприємства:

- екологічність (сонячні панелі не виробляють викидів вуглекислого газу чи інших забруднюючих речовин під час роботи, що робить їх екологічно чистим джерелом енергії);
- низькі експлуатаційні витрати (після встановлення сонячних панелей витрати на обслуговування мінімальні, оскільки панелі мають довгий термін служби та потребують мінімального обслуговування);
- доступ до відновлюваної енергії (сонячні панелі використовують енергію сонця, яка є відновлюваним ресурсом, що робить підприємство менш залежним від традиційних джерел енергії);
- автономність (становлення сонячних панелей дозволяє підприємству зменшити залежність від зовнішніх джерел електропостачання, забезпечуючи резервну або основну підтримку енергії в разі збоїв в основній мережі).

Недоліки енергозабезпечення за допомогою сонячних панелей для енергонезалежності мережі підприємства:

- залежність від погодних умов (ефективність сонячних панелей значно знижується у похмуру погоду або вночі, що може потребувати додаткових джерел енергії для забезпечення стабільного живлення);
- високі початкові витрати (вартість встановлення сонячних панелей, інверторів і акумуляторних систем є досить високою, що може бути суттєвим фінансовим бар'єром для підприємства);
- необхідність в акумуляторах для зберігання енергії (для забезпечення безперебійного живлення у нічний час або під час негоди потрібні акумулятори, що збільшує витрати на обладнання та обслуговування);
- займана площа (для встановлення сонячних панелей потрібна велика площа, що може бути проблемою для підприємств із обмеженими земельними ресурсами або недостатнім простором на дахах будівель) [9].

Ці фактори слід враховувати при плануванні впровадження сонячних панелей як джерела резервного або основного енергопостачання для забезпечення енергонезалежності мережі підприємства.

1.3.2. Резервне живлення за допомогою вітрових турбін

Вітрові турбіни є ефективним джерелом резервного живлення для мережевої інфраструктури, особливо в місцевостях, де є стабільні вітрові потоки. Використання вітрових турбін дозволяє отримувати електроенергію за рахунок кінетичної енергії вітру, яка приводить у дію лопаті турбіни, що, у свою чергу, генерує електричний струм. Такий спосіб енергозабезпечення є екологічно чистим та економічно вигідним, оскільки він не потребує використання пального і не призводить до викидів вуглекислого газу. Накопичена за допомогою вітрових турбін енергія може використовуватися як основне або резервне джерело живлення для мереж, що робить цей підхід гнучким та універсальним [10].

Як видно на рис. 1.3, система резервного енергопостачання на основі вітрових турбін включає кілька ключових компонентів. Генерація енергії здійснюється вітровими турбінами, які конвертують кінетичну енергію вітру в електричну. Ця електроенергія передається до контролера, що регулює заряджання акумуляторів, забезпечуючи їхню ефективну роботу та захист від перенапруги. Акумулятори накопичують енергію, яку можна використовувати під час періодів відсутності вітру або в моменти пікового навантаження на систему.

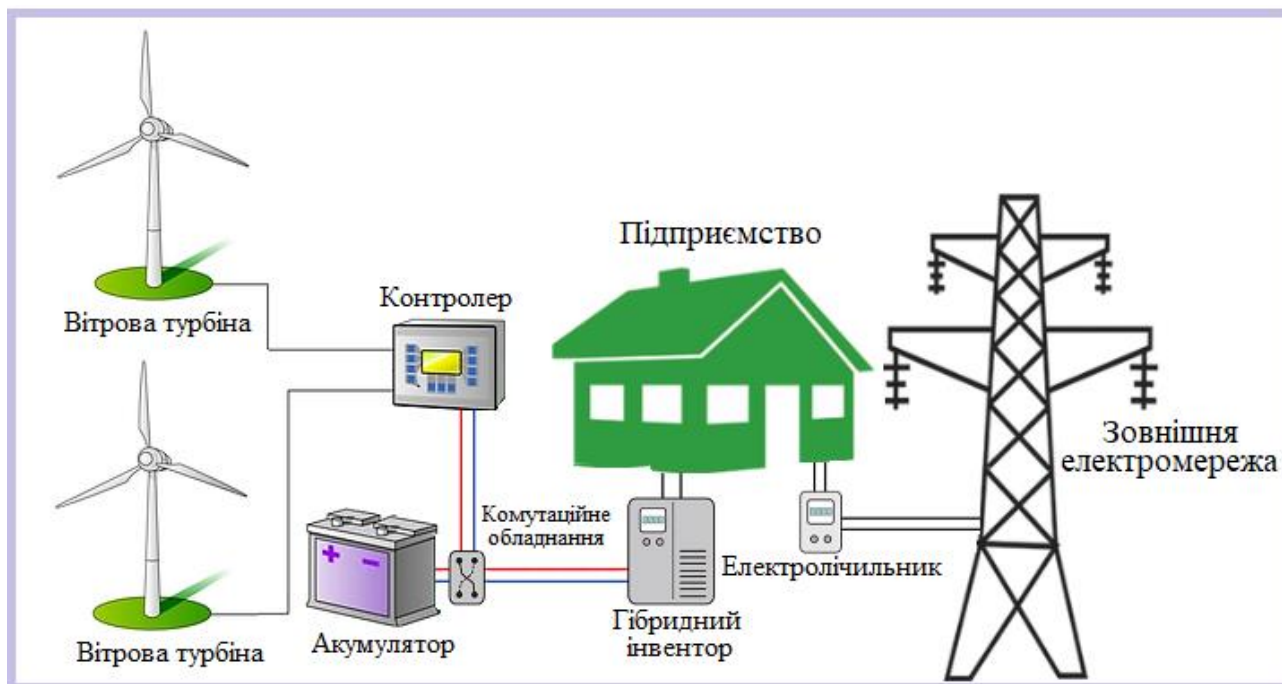


Рисунок 1.3 – Енергозабезпечення за допомогою вітрових турбін

Комутаційне обладнання забезпечує безперервне з'єднання між компонентами, направляючи електричний струм від турбін до гібридного інвертора. Гібридний інвертор виконує перетворення постійного струму з акумуляторів на змінний, придатний для живлення електроприладів підприємства. У разі недостатнього обсягу енергії, накопиченої в акумуляторах, інвертор автоматично підключає зовнішню електромережу як резервне джерело живлення, що забезпечує стабільність і безперервність енергопостачання.

Система за допомогою вітрових турбін дозволяє значно підвищити енергонезалежність підприємства, знижуючи витрати на електроенергію та забезпечуючи екологічно чисте джерело живлення. Використання вітрових турбін у якості резервного живлення сприяє стабільності мережевої інфраструктури та є ефективним рішенням для забезпечення надійного енергопостачання [11].

Переваги резервного живлення за допомогою вітрових турбін для енергонезалежності мережі підприємства включають:

- екологічність (вітрові турбіни використовують кінетичну енергію вітру, що є відновлюваним і екологічно чистим джерелом енергії, яке не спричиняє викидів вуглекислого газу);
- зниження витрат на електроенергію (вітрова енергія безкоштовна після встановлення обладнання, що дозволяє підприємству знизити витрати на електроенергію в довгостроковій перспективі);
- автономність та енергонезалежність (використання вітрових турбін як резервного джерела живлення дозволяє підприємству забезпечувати стабільне енергопостачання в разі збоїв основної мережі);
- довгий термін служби (вітрові турбіни мають тривалий термін служби та при належному обслуговуванні можуть ефективно працювати понад 20 років, забезпечуючи підприємство енергією протягом усього цього періоду).

До недоліків резервного живлення за допомогою вітрових турбін належать:

- залежність від погодних умов (ефективність вітрових турбін залежить від наявності стабільних вітрових потоків; за відсутності вітру генерація енергії припиняється);
- високі початкові витрати (встановлення вітрових турбін є капіталомістким процесом, що включає витрати на обладнання, монтаж та підключення до мережі);
- проблеми з шумом і естетикою (вітрові турбіни можуть створювати значний рівень шуму і можуть бути непривабливими з естетичної точки зору, що може бути проблемою в населених районах);
- вимоги до простору (для встановлення турбін потрібна достатня площа з оптимальними умовами для роботи, що може бути недоступним для деяких підприємств).

Використання вітрових турбін для резервного живлення може бути ефективним рішенням для забезпечення енергонезалежності підприємства, особливо в регіонах з постійними вітровими потоками. Ця технологія дозволяє знизити залежність від традиційних джерел енергії та зменшити викиди вуглекислого газу, сприяючи сталому розвитку. Проте, перед впровадженням такого рішення важливо врахувати потенційні недоліки, такі як залежність від погодних умов і високі початкові витрати [12]. Загалом, вітрові турбіни є перспективним, але не універсальним варіантом резервного енергопостачання, що потребує попереднього аналізу доцільності та умов експлуатації.

1.4. Системи зберігання та розподілу енергії

Системи зберігання та розподілу дозволяють накопичувати енергію з різних джерел, таких як сонячні панелі чи вітрові турбіни, та розподіляти її відповідно до потреб мережевої інфраструктури, забезпечуючи безперебійне живлення в умовах нестабільного зовнішнього електропостачання. Основними складовими таких систем є акумуляторні батареї, інвертори, контролери заряду та комутаційне обладнання, кожне з яких виконує специфічні функції для ефективного управління енергією.

1.4.1. Підтримка абонентських підключень за допомогою систем зберігання енергії

Системи зберігання енергії дозволяють забезпечувати надійне та безперервне живлення для абонентських підключень, що особливо важливо в умовах нестабільного енергопостачання. Завдяки використанню резервних акумуляторних систем, абоненти можуть залишатися підключеними до мережі навіть під час відключень електроенергії або аварійних ситуацій. Такі системи забезпечують необхідну потужність для підтримки роботи комунікаційного обладнання, що робить зв'язок більш стійким і надійним.

Одним із прикладів технологій для абонентських підключень, які використовують можливості систем зберігання енергії, є HomePlug (PLC, Power Line Communications). Вона працює на основі використання існуючих електричних ліній об'єднаної енергетичної системи України, яка з 2022 року інтегрована до європейської енергосистеми ENTSO-E, для передачі даних. Напруга передається до абонентських пристроїв через стандартну електромережу з напругою 230 В і частотою 50 Гц. У технології HomePlug використовується ієрархічна модель передачі даних: сигнали передаються від центрального вузла до проміжної розподільчої станції, а звідти – до кінцевого абонентського пристрою [13].

Центральний комутатор PLC з'єднаний з глобальною мережею WAN або регіональною мережею MAN за допомогою волоконно-оптичного кабелю, забезпечуючи абонентам доступ до високошвидкісного інтернету (рис. 1.4). Технологія HomePlug базується на магістральних і кінцевих вузлах мережі PLC, які виконують роль комутаторів і шлюзів для підключення абонентських пристроїв. Розміщення магістрального комутатора розраховане на обслуговування певної зони з фіксованою кількістю користувачів (N).

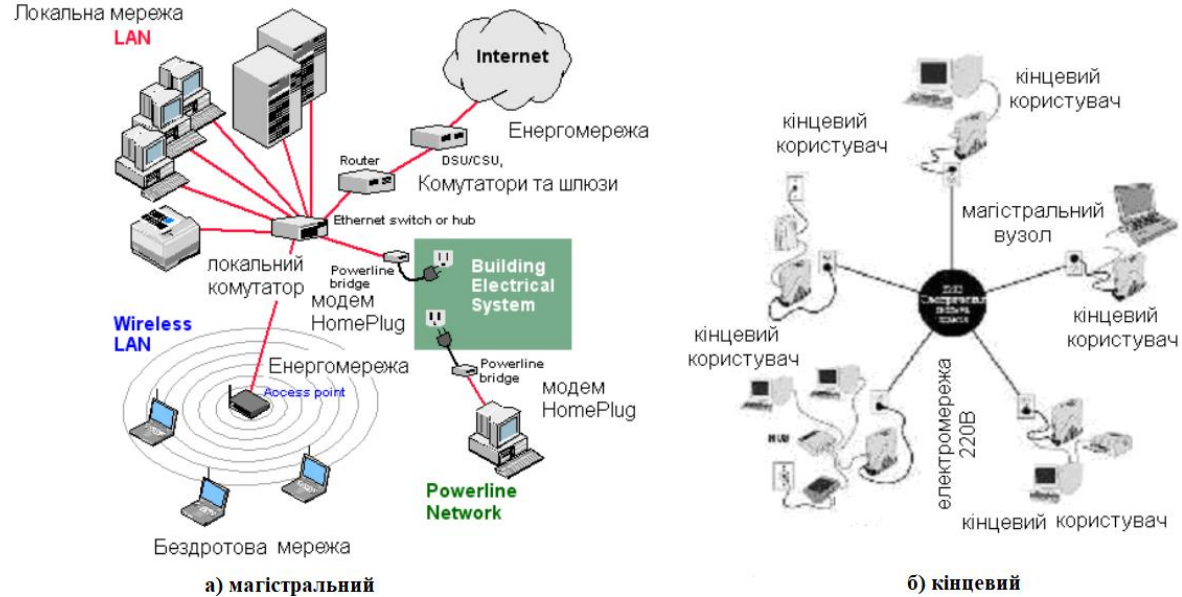


Рисунок 1.4 – Вузли мережі HomePlug

Кінцевий вузол мережі PLC представлений абонентським модемом, який використовує метод частотного розділення каналів (FDM) на основі стандартів ADSL, XDSL або VDSL. Модем PLC забезпечує інтерфейс для підключення до персональних комп'ютерів через USB або Ethernet. Вхід модему підключається безпосередньо до побутової електромережі на 220 В. HomePlug є однією з перших технологій, яка об'єднує мережеву інфраструктуру з широкодоступною електроенергетичною мережею, забезпечуючи зручне та стабільне підключення для користувачів.

Стандарт HomePlug реалізує модель множинного доступу «точка - множина точок» і може використовуватися для підвищення стабільності передачі та обробки інформаційних потоків у комп'ютерних мережах. Проте, ця технологія має певні суттєві недоліки:

- високий рівень шумів у каналах передачі даних, які виникають через частотні гармоніки в каналах електромережі, з якими вони поєднані;
- обмежена швидкість передачі даних (10-100 Мбіт/с), зумовлена вузькою смугою частот, властивою електромережам 220В 50Гц, що не призначені для інформаційних передач;
- недостатній рівень стабільності в передачі та обробці даних, що пов'язано з ієрархічною структурою подачі енергії в електромережі, яка має вертикальний розподіл від центральних енергостанцій до кінцевих користувачів.

До того ж, впровадження HomePlug обмежується максимальною дальністю передачі сигналу (до 1-2 км), що залежить від рівня завад у мережі з напругою ~220В. Ці обмеження значно ускладнюють використання HomePlug як фізичної мережі для побудови інфраструктур моніторингу та передачі даних, а також для геоінформаційних систем, які потребують стабільної роботи та розширених функціональних можливостей [14].

Отже, технологія HomePlug, хоч і забезпечує доступність підключень через існуючу електромережу, має низку обмежень, зокрема низьку швидкість передачі, високий рівень шумів і обмежену дальність сигналу. Ці фактори знижують її ефективність для створення стабільних інформаційних мереж, особливо у випадках, коли потрібна висока надійність та функціональність передачі даних. Для складних інфраструктурних завдань можуть бути необхідні альтернативні або додаткові рішення, які зможуть забезпечити кращу якість обслуговування і стабільність мережевих з'єднань.

1.4.2. Power over Ethernet для забезпечення живлення абонентського обладнання

Одним із способів об'єднаного передавання даних та живлення для абонентського обладнання є технологія Power over Ethernet (PoE). Ця технологія дозволяє одночасно передавати інформацію та електричну енергію (при напрузі 48 В) до кінцевих мережевих пристроїв за допомогою стандартного Ethernet-кабелю. Концепція PoE полягає в тому, що такі компоненти, як точки доступу WLAN, проміжні ретранслятори, IP-камери та мережеві комутатори, можуть отримувати необхідне живлення через один кабель, який використовується також для передачі даних (рис. 1.5). Це спрощує процес підключення обладнання та знижує потребу в окремих електричних розетках для кожного пристрою.

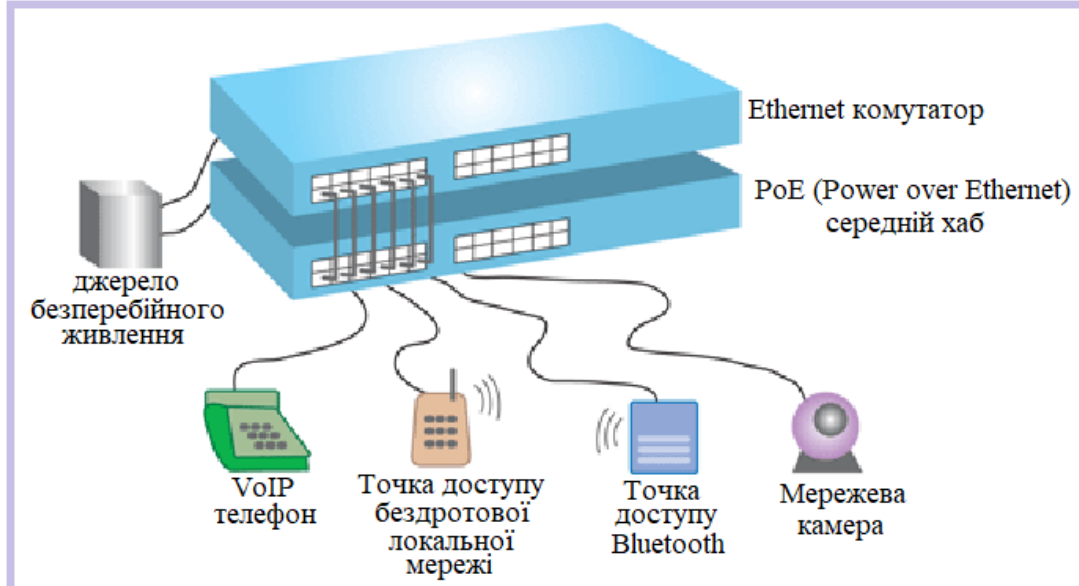


Рисунок 1.5 – Приклад підключення абонентських пристроїв до PoE [15]

Стандарт PoE, згідно зі специфікацією 802.3af, передбачає два типи пристроїв для подачі живлення (Power Sourcing Equipment, PSE): кінцеві пристрої (End-Span) і проміжні пристрої (Mid-Span). До End-Span належать Ethernet-комутатори з інтегрованою функцією передачі живлення через мережевий кабель. Mid-Span пристрої є концентраторами PoE з кількістю портів від 6 до 24, які розташовуються між комутатором і кінцевими пристроями, такими як точки доступу WLAN, VoIP-телефони або IP-камери, забезпечуючи передачу даних і живлення через комбінований вихід RJ45.

Номінальна сила струму для кожного вузла при напрузі 48 В постійного струму обмежена 350 мА, що відповідає вимогам безпеки. Максимальна потужність через кабель Ethernet складає близько 16,8 ВА ($P=48 \times 0.35=16.8$ ВА), тоді як фактична споживана потужність на кінцевому пристрої досягає 13 Вт з урахуванням втрат у кабелі до 100 м. Споживання точок доступу та IP-камер становить від 3,5 до 9 Вт.

PoE розроблена на основі стандартів HomePNA, який спочатку використовувався для передачі голосових даних через телефонні лінії. З часом HomePNA став недостатньо ефективним для сучасних мереж Ethernet через обмеження швидкості та дальності. Технологія USB також може забезпечувати живлення для пристроїв з робочою напругою 5 В, але, як показано в табл. 1.3, USB має обмежене побутове застосування, тоді як PoE є промисловим стандартом з вищою надійністю для корпоративних мереж [16-17].

Порівняння основних параметрів PoE та USB

Технічні показники	PoE	USB
Довжина кабелю, м	До 120-150	До 5
Макс. потужність споживання, Вт	15	2.5
Швидкість передачі (Мбіт/с)	1000	11
Напруга живлення, В	48	5
Максимальний робочий струм, А	0.35	0.5
Максимальна бітова швидкість даних	10 Гбіт/с	480 Мбіт/с

Інтеграція зв'язку в енергетичні канали електромереж може вважатися частковим рішенням для створення інформаційно-енергетичних мереж. Проте такі рішення не здатні забезпечити інтелектуальне управління інформаційно-енергетичними ресурсами, що є важливим для обробки зростаючих обсягів даних у сучасних магістральних мережах.

Таким чином, технологія PoE, завдяки своїм технічним характеристикам і здатності передавати як дані, так і живлення на значні відстані, є надійним промисловим стандартом для мережевих рішень. На відміну від USB, який придатний лише для побутових застосувань з обмеженою довжиною кабелю та низькою потужністю, PoE забезпечує стабільне і енергоефективне підключення, що задовольняє вимоги корпоративної та промислової інфраструктури.

1.5. Автоматизоване управління енергорозподілом у вузлах

Автоматизоване управління енергорозподілом у вузлах мережі дозволяє ефективно контролювати потоки енергії в різних сегментах мережі, оптимізуючи споживання та підтримуючи стабільне живлення для критичних компонентів, таких як комутатори, сервери та інші вузли мережі. Це особливо важливо для інфраструктури з розподіленими джерелами живлення, зокрема сонячними панелями, вітровими турбінами або іншими альтернативними джерелами енергії [31].

Для забезпечення стабільності та ефективності роботи мережі з розподіленими джерелами живлення часто використовуються комплексні рішення, які дозволяють централізовано керувати енергорозподілом, моніторити

споживання та автоматизувати управління енергією. Одним з таких рішень є EcoStruxure від компанії Schneider Electric.

EcoStruxure – це інтегрована платформа для управління енергетичними ресурсами, створена для забезпечення надійності та ефективності в енергетичних і промислових системах (рис. 1.6). Вона призначена для моніторингу, управління, аналізу і оптимізації енергоспоживання в режимі реального часу, що дозволяє підприємствам знижувати енергозатрати, мінімізувати ризики простоїв і покращувати експлуатаційну ефективність [18].

Архітектура EcoStruxure побудована на багаторівневому принципі, що включає пристрої, системи управління та аналітичні платформи. На першому рівні розташовані інтелектуальні пристрої, такі як датчики та вимірювальні прилади, що збирають дані з усього об'єкта. Ці дані передаються на рівень управління, де містяться контролери і програмні рішення, які відповідають за обробку та розподіл енергії. Наступний рівень – це аналітична платформа, яка забезпечує зберігання, аналіз даних і підтримку рішень за допомогою інструментів штучного інтелекту і машинного навчання. Платформа також інтегрується з хмарними сервісами, що дозволяє здійснювати моніторинг і управління віддалено, забезпечуючи повний контроль і прозорість у використанні енергії.

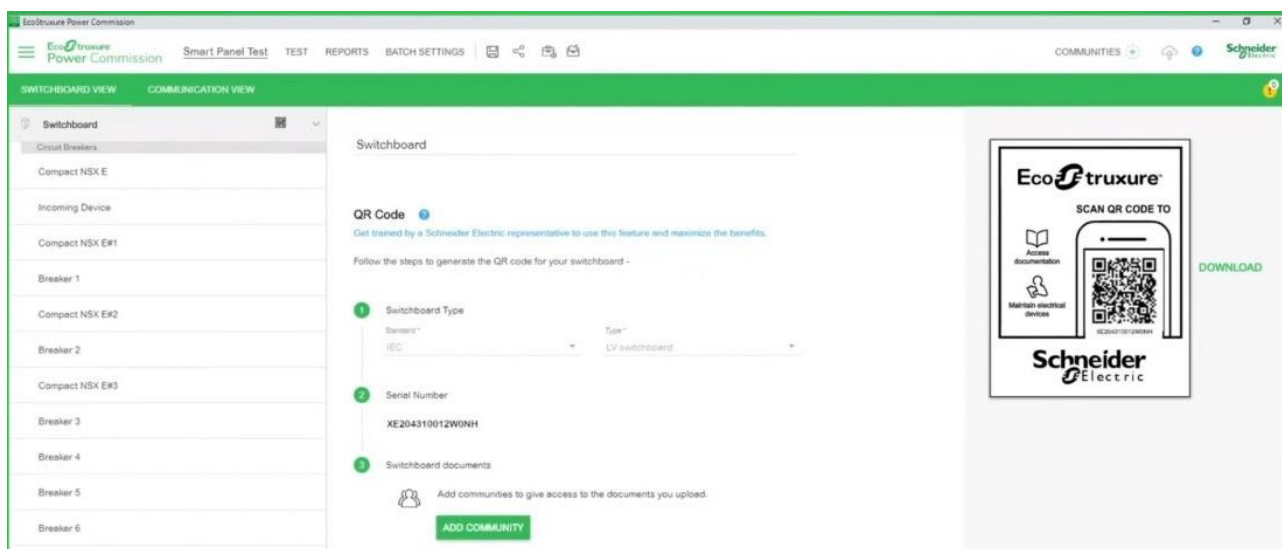


Рисунок 1.6 – Приклад користувацького інтерфейсу «EcoStruxure» [19]

EcoStruxure створена для гнучкої інтеграції з різними джерелами енергії та елементами інфраструктури, що робить її універсальним інструментом для підвищення енергоефективності в мережах будь-якого масштабу.

Переваги EcoStruxure включають:

- централізоване управління та моніторинг (платформа забезпечує централізований контроль над енергоспоживанням та роботою обладнання, що підвищує ефективність управління);
- віддалений доступ і моніторинг (інтеграція з хмарними сервісами дозволяє здійснювати управління віддалено, що зручно для великих підприємств з географічно розподіленими об'єктами);
- інтелектуальний аналіз даних (завдяки використанню інструментів штучного інтелекту та машинного навчання платформа здатна аналізувати споживання енергії та пропонувати оптимальні рішення для його зменшення);
- гнучка інтеграція з різними джерелами енергії (EcoStruxure підтримує роботу з різними типами енергетичних систем, включаючи відновлювані джерела, що підвищує загальну стабільність енергопостачання).

До недоліків EcoStruxure входять:

- висока вартість впровадження (через складну архітектуру і широкі функціональні можливості впровадження EcoStruxure може бути дорогим для малих та середніх підприємств);
- вимоги до інфраструктури (для повноцінного використання функцій EcoStruxure необхідно мати сучасну інфраструктуру з відповідним обладнанням та інтерфейсами, що може потребувати додаткових витрат);
- залежність від інтернет-з'єднання (хмарна інтеграція вимагає стабільного інтернет-з'єднання для доступу до віддалених функцій, що може бути проблемою в регіонах з нестабільним доступом до мережі);
- складність обслуговування (система потребує кваліфікованих фахівців для управління і підтримки, що може збільшити витрати на персонал).

Отже, EcoStruxure є потужним рішенням для автоматизованого управління енергорозподілом, яке поєднує в собі гнучкість, функціональність та високу ефективність. Завдяки можливостям централізованого моніторингу,

інтелектуального аналізу та підтримки різних джерел енергії, платформа є ефективним інструментом для великих підприємств, які прагнуть до енергоефективності та надійності. Проте, високі початкові витрати та вимоги до інфраструктури можуть обмежити її застосування для невеликих компаній, що вимагає ретельного аналізу доцільності впровадження цієї технології.

ABB Ability – це інтегрована платформа для моніторингу та управління енергетичними процесами, створена для підвищення ефективності та надійності в енергетичних і промислових системах (рис. 1.7). Призначення ABB Ability полягає у забезпеченні комплексного контролю над енергоспоживанням, управлінні розподілом енергії, а також у виявленні та усуненні аномалій, що дозволяє підприємствам оптимізувати витрати на енергію та зменшити екологічний слід. Платформа підходить для використання в різних галузях, зокрема в промисловості, енергетиці, транспорті та інфраструктурі.

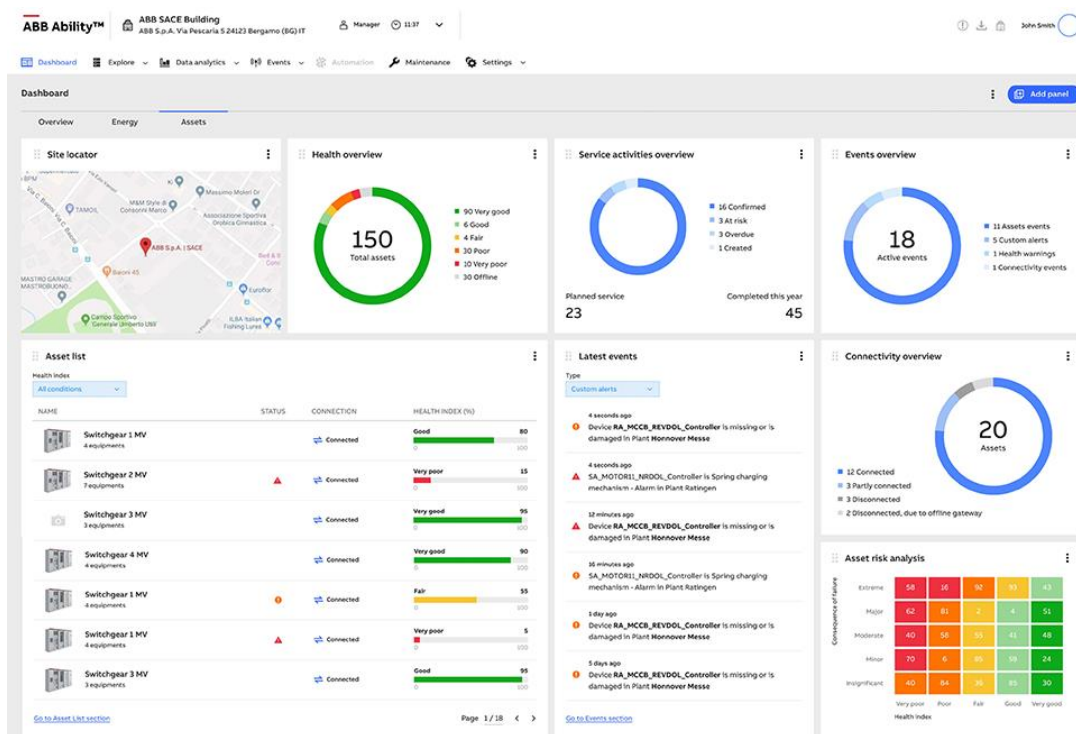


Рисунок 1.7 – Приклад користувацького інтерфейсу «ABB Ability» [20]

Архітектура ABB Ability базується на багаторівневій структурі, що включає інтелектуальні пристрої, системи управління та аналітичні інструменти. Перший рівень складається з різноманітних сенсорів та приладів, які збирають інформацію про стан системи, наприклад, показники температури, напруги, тиску тощо. Ці дані передаються на рівень управління, де контролери

обробляють і аналізують інформацію, координуючи роботу різних частин системи. Далі дані потрапляють на рівень аналітики, який працює з великими обсягами інформації, виконуючи прогнозування і підтримку прийняття рішень за допомогою інструментів штучного інтелекту. Хмарна інтеграція платформи забезпечує віддалений доступ до даних, дозволяючи керувати об'єктами з будь-якої точки світу та підтримувати гнучкість і масштабованість системи [21].

Переваги ABB Ability:

- висока масштабованість (платформа підтримує розширення на різні види обладнання та інфраструктури, що дозволяє адаптувати її для підприємств різного масштабу);
- віддалений доступ і моніторинг (хмарна інтеграція забезпечує доступ до даних у реальному часі з будь-якої точки світу, що зручно для керування розподіленими об'єктами);
- інтелектуальний аналіз даних (ABB Ability використовує інструменти штучного інтелекту та машинного навчання для прогнозування споживання та оптимізації енергетичних процесів, підвищуючи ефективність);
- сумісність з різними галузями (платформа підходить для різних секторів – промисловість, енергетика, транспорт, інфраструктура, що розширює її функціональні можливості).

Недоліки ABB Ability:

- висока вартість впровадження (установка та налаштування системи можуть бути дорогими, що може стати бар'єром для малих та середніх підприємств);
- вимоги до інфраструктури (для повноцінного використання функціоналу потрібна сучасна інфраструктура та відповідне обладнання, що може потребувати додаткових інвестицій);
- залежність від стабільного інтернет-з'єднання (хмарна інтеграція залежить від надійного інтернет-з'єднання, що може бути проблемою в деяких регіонах);

– необхідність кваліфікованого персоналу (для обслуговування та налаштування платформи потрібні фахівці, що може підвищити витрати на навчання та технічне забезпечення).

Отже, ABB Ability є потужною платформою для управління енергетичними процесами, що забезпечує інтелектуальний аналіз, масштабованість та гнучкість для підприємств у різних галузях. Завдяки інструментам для віддаленого моніторингу та аналізу в реальному часі, платформа допомагає підвищити енергоефективність та зменшити експлуатаційні витрати. Проте висока вартість впровадження та вимоги до інфраструктури можуть обмежити її доступність для невеликих компаній, що вимагає ретельного аналізу доцільності її використання.

Honeywell Forge – це комплексна платформа для управління операційними процесами та енергоресурсами, яка забезпечує підвищення ефективності та зниження витрат у промислових, комерційних та інфраструктурних системах (рис. 1.8). Призначення Honeywell Forge полягає в оптимізації виробничих і операційних процесів за допомогою моніторингу в реальному часі, автоматизації управління енергорозподілом і аналізу продуктивності, що дозволяє підприємствам підвищувати енергоефективність, зменшувати простой та мінімізувати експлуатаційні витрати.

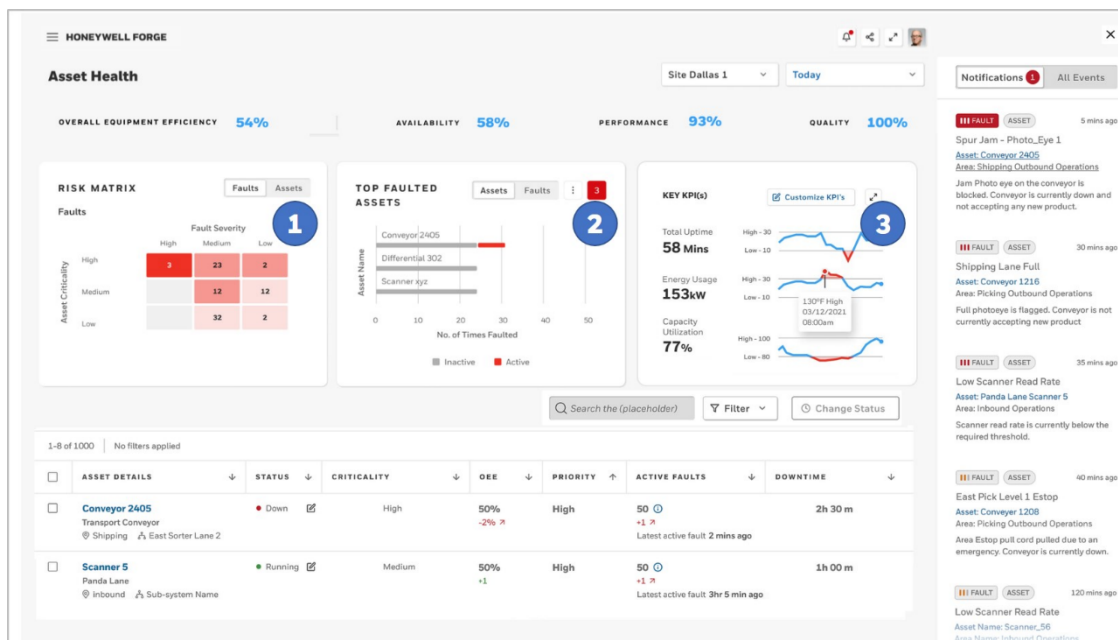


Рисунок 1.8 – Приклад користувацького інтерфейсу «Honeywell Forge» [22]

Архітектура Honeywell Forge базується на багаторівневій структурі, яка включає пристрої збору даних, аналітичні модулі та хмарні рішення. На першому рівні працюють сенсори та контролери, що збирають дані про стан обладнання, параметри споживання енергії, навантаження тощо. Ці дані передаються на рівень аналітики, де платформа використовує алгоритми машинного навчання та штучного інтелекту для обробки, аналізу й прогнозування. Honeywell Forge інтегрується з хмарними сервісами, що забезпечує доступ до інформації у реальному часі та дозволяє управляти процесами віддалено, підвищуючи гнучкість і масштабованість системи.

Переваги Honeywell Forge:

- моніторинг у реальному часі (платформа забезпечує доступ до даних у реальному часі, що дозволяє швидко реагувати на зміни в операційних умовах та підвищує ефективність управління);
- інтелектуальний аналіз і прогнозування (використання штучного інтелекту та машинного навчання дозволяє точно аналізувати дані та прогнозувати тенденції, що сприяє оптимізації виробничих процесів);
- віддалений доступ і управління (хмарна інтеграція дозволяє управляти операціями віддалено, що зручно для підприємств з розподіленими об'єктами та підвищує гнучкість платформи);
- скорочення експлуатаційних витрат (завдяки оптимізації процесів та моніторингу Honeywell Forge допомагає зменшити витрати на обслуговування та енергоспоживання).

Недоліки Honeywell Forge:

- висока вартість впровадження (встановлення та налаштування платформи можуть бути дорогими, що обмежує її доступність для малих і середніх підприємств);
- складність інтеграції (платформа потребує сумісності з існуючою інфраструктурою, що може вимагати додаткових ресурсів і часу для впровадження);

- залежність від інтернет-з'єднання (віддалене управління залежить від стабільного інтернет-з'єднання, що може бути обмеженням у віддалених або слабо забезпечених регіонах);
- необхідність кваліфікованого персоналу (для налаштування та обслуговування платформи потрібен досвідчений персонал, що може підвищити витрати на навчання та підтримку) [23].

Отже, Honeywell Forge є потужною платформою для оптимізації операційних і енергетичних процесів, що підвищує ефективність і знижує експлуатаційні витрати. Завдяки інтелектуальному аналізу даних, віддаленому моніторингу та прогнозуванню, Honeywell Forge дозволяє підприємствам підвищувати продуктивність і адаптивність до змінних умов. Однак, висока вартість впровадження та вимоги до інфраструктури можуть обмежити її доступність для малих підприємств, що робить цю платформу найбільш ефективною для великих організацій з високими вимогами до управління енергорозподілом.

1.6. Дослідження діяльності організації та постановка задачі на розробку комплексного рішення

ТОВ «Net City» є інтернет-сервіс-провайдером, який з 2003 року займає стабільну позицію на ринку телекомунікаційних послуг. Компанія спеціалізується на забезпеченні швидкісного доступу до інтернету для приватних і корпоративних клієнтів, пропонуючи швидкість з'єднання до 1 Гбіт/с. Вона відома своїм індивідуальним підходом до кожного абонента, що дозволяє адаптувати послуги до конкретних потреб користувачів. Завдяки компактній організаційній структурі та оптимізації бізнес-процесів, компанія ефективно функціонує навіть в умовах динамічних змін ринку (рис. 1.9).



Організаційна структура ТОВ «Net City» побудована таким чином, щоб невеликий штат спеціалістів міг ефективно забезпечувати всі ключові процеси компанії – від технічної підтримки клієнтів до розширення мережевої інфраструктури та управління фінансами. Така структура дає можливість компанії оперативно реагувати на потреби клієнтів, підтримувати високий рівень сервісу та активно розвиватися у напрямку розширення зон покриття та вдосконалення послуг [24].

Організаційна структура ТОВ «Net City» складається із:

- директора ТОВ «Net City», який відповідальний за стратегічне управління компанією, визначення напрямків розвитку та загальну координацію діяльності всіх відділів. Завдяки досвіду в телекомунікаціях та бізнес-адмініструванні, директор підтримує сучасний підхід до управління, який базується на впровадженні новітніх технологій та забезпеченні високого рівня обслуговування клієнтів;

- відділу технічної підтримки – забезпечує безперервну технічну підтримку абонентів, вирішення технічних питань, налаштування обладнання для нових підключень та обслуговування існуючих абонентів. Складається з трьох фахівців, відповідальних за оперативне реагування на запити клієнтів, віддалене та місцеве обслуговування. Відділ відзначається високим рівнем клієнтоорієнтованості, залучаючи більшість клієнтів завдяки рекомендаціям;

- відділу інженерного забезпечення та експлуатації мережі – займається монтажем, обслуговуванням і модернізацією мережевої інфраструктури, включаючи оптоволоконні лінії та вузли зв'язку, а також координує розширення мережі. До складу відділу входять чотири працівники, спеціалізовані на побудові та підтримці мережі, забезпеченні стабільності та надійності підключень. Працівники активно розширюють зону покриття, займаючись монтажем обладнання та інтеграцією нових технологій;

- відділу фінансів та адміністративного управління – відповідальний за фінансову звітність, контроль бюджетів та адміністрування, а також організацію оплати послуг постачальників, облік доходів та витрат. Відділ включає двох

працівників, бухгалтера та адміністративного менеджера, які також опікуються юридичними питаннями та документообігом. Оскільки компанія є невеликою, працівники цього відділу часто поєднують кілька функцій для оптимізації процесів;

– відділу розвитку та маркетингу – відповідає за маркетингову діяльність, залучення нових клієнтів і розробку пропозицій відповідно до ринкового попиту, а також підтримує інформаційну діяльність та аналізує ринок. У відділі працює один спеціаліст, який займається рекламною діяльністю, обробкою клієнтських запитів і аналізом тенденцій для розширення спектру послуг компанії [25].

Для створення високонадійної мережної інфраструктури з енергонезалежним функціонуванням, яке дозволить забезпечити безперебійне надання послуг зв'язку, необхідним етапом є моделювання інфраструктури з абонентським підключенням. Врахування різних сценаріїв підключення та експлуатації мережевої інфраструктури дає змогу розробити оптимальну модель, яка забезпечить стабільність та автономність роботи системи.

Основною задачею мережної інфраструктури є забезпечення надійного підключення абонентів з різними технічними потребами. У випадку оптоволоконного підключення, яке використовується ТОВ «Net City», важливо врахувати особливості розподілу мережевих ресурсів для окремих підключень. Висока пропускна здатність оптоволокна дозволяє забезпечити швидкісний інтернет-доступ для кожного клієнта, але вимагає налаштування додаткових параметрів, таких як резервування та захист каналів передачі даних, особливо в умовах відсутності стабільного живлення.

На етапі моделювання важливо створити мережеву структуру, яка дозволить ефективно використовувати ресурси для забезпечення безперебійного доступу клієнтів. Модель повинна враховувати резервне живлення, налаштування параметрів VLAN та DHCP для маршрутизаторів, а також конфігурацію VLAN на комутаторах доступу. Серед важливих аспектів моделювання є:

- налаштування VLAN та DHCP. Конфігурація цих параметрів дозволяє оптимізувати передачу даних для різних груп абонентів, мінімізуючи навантаження на центральний маршрутизатор та забезпечуючи якісне підключення;

- адресація абонентів. Моделювання передбачає використання динамічної (DHCP) та статичної адресації залежно від потреб абонентів, що дозволяє адаптувати підключення для великої кількості користувачів із різним типом обслуговування;

- інтеграція з резервними джерелами живлення. Важливим елементом моделі є забезпечення безперебійного живлення мережевих вузлів через альтернативні джерела енергії, що забезпечує стабільність роботи в умовах можливих перебоїв основного енергопостачання.

Важливо також передбачити налаштування мережевих фільтрів, таких як ACL та NAT, для захисту інфраструктури від несанкціонованого доступу та забезпечення ефективного використання IP-адрес. Контроль коректності налаштувань, перевірка конфігурацій безпеки та адресації дозволяють підтримувати стабільність та надійність підключень. Це особливо важливо для організацій, що працюють у зоні ризику регулярних перебоїв електропостачання, оскільки такі заходи підвищують надійність мережі.

З огляду на вимоги до стабільного функціонування інфраструктури, основні задачі включають:

- моделювання мережевої структури з абонентським підключенням – створення архітектури, яка ефективно розподіляє ресурси для забезпечення стабільності та енергонезалежності мережі;

- налаштування параметрів VLAN, DHCP, NAT та безпеки – оптимізація мережі для роботи з великим числом абонентів при зниженій залежності від зовнішніх енергетичних джерел;

- інтеграція систем автоматизованого моніторингу та енергорозподілу – забезпечення контролю над станом енергоспоживання у вузлах мережі для підвищення ефективності та зменшення ризику аварій;

– розробка сценаріїв резервування та безпеки для підтримки підключення – врахування можливих ризиків для створення захищеного та стабільного з'єднання для кожного абонента.

Детальне моделювання інфраструктури абонентського підключення та створення комплексного рішення для ТОВ «Net City» є стратегічно важливим для досягнення високої надійності та незалежності мережі від основних джерел живлення. Розробка такої моделі забезпечить необхідний рівень автономності та знизить витрати на обслуговування завдяки оптимізації підключень та енергозабезпечення, що є актуальним як з комерційної, так і з технічної точки зору.

Висновок до першого розділу

У рамках даного розділу досліджено та обґрунтовано інфраструктурні й технологічні рішення для створення енергонезалежної мережі з оптоволоконним підключенням. Вивчено переваги оптоволоконних мереж, зокрема їхню стабільність і енергоефективність, а також розглянуто інтеграцію альтернативних джерел живлення (сонячних панелей та вітрових турбін) для підвищення автономності. Описано рішення для зберігання та розподілу енергії, включаючи технологію PoE для стабільного функціонування підключень. Досліджено також автоматизовані системи енергорозподілу (EcoStruxure, ABB Ability, Honeywell Forge) і діяльність ТОВ «Net City» для визначення вимог до стабільної роботи мережі.

Отримані результати є підґрунтям для переходу до наступного розділу, де буде розглянуто процес моделювання інфраструктури з абонентським підключенням, налаштування мережевих параметрів та розробку сценаріїв тестування стабільності та ефективності запропонованих рішень.

2.1. Аналіз технічного завдання

Згідно із технічним завданням, мережа ТОВ «Net City» буде побудована за гнучкою моделлю, що включатиме інтеграцію технологій GPON та DWDM для надійного підключення в містах Коростень і Звягель. Будуть застосовані дві незалежні оптичні лінії зв'язку, які забезпечать безперебійність і відмовостійкість мережі, завдяки чому трафік зможе перенаправлятися у разі відмови одного з каналів.

Клієнти будуть підключені через GPON-комутатори із сегментацією на VLAN, що сприятиме безпеці та дозволить розмежувати трафік між групами користувачів. Для кожного регіону буде розроблено окремий адресний план, що полегшить управління ресурсами та сприятиме захисту даних.

Безпека мережі буде забезпечена використанням мережевого екрану ASA5505 для обмеження доступу до внутрішніх ресурсів та запобігання зовнішнім загрозам. Кожен комутатор, маршрутизатор та сервер будуть захищені паролями для обмеження несанкціонованого доступу.

Для управління трафіком буде реалізовано протокол EIGRP, що забезпечить оптимальну маршрутизацію між регіонами. Використання цього протоколу дозволить динамічно оновлювати маршрути, що підвищить стабільність з'єднання між абонентами.

Налаштування DNS та DHCP-серверів будуть виконані для спрощення управління клієнтськими підключеннями, автоматичної видачі адрес, а також для забезпечення доступу до внутрішніх ресурсів. Також буде налаштовано NAT для виходу в Інтернет з використанням єдиної зовнішньої IP-адреси на головному маршрутизаторі.

Таким чином, у результаті реалізації технічного завдання буде створено стабільну, надійну та захищену мережеву інфраструктуру, здатну забезпечити сучасні вимоги до доступу в інтернет для клієнтів у кількох регіонах.

2.2. Планування структури мережі

Логічна схема мережі інтернет-провайдера, приведена на рис. 2.1, демонструє архітектуру мережевої інфраструктури ТОВ «Net City». У системі передбачені дві основні точки доступу, розташовані в містах Коростень і Звягель. До кожної з цих точок підключено по два віддалені сегменти мережі (загалом чотири), кожен з яких має два незалежні оптичні канали, що забезпечують надійність та відмовостійкість з'єднань.

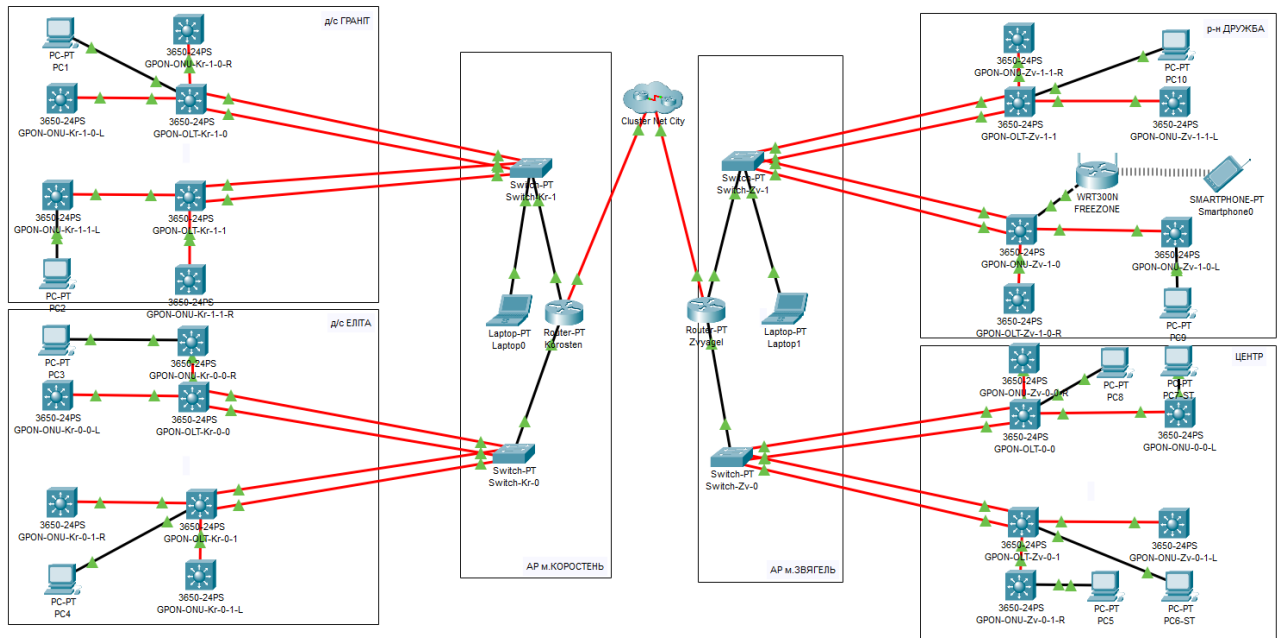


Рисунок 2.1 – Логічна схема мережі інтернет-провайдера ТОВ «Net City»

Кожен з оптичних каналів підключається до загального оптоволоконного кільця, яке обслуговується різними провайдерами на основі технології DWDM (Dense Wavelength Division Multiplexing). Це рішення дозволяє інтегрувати численні канали в єдину транспортну мережу, що значно збільшує пропускну здатність і стабільність зв'язку між точками доступу та клієнтськими сегментами. Центральний кластер, що об'єднує мережу інтернет-провайдера, включає комутатори і маршрутизатори, які керують зв'язком між регіональними вузлами, а також забезпечують підключення серверних ресурсів та мережі адміністрації.

Центральною частиною архітектури є кластер підмережі, який поєднує серверну інфраструктуру та мережу адміністрації, розташовані в місті Житомир. На рис. 2.2 видно, що до центрального маршрутизатора «Zhitomir» підключені

оптичні канали, що з'єднують його з маршрутизаторами точок доступу в містах Коростень («Router IT Korosten») та Звягель («Router IT Zvyagel»). Завдяки цьому забезпечується основний зв'язок між регіональними вузлами.

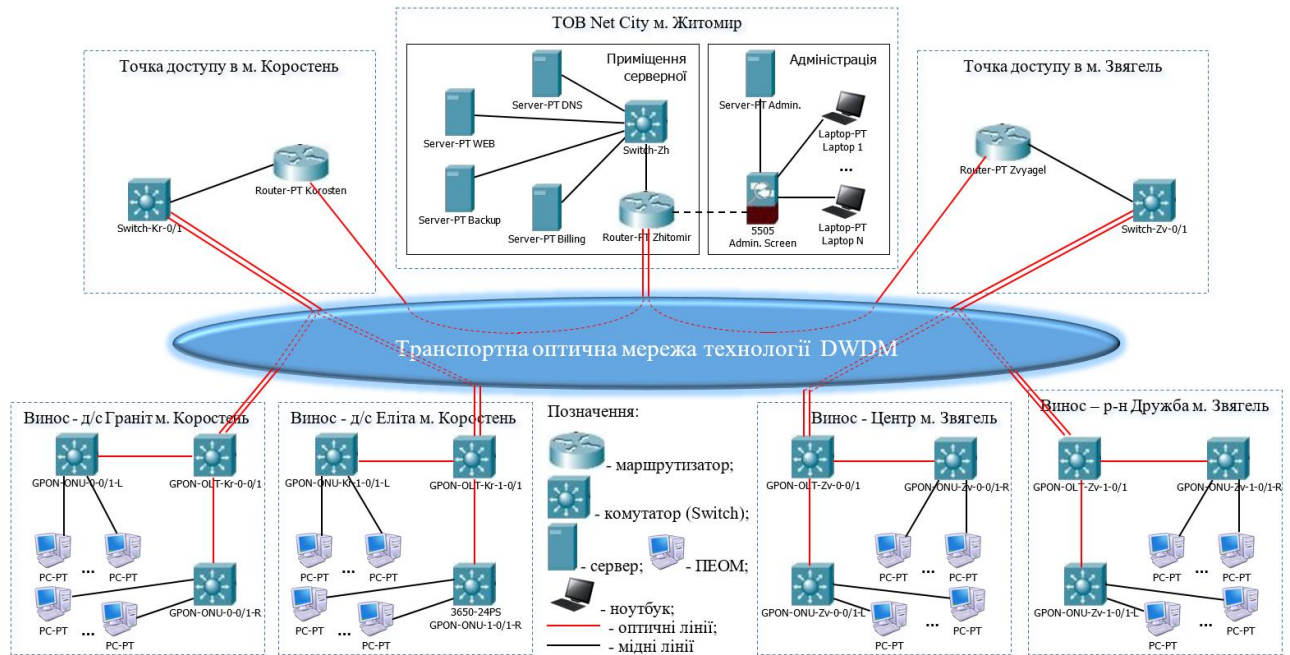


Рисунок 2.2 – Функціональна схема підключень клієнтів до DWDM

До центрального маршрутизатора також підключені сервери різного призначення: сервери DNS, WEB, Backup, а також сервер для білінгу, що забезпечують основні функції інфраструктури провайдера. У кластері також розміщено обладнання адміністрації, включно з комп'ютерами та екранами для моніторингу і управління мережею.

На рівні точок доступу в містах Коростень і Звягель передбачено по декілька віддалених сегментів, кожен з яких обладнаний комутаторами GPON OLT для підключення кінцевих абонентських пристроїв. Кожен винос містить набір клієнтських пристроїв (PC/FT), які під'єднані до мережі через термінали GPON ONU, що підключені до головних комутаторів GPON OLT.

Транспортна мережа DWDM є критичною для стабільного зв'язку, оскільки вона дозволяє використовувати декілька незалежних оптичних каналів, що об'єднані в єдине оптоволоконне кільце. Це рішення значно підвищує відмовостійкість мережі, оскільки при виникненні проблем на одному з каналів трафік може бути перенаправлений через інші, що забезпечує безперебійність сервісу для користувачів.

На рис. 2.3 приведено логічну схему організації мережевої інфраструктури, призначеної для управління, обслуговування та підтримки мережевих ресурсів компанії.

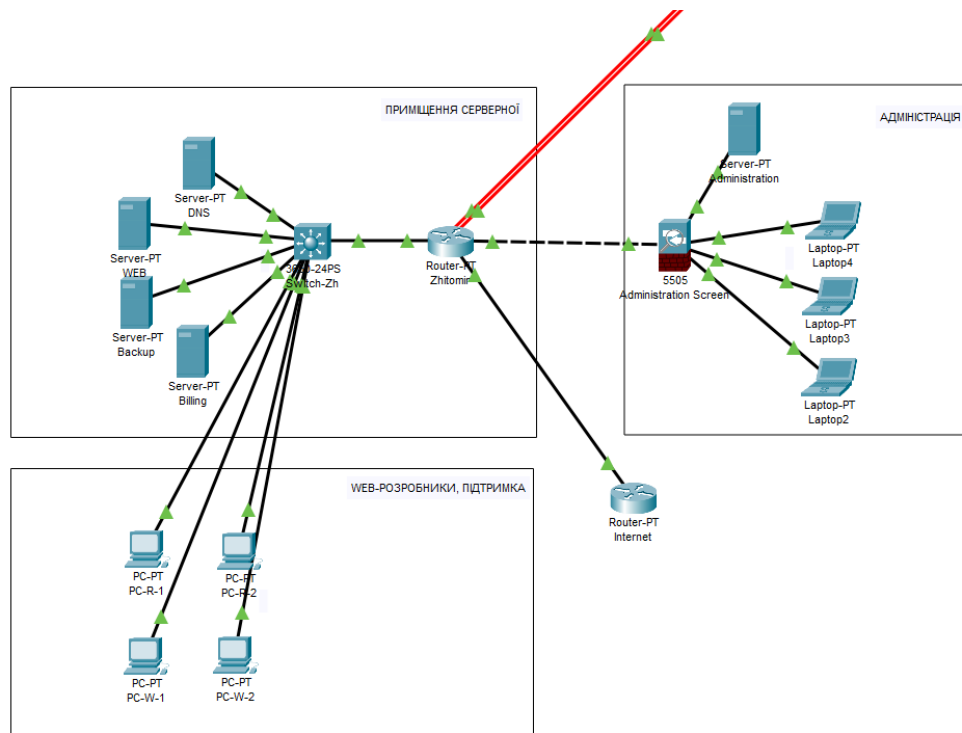


Рисунок 2.3 – Логічна схема мережі адміністрації ТОВ «Net City»

Центральним елементом цієї схеми є маршрутизатор «Router Zhitomir», до якого підключені основні мережеві компоненти, такі як сервери, комутатори, адміністративна мережа та зовнішні підключення. Через комутатор підмережі «Switch 24p S-Zh» здійснюється інтеграція кількох серверів, розміщених у приміщенні серверної.

У серверній підмережі знаходяться такі сервери:

- Server-PT DNS – для обробки та розподілу доменних імен всередині мережі;
- Server-PT WEB – використовується для розміщення веб-ресурсів компанії;
- Server-PT Backup – відповідає за резервне копіювання даних, забезпечуючи їх збереження та доступність у випадку збоїв;
- Server-PT Billing – здійснює функції білінгу, що є важливим для обліку та управління платежами клієнтів.

Мережа адміністрації підключена до центрального маршрутизатора через мережевий екран ASA5505, який забезпечує захист та обмеження доступу до внутрішніх ресурсів компанії. У мережі адміністрації розташовані робочі станції для співробітників, зокрема ноутбуки (Laptop-PT Administration, Laptop-PT Laptop1, Laptop-PT Laptop2), що підключені до внутрішньої мережі через екран ASA5505.

Для забезпечення підтримки веб-ресурсів та розробки передбачено окремий сегмент мережі, де розміщені робочі станції веб-розробників (PC-PT A, PC-PT W1, PC-PT W2). Цей сегмент мережі має доступ до центрального маршрутизатора та необхідних серверних ресурсів. Зовнішнє підключення здійснюється через окремий маршрутизатор Router-PT Internet, що забезпечує зв'язок із глобальною мережею Інтернет для всіх компонентів внутрішньої інфраструктури.

На рис. 2.4 приведені гіперканали через мережу DWDM, що використовуються для з'єднання точок доступу в мережі ТОВ «Net City».

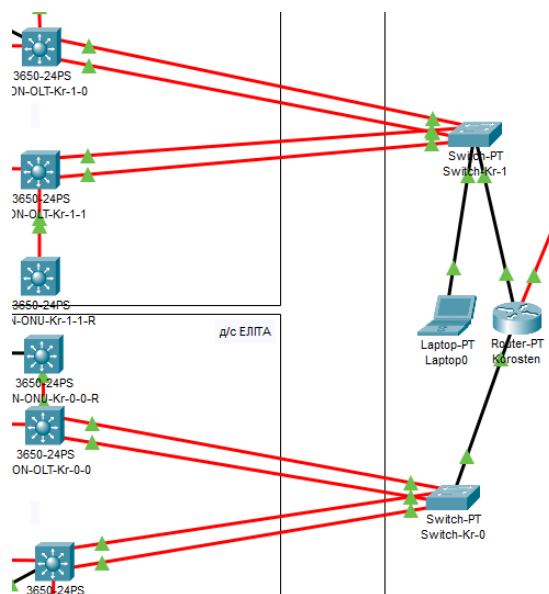


Рисунок 2.4 – Гіперканали через мережу DWDM

На рівні кожної точки доступу маршрутизатор підключається до двох комутаторів (Switch-PT Switch-Kr-1 і Switch-PT Switch-Kr-0), які реалізують гіперканали. Кожен з цих гіперканалів є окремим логічним сегментом, який представлений відокремленим VLAN. Це дозволяє розмежовувати трафік для різних абонентських підключень, забезпечуючи ізоляцію даних і підвищену безпеку.

Гіперканали, позначені червоними лініями, здійснюють зв'язок між основними комутаторами та комутаторами GPON OLT серії 3650-24PS, розташованими у віддалених сегментах. Кожен з комутаторів GPON OLT (наприклад, GPON-OLT-Kr-1-0, GPON-OLT-Kr-0-1-R) обслуговує окремі абонентські підключення, які надходять через оптичні лінії зв'язку. Таке рішення забезпечує резервування каналів, оскільки у випадку виходу з ладу одного гіперканалу трафік може бути перенаправлений через інший.

Для підтримки стабільного з'єднання всі гіперканали організовані таким чином, щоб у разі відмови одного з комутаторів або оптичних каналів з'єднання залишалося безперервним завдяки дублюванню маршрутів. Це є ключовою особливістю побудови мережі з використанням DWDM, що дозволяє забезпечувати високу пропускну здатність і надійність передачі даних між точками доступу та клієнтськими сегментами мережі.

Логічна структура гіперканалів, приведена на рис. 2.4, дозволяє не лише збільшити загальну пропускну здатність мережі, але й забезпечити гнучкість і надійність мережевих з'єднань.

На рис. 2.5 наведено останній логічний сегмент структурної мережі, який можна визначити як «віддалений сегмент». У цьому віддаленому сегменті розміщено комутатор GPON OLT, який приймає гіперканал, що забезпечує підключення абонентських пристроїв через мережу DWDM.

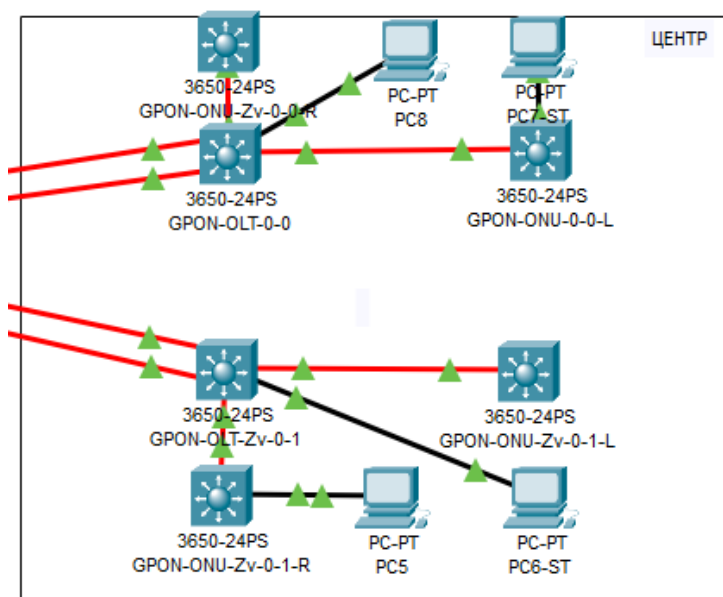


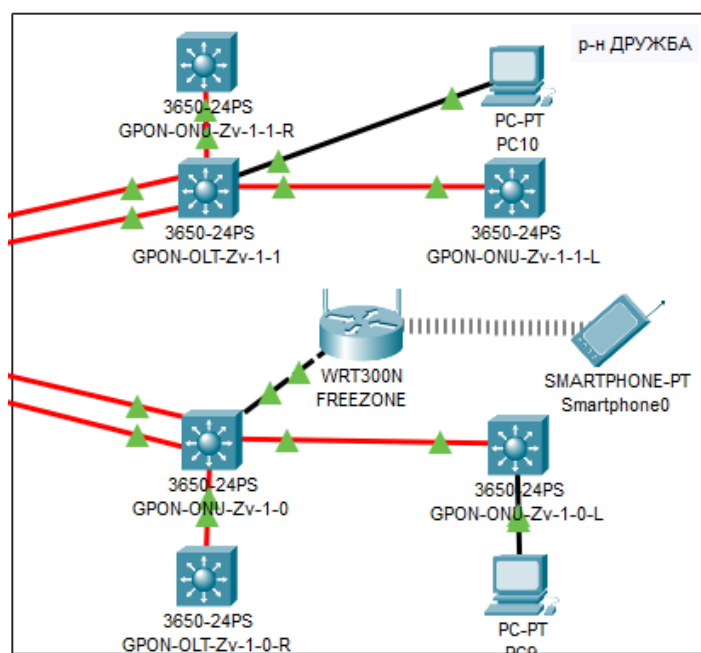
Рисунок 2.5 – Логічний сегмент структурної мережі

До кожного комутатора GPON OLT підключені клієнтські пристрої через модулі GPON ONU, що працюють як кінцеві точки для підключення абонентів. Наприклад, пристрої PC-PT PC8 та PC-PT PC8-ST під'єднані до модулів GPON-ONU-Zv-0-0 та GPON-ONU-0-0-L відповідно. Це підключення забезпечує стабільне та швидке з'єднання з центральною мережею провайдера.

Кожен GPON OLT-комутатор обслуговує групу GPON ONU, що дозволяє ефективно розподіляти пропускну здатність та забезпечувати ізоляцію трафіку для кожного клієнта. У разі потреби клієнти можуть бути підключені як безпосередньо до GPON ONU, так і через власні комутатори, що забезпечує гнучкість та можливість адаптації мережі під специфічні вимоги абонентів.

Завдяки такій структурі віддалені сегменти забезпечують масштабованість мережі та дозволяють легко підключати нових клієнтів, зберігаючи високу якість обслуговування та пропускну здатність.

На рис. 2.6 зображено логічну схему віддаленого сегмента мережі, розташованого в районі «Дружба», з додатковою точкою доступу Wi-Fi для забезпечення вільного підключення. У цьому сегменті комутатор GPON OLT (наприклад, GPON-OLT-Zv-1-1 та GPON-OLT-Zv-1-0) підключає клієнтські пристрої через модулі GPON ONU, такі як GPON-ONU-Zv-1-1-L і GPON-ONU-Zv-1-0-R. До цих модулів підключені абонентські пристрої, серед яких комп'ютери (PC-PT PC10 та PC-PT PC9) і смартфони (SMARTPHONE-PT Smartphone0), що дозволяє надавати стабільний зв'язок і доступ до мережі.



Особливістю цього віддаленого сегмента є наявність бездротової точки доступу WRT300N Freezone, яка забезпечує додаткову можливість для підключення клієнтів через Wi-Fi. Ця точка доступу інтегрована в мережу шляхом підключення до комутатора GPON OLT, що дозволяє покрити певну зону вільним доступом до інтернету для користувачів з мобільними пристроями. Це рішення забезпечує гнучкість підключень та підвищує зручність використання мережі для кінцевих користувачів у районі «Дружба».

Дана схема дозволяє надавати послуги широкому колу користувачів, зокрема через GPON ONU для фіксованих підключень та Wi-Fi для бездротового доступу, що відповідає потребам сучасних користувачів у стабільному і гнучкому інтернет-зв'язку.

Одним із важливих аспектів у схемі є можливість підключення клієнта до оптичної GPON-мережі без використання додаткового обладнання GPON ONU. Такий підхід реалізовано у виносі «Еліта», де клієнтське обладнання інтегрується безпосередньо в GPON-мережу за допомогою мережевої карти, оснащеної SFP+ модулем, що підтримує технологію XPON. (рис. 2.7)

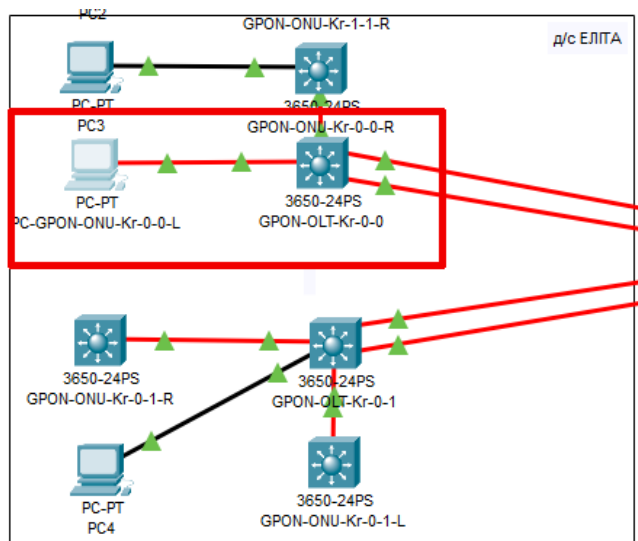


Рисунок 2.7 – Логічна схема виносів з підключенням клієнта без окремого обладнання GPON ONU

Дана схема є економічно вигідною, оскільки усуває потребу у використанні додаткового клієнтського пристрою, такого як GPON ONU. Це також зменшує затрати на обслуговування та підвищує швидкість передачі даних

між кінцевим пристроєм та мережею. Клієнтський пристрій (PC3) підключено безпосередньо до GPON OLT за допомогою порту SFP+, що дозволяє забезпечити високу пропускну здатність і мінімізувати затримки у передачі даних.

Особливістю такого підключення є можливість використання окремих VLAN для ізоляції трафіку. Це забезпечує високу безпеку переданих даних і підтримку різних рівнів доступу для клієнтів. У випадку виходу з ладу окремих компонентів передбачено резервування каналів, яке забезпечує стабільність роботи мережі.

2.3. Конфігурація та налаштування роботи проекту мережі

Для організації мережевої інфраструктури інтернет-провайдера передбачено детальну схему адресації, яка забезпечує ефективний розподіл IP-адрес для кожного сегмента мережі, включаючи центральний маршрутизатор, серверну інфраструктуру, точки доступу та адміністративну мережу (табл. 2.1).

Таблиця 2.1

Схема адресації мережі

Ділянка	Мережа
Zhitomir-Korosten	192.168.1.0/24
клієнтські VLAN	10.0.0.0/20 (4096 адрес, планується 1000 абонентів GPON)
Zhitomir-Zvyagel	192.168.2.0/24
Клієнтські VLAN	10.0.16.0/20 (4096 адрес, планується 2000 абонентів GPON)
Wi-Fi (вільний доступ)	10.0.32.0/24
Zhitomir-серверна, програмісти	192.168.7.0/24
DNS-сервер	192.168.7.2
WEB-сервер	192.168.7.5
Zhitomir-адміністрація	192.168.8.0/24
Zhitomir-Internet	88.88.88.0/30

Для кожного клієнта, підключеного через GPON OLT, виділено окремий VLAN, що дозволяє забезпечити сегментацію трафіку та підвищити рівень безпеки в мережі. Відповідно до адресного плану, у табл. 2.2 наведено схему розподілу IP-адрес для VLAN на маршрутизаторах точок доступу в місті Коростень. Кожен VLAN отримує власний діапазон адрес, які розподіляються через відповідний комутатор GPON OLT, налаштований у динамічному режимі для гнучкого управління підключеннями.

Таблиця 2.2

Розподіл адресації на AP Коростень

VLAN	Адресація	Тип (комутатор)
10	10.0.0.0/22	Динамічна* (GPON-OLT-Kr-0-1)
11	10.0.4.0/22	Динамічна* (GPON-OLT-Kr-0-0)
12	10.0.8.0/22	Динамічна* (GPON-OLT-Kr-1-0)
13	10.0.12.0/22	Динамічна* (GPON-OLT-Kr-1-1)

Для організації мережі в точці доступу AP Звягель також передбачено розподіл клієнтів за окремими VLAN, кожен з яких отримує власний діапазон IP-адрес (табл. 2.3). Це забезпечує можливість гнучкого управління підключеннями та підвищує безпеку, ізолюючи трафік кожного клієнта. У даному випадку комутатори GPON OLT на точці доступу Звягель працюють у динамічному режимі, окрім VLAN для Wi-Fi, де використовується статична адресація для підтримки стабільності бездротового підключення.

Таблиця 2.3

Розподіл адресації на AP Звягель

VLAN	Адресація	Тип (комутатор)
10	10.0.16.0/22	Динамічна* (GPON-OLT-Zv-0-1)
11	10.0.20.0/22	Динамічна* (GPON-OLT-Zv-0-0)
12	10.0.24.0/22	Динамічна* (GPON-OLT-Zv-1-0)
13	10.0.28.0/22	Динамічна* (GPON-OLT-Zv-1-1)
15 (Wi-Fi)	10.0.32.0/24	Статична* (GPON-OLT-Zv-1-0)

Для ефективного управління мережею передбачено резервування певних адрес для ключових мережевих ресурсів (табл. 2.4).

Резервування адрес для управління мережею

Ресурс	Адреса з підмережі	Примітка
Адреса VLAN-інтерфейсу на маршрутизаторі	.1/22 (для 15 VLAN .1/24)	Шлюз за замовчуванням для клієнтів VLAN
Комутатор точки доступу	.2/22	Для віддаленого налаштування комутатора точки доступу
Management-адреса GPON OLT	.3/22	Для віддаленого налаштування OLT-комутатора віддаленого сегмента
Клієнти локального свіча GPON OLT	.4/22, .5/22	На схемі ці адреси будуть встановлені на комутатори, що емулюють GPON ONU

Резервування адрес дозволить адмініструвати мережеві пристрої з віддаленим доступом та забезпечує стабільне функціонування інфраструктури. Адреси виділено для VLAN-інтерфейсів маршрутизаторів, комутаторів, управлінських адрес GPON OLT та клієнтських пристроїв. Такий розподіл адрес дозволяє мінімізувати можливі конфлікти в мережі та забезпечити надійність адміністрування.

2.3.1. Налаштування основних параметрів та безпеки пристроїв

Для забезпечення безпеки та контролю доступу до кожного елемента мережі передбачено налаштування паролів користувача для входу, а також паролів для режиму конфігурації. Це дозволяє захистити мережеве обладнання від несанкціонованого доступу та забезпечує зручне адміністрування. На прикладі маршрутизатора «Router» виконується налаштування паролів на всі консольні та термінальні лінії, встановлення паролю в режимі «enable», а також задається ім'я хоста для зручності ідентифікації пристрою (рис. 2.8).

```

Router#conf term
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#line con 0
Router(config-line)#password 1
Router(config-line)#line aux 0
Router(config-line)#password 1
Router(config-line)#line vty 0 4
Router(config-line)#pass 1
Router(config-line)#exit
Router(config)#ena password 1
Router(config)#hostname Router1
Router1(config)#exi
Router1#
%SYS-5-CONFIG_I: Configured from console by console

Router1#copy runn start
Destination filename [startup-config]?
Building configuration...
[OK]
Router1#

```

Рисунок 2.8 – Встановлення паролів користувачів

Після завершення налаштувань маршрутизатор буде мати встановлені паролі для захисту всіх консольних, допоміжних та термінальних ліній, а також визначене ім'я хоста для зручної ідентифікації. Цей процес є важливим для безпеки мережі та забезпечує захищений доступ до конфігураційного режиму пристрою.

2.3.2. Конфігурація VLAN та DHCP маршрутизаторів

Для забезпечення коректної роботи мережевої інфраструктури, на центральному маршрутизаторі «Zhitomir» виконано налаштування інтерфейсів. Це дозволяє маршрутизатору обробляти трафік між різними сегментами мережі, забезпечуючи стабільне з'єднання з іншими мережевими елементами. На рис. 2.9 приведено конфігурацію кожного інтерфейсу з відповідними IP-адресами та параметрами швидкості та дуплексу.

```

!
interface GigabitEthernet0/0
ip address 88.88.88.2 255.255.255.252
duplex auto
speed auto
!
interface GigabitEthernet1/0
ip address 192.168.7.1 255.255.255.0
duplex auto
speed auto
!
interface GigabitEthernet2/0
ip address 192.168.1.1 255.255.255.0
!
interface GigabitEthernet3/0
ip address 192.168.2.1 255.255.255.0
!
...
!
interface GigabitEthernet9/0
ip address 192.168.8.1 255.255.255.0
duplex auto
speed auto
!

```

Рисунок 2.9 – Конфігурація інтерфейсу IP-адрес

Кожен інтерфейс отримує унікальну IP-адресу, що дозволяє маршрутизатору «Zhitomir» комунікувати з іншими сегментами мережі. Наприклад, інтерфейс GigabitEthernet0/0 налаштований для виходу в Інтернет через адресу 88.88.88.2 з маскою 255.255.255.252, що відповідає вимогам безпеки та стабільності підключення. Інші інтерфейси обслуговують внутрішні підмережі, такі як серверна, підмережа адміністрації, а також з'єднання з точками доступу у Коростені та Звягелі.

Дана конфігурація дозволить забезпечити ефективну маршрутизацію, гнучкість у налаштуванні та надійність з'єднань для підтримки безперебійної роботи мережевих послуг провайдера.

На маршрутизаторі «Korosten» виконано конфігурацію інтерфейсів, VLAN та налаштування DHCP для забезпечення автоматичної видачі IP-адрес клієнтам та ефективного управління трафіком у різних VLAN. На рис. 2.10 наведено лістинг налаштувань, що охоплює виключення адрес для DHCP, створення DHCP-пулів для кожного VLAN, налаштування IP-адрес інтерфейсів, а також конфігурацію encapsulation для відповідних VLAN.

```

!
ip dhcp excluded-address 10.0.0.1 10.0.0.5
ip dhcp excluded-address 10.0.4.1 10.0.4.5
ip dhcp excluded-address 10.0.8.1 10.0.8.5
ip dhcp excluded-address 10.0.12.1 10.0.12.5
!
ip dhcp pool p0
network 10.0.0.0 255.255.252.0
default-router 10.0.0.1
ip dhcp pool p1
network 10.0.4.0 255.255.252.0
default-router 10.0.4.1
ip dhcp pool p2
network 10.0.8.0 255.255.252.0
default-router 10.0.8.1
ip dhcp pool p3
network 10.0.12.0 255.255.252.0
default-router 10.0.12.1
!
ip name-server 192.168.7.2
!
interface GigabitEthernet0/0
no ip address
duplex auto
speed auto
!
interface GigabitEthernet0/0.10
encapsulation dot1Q 10
ip address 10.0.0.1 255.255.252.0
!
interface GigabitEthernet0/0.11
encapsulation dot1Q 11
ip address 10.0.4.1 255.255.252.0
!
interface GigabitEthernet1/0
no ip address
duplex auto
speed auto
!
interface GigabitEthernet1/0.12
encapsulation dot1Q 12
ip address 10.0.8.1 255.255.252.0
!
interface GigabitEthernet1/0.13
encapsulation dot1Q 13
ip address 10.0.12.1 255.255.252.0
!
interface GigabitEthernet2/0
ip address 192.168.1.2 255.255.255.0
!

```

Рисунок 2.10 – Налаштування інтерфейсів VLAN та DHCP маршрутизатора Korosten

Налаштування маршрутизатора «Korosten» забезпечує розподіл IP-адрес серед клієнтів через DHCP-пули, а також дає можливість створити ізольовані сегменти для кожного VLAN. Це рішення підвищує стабільність та безпеку мережі, забезпечуючи надійне підключення для кожного клієнта.

Також виконано конфігурацію інтерфейсів на маршрутизаторі «Zvyagel», VLAN та налаштування DHCP для автоматичної видачі IP-адрес клієнтам і керування трафіком у різних VLAN (рис. 2.11).

```

!
ip dhcp excluded-address 10.0.16.1 10.0.16.5
ip dhcp excluded-address 10.0.20.1 10.0.20.5
ip dhcp excluded-address 10.0.24.1 10.0.24.5
ip dhcp excluded-address 10.0.28.1 10.0.28.5
!
ip dhcp pool p4
network 10.0.16.0 255.255.252.0
default-router 10.0.16.1
ip dhcp pool p5
network 10.0.20.0 255.255.252.0
default-router 10.0.20.1
ip dhcp pool p6
network 10.0.24.0 255.255.252.0
default-router 10.0.24.1
ip dhcp pool p7
network 10.0.28.0 255.255.252.0
default-router 10.0.28.1
!
ip name-server 192.168.7.2
!
interface GigabitEthernet0/0
no ip address
duplex auto
speed auto
!
interface GigabitEthernet0/0.10
encapsulation dot1Q 10
ip address 10.0.16.1 255.255.252.0
!
interface GigabitEthernet0/0.11
encapsulation dot1Q 11
ip address 10.0.20.1 255.255.252.0
!
interface GigabitEthernet1/0
no ip address
duplex auto
speed auto
!
interface GigabitEthernet1/0.12
encapsulation dot1Q 12
ip address 10.0.24.1 255.255.252.0
!
interface GigabitEthernet1/0.13
encapsulation dot1Q 13
ip address 10.0.28.1 255.255.252.0
!
interface GigabitEthernet1/0.15
encapsulation dot1Q 15
ip address 10.0.32.1 255.255.255.0
!
interface GigabitEthernet2/0
ip address 192.168.2.2 255.255.255.0
!

```

Рисунок 2.11 – Налаштування інтерфейсів VLAN та DHCP маршрутизатора Zvyagel

Дана конфігурація маршрутизатора «Zvyagel» дозволяє забезпечити надійний розподіл IP-адрес для клієнтів у різних VLAN, що сприяє безпеці та ефективному керуванню мережею. DHCP-сервери на маршрутизаторі забезпечують автоматичний розподіл адрес, а VLAN сегментують трафік для кожної групи клієнтів, що підвищує стабільність і керованість мережі.

2.3.3. Конфігурація VLAN на комутаторах точок доступу

Для налаштування комутатора точки доступу «Switch-Kr-1» виконано конфігурацію гіперканалів і VLAN для забезпечення надійного та ефективного з'єднання між різними компонентами мережі (рис. 2.12). На цьому комутаторі створено дві групи інтерфейсів, кожна з яких обслуговує окремий гіперканал через оптичні канали DWDM. Групи інтерфейсів переведені в режим транку для підтримки VLAN, що дозволяє гнучко керувати трафіком і підтримувати високу пропускну здатність.

```

!
interface Port-channel1
switchport trunk allowed vlan 12
switchport mode trunk
!
interface Port-channel2
switchport trunk allowed vlan 13
switchport mode trunk
!
interface GigabitEthernet0/1
switchport trunk allowed vlan 12-13
switchport mode trunk
!
interface GigabitEthernet1/1
switchport access vlan 13
switchport mode access
!
interface GigabitEthernet2/1
switchport trunk allowed vlan 12
switchport mode trunk
channel-group 1 mode active
!
interface GigabitEthernet3/1
switchport trunk allowed vlan 12
switchport mode trunk
channel-group 1 mode active
!
interface GigabitEthernet4/1
switchport trunk allowed vlan 13
switchport mode trunk
channel-group 2 mode active
!
interface GigabitEthernet5/1
switchport trunk allowed vlan 13
switchport mode trunk
channel-group 2 mode active
!
...
!
interface Vlan1
no ip address
shutdown
!
interface Vlan12
ip address 10.0.8.2 255.255.252.0
!
interface Vlan13
ip address 10.0.12.2 255.255.252.0
!

```

Рисунок 2.12 – Налаштування гіперканалу та VLAN комутатора точки доступу

Ця конфігурація забезпечить стабільну роботу гіперканалів, які обслуговують VLAN 12 та 13, а також дозволяє надійно керувати трафіком між основними сегментами мережі. Налаштування VLAN для кожного інтерфейсу дозволяє ефективно розподіляти трафік і гарантує високу пропускну здатність, що є важливим для оптимальної роботи мережевих сервісів.

2.3.4. Налаштування гіперканалу та VLAN комутатора виносу

Для побудови мережі було обрано обладнання ZYXEL GPON OLT 1404A та GPON OLT 1408A, які підтримують технологію LACP для створення гіперканалу, що підвищує надійність і пропускну здатність зв'язку. Для емуляції налаштувань даних пристроїв на комутаторі Cisco 3650 налаштовано інтерфейси на прикладі вузла GPON-OLT-Kr-1-0 (рис. 2.13).

```

!
interface Port-channel1
 switchport trunk allowed vlan 12
 switchport mode trunk
!
interface GigabitEthernet1/0/1
 switchport access vlan 12
 switchport mode access
 switchport nonegotiate
!
...
!
interface GigabitEthernet1/0/24
 switchport access vlan 12
 switchport mode access
 switchport nonegotiate
!

!
interface GigabitEthernet1/1/1
 switchport trunk allowed vlan 12
 switchport mode trunk
 channel-group 1 mode passive
!
interface GigabitEthernet1/1/2
 switchport trunk allowed vlan 12
 switchport mode trunk
 channel-group 1 mode passive
!
...
!
interface Vlan12
 mac-address 00e0.f9b7.9b01
 ip address 10.0.8.3 255.255.252.0
!

```

Рисунок 2.13 – Налаштування гіперканалу та VLAN комутатора виносу

На основі конфігурації комутатора Switch-Kr-1 була виконана перевірка стану гіперканалів, налаштованих для забезпечення високошвидкісного з'єднання з віддаленим сегментом мережі (д/с «Граніт»). Рис. 2.14 ілюструє стан створеного гіперканалу Port-channel1, який об'єднує інтерфейси Gig1/0/1 та Gig1/0/2 і забезпечує загальну пропускну здатність до 2000 Мбіт/с.

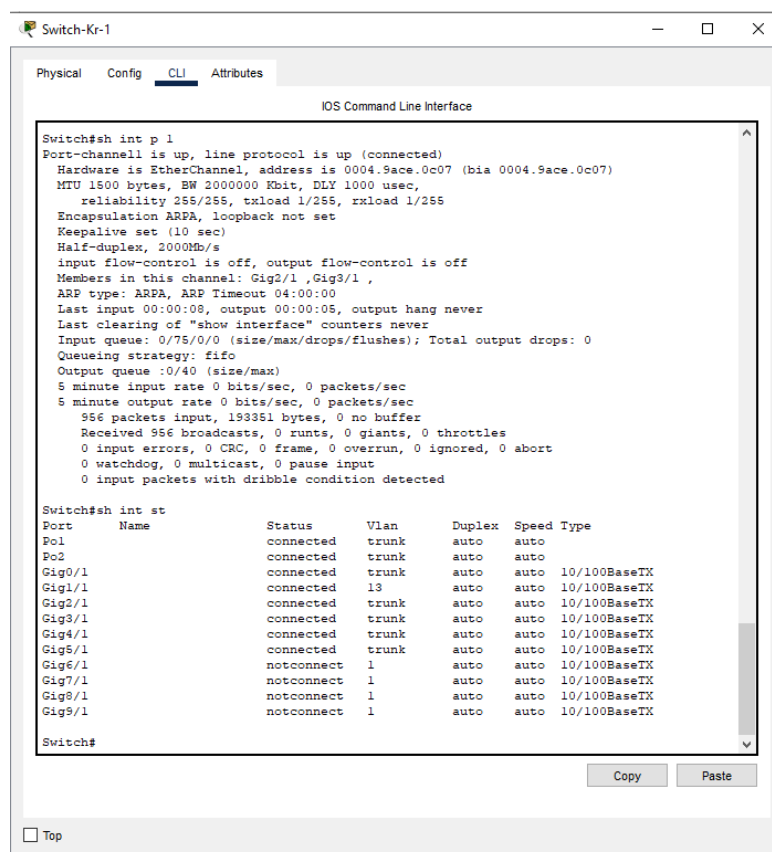


Рисунок 2.14 – Гіперканали на винос д/с Граніт у роботі (2000Мбіт/с)

Перевірка підтверджує, що гіперканал Port-channel1 успішно об'єднує інтерфейси Gig1/0/1 і Gig1/0/2, забезпечуючи високу пропускну здатність у 2000 Мбіт/с для передачі трафіку VLAN 12. Це рішення оптимізує мережеву інфраструктуру та підвищує стійкість з'єднання, забезпечуючи надійний зв'язок між комутатором і віддаленими сегментами мережі.

Для перевірки доступності management-інтерфейсів усіх GPON OLT з маршрутизатора Korosten було виконано команду ping до IP-адрес, призначених для управління кожним GPON-OLT. Результати наведено на рис. 2.15, який підтверджує доступність усіх інтерфейсів для віддаленого управління.

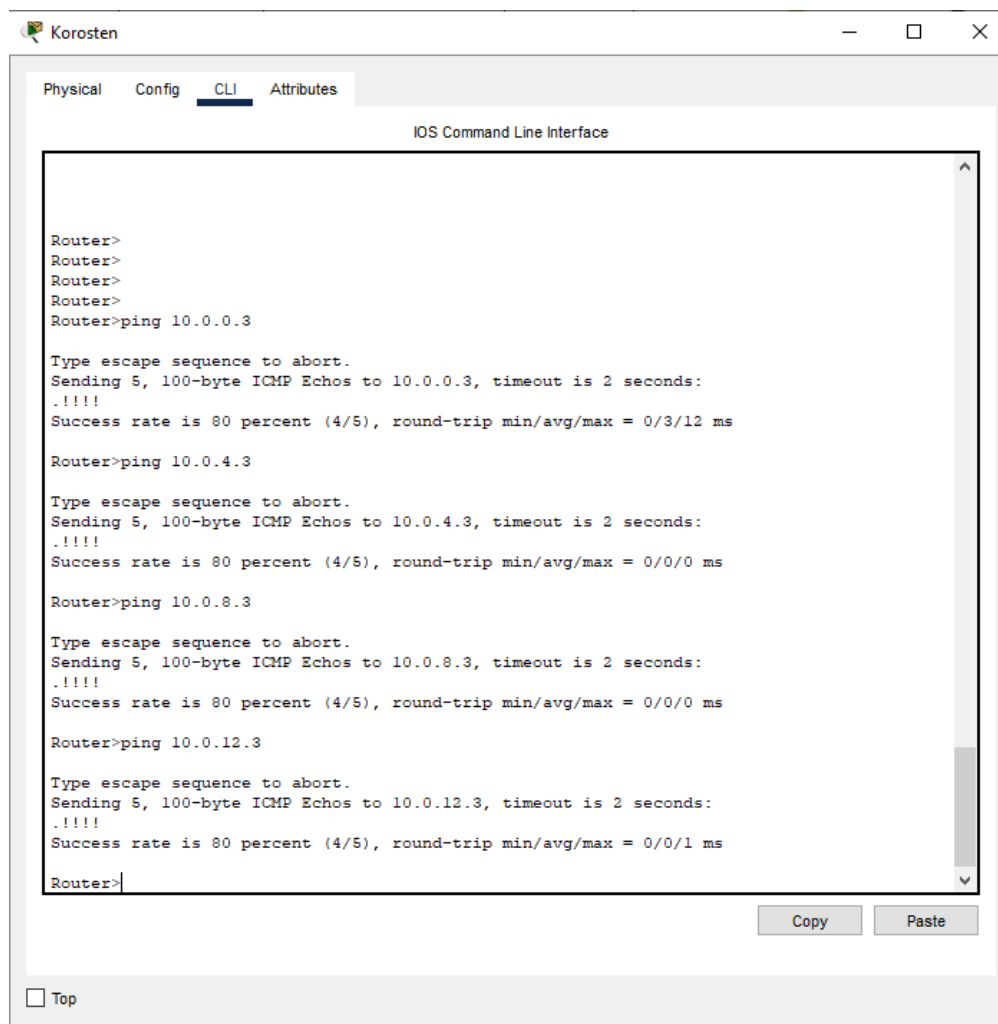


Рисунок 2.15 – Management-інтерфейси доступні для управління

Виконані пінги підтверджують доступність усіх management-інтерфейсів GPON-OLT для віддаленого управління з маршрутизатора Korosten. Це дозволяє забезпечити надійний контроль і моніторинг кожного пристрою GPON-OLT через призначені IP-адреси. Високий відсоток успішних запитів (80%) та низький

час відповіді свідчать про стабільне мережеве підключення і готовність інфраструктури до віддаленого адміністрування.

2.3.5. Налаштування адресації хостів клієнтів мережі (DHCP)

На основі проведених налаштувань було виконано налаштування IP-адресації для клієнтських пристроїв у віддаленому сегменті, прикладом чого є клієнтський комп'ютер PC1 (рис. 2.16). Налаштування виконувались через DHCP, що дозволило автоматично присвоїти IP-адресу та інші параметри мережі.

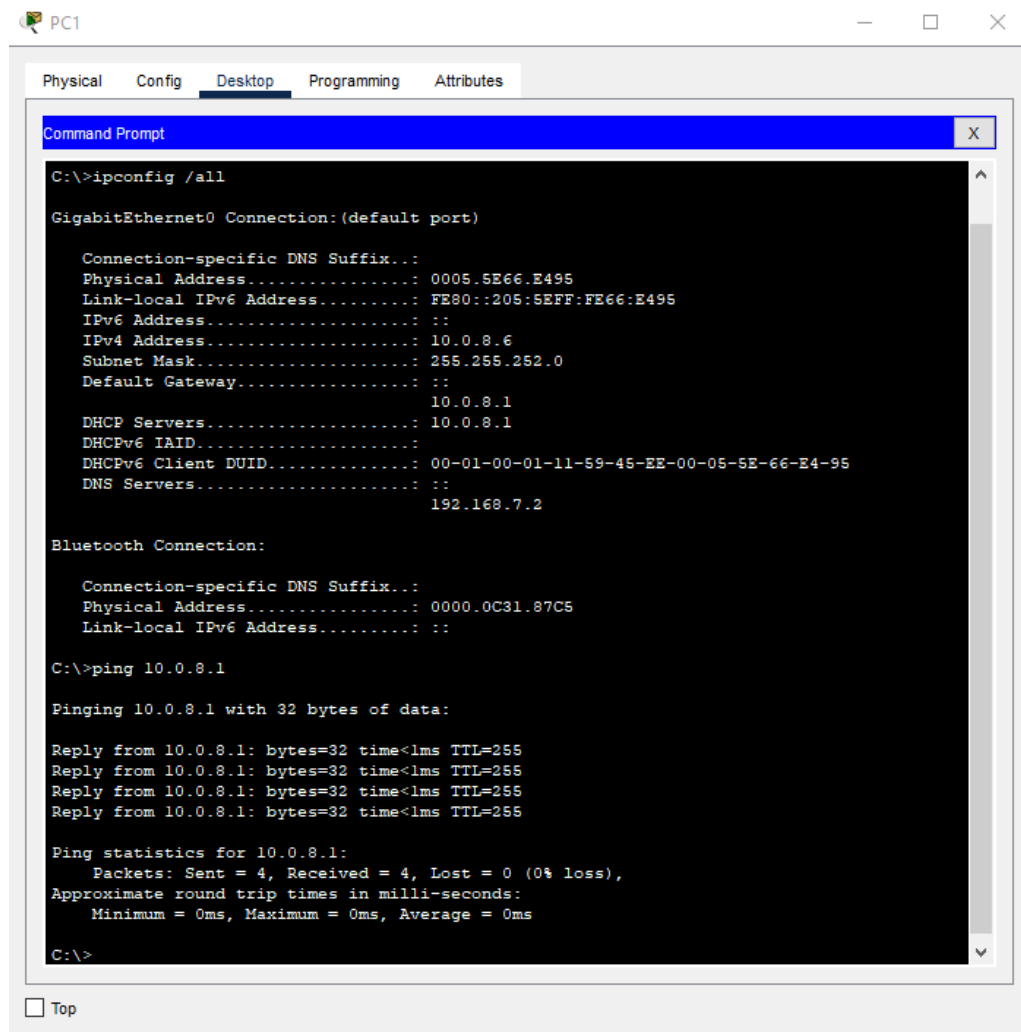
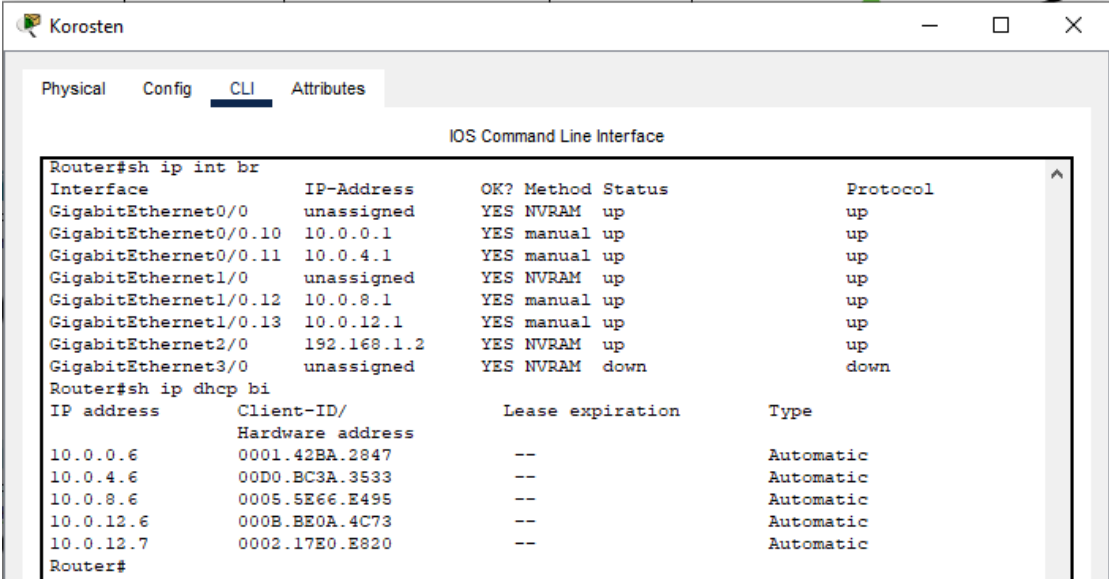


Рисунок 2.16 – Налаштування по DHCP

Як видно із рис. 2.15 PC1, DHCP-сервер коректно присвоїв IP-адресу, маску підмережі, адресу шлюзу та DNS-сервера. Команда ping підтвердила стабільне з'єднання з шлюзом, що забезпечує коректний доступ до мережі. Даний результат свідчить про правильність налаштувань DHCP та VLAN, а також про

успішне функціонування мережевої інфраструктури в частині автоматичної адресації та маршрутизації для віддалених клієнтів.

На основі конфігураційних налаштувань маршрутизатора «Korosten» було перевірено стан інтерфейсів і DHCP-серверів для кожного VLAN, підключеного до даного маршрутизатора. Рис. 2.17 демонструє інформацію про активні інтерфейси та присвоєні IP-адреси для клієнтів.



The screenshot shows the Korosten router's CLI interface. The top bar has tabs for Physical, Config, CLI (selected), and Attributes. The main window displays the output of two commands: 'show ip int br' and 'show ip dhcp bi'.

IOS Command Line Interface

Router#sh ip int br

Interface	IP-Address	OK?	Method	Status	Protocol
GigabitEthernet0/0	unassigned	YES	NVRAM	up	up
GigabitEthernet0/0.10	10.0.0.1	YES	manual	up	up
GigabitEthernet0/0.11	10.0.4.1	YES	manual	up	up
GigabitEthernet1/0	unassigned	YES	NVRAM	up	up
GigabitEthernet1/0.12	10.0.8.1	YES	manual	up	up
GigabitEthernet1/0.13	10.0.12.1	YES	manual	up	up
GigabitEthernet2/0	192.168.1.2	YES	NVRAM	up	up
GigabitEthernet3/0	unassigned	YES	NVRAM	down	down

Router#sh ip dhcp bi

IP address	Client-ID/ Hardware address	Lease expiration	Type
10.0.0.6	0001.42BA.2847	--	Automatic
10.0.4.6	00D0.BC3A.3533	--	Automatic
10.0.8.6	0005.5E66.E495	--	Automatic
10.0.12.6	000B.BE0A.4C73	--	Automatic
10.0.12.7	0002.17E0.E820	--	Automatic

Рисунок 2.17 – Налаштовані клієнти усіх під'єднаних VLAN

Отже, маршрутизатор «Korosten» успішно обслуговує DHCP-запити для кожного VLAN, забезпечуючи клієнтів автоматичними IP-адресами відповідно до налаштованих підмереж. Кожному клієнту виділено унікальну IP-адресу, яка відповідає сегменту мережі, до якого він належить. Стан інтерфейсів показує, що всі основні підключення активні (режим up), за винятком інтерфейсу GigabitEthernet3/0, який наразі не використовується.

Це підтверджує правильність налаштувань DHCP-сервера та забезпечує стабільне функціонування мережі, дозволяючи клієнтам автоматично отримувати IP-адреси та забезпечуючи ефективне керування мережевими ресурсами.

2.3.6. Налаштування адресації Wi-Fi вільної точки доступу

Для забезпечення коректної роботи вільної точки доступу у віддаленому сегменті мережі (р-н Дружба), відповідно до табл. 2.4, було налаштовано

статичну IP-адресацію у VLAN 15. На рис. 2.18 зображено параметри конфігурації цієї точки доступу.

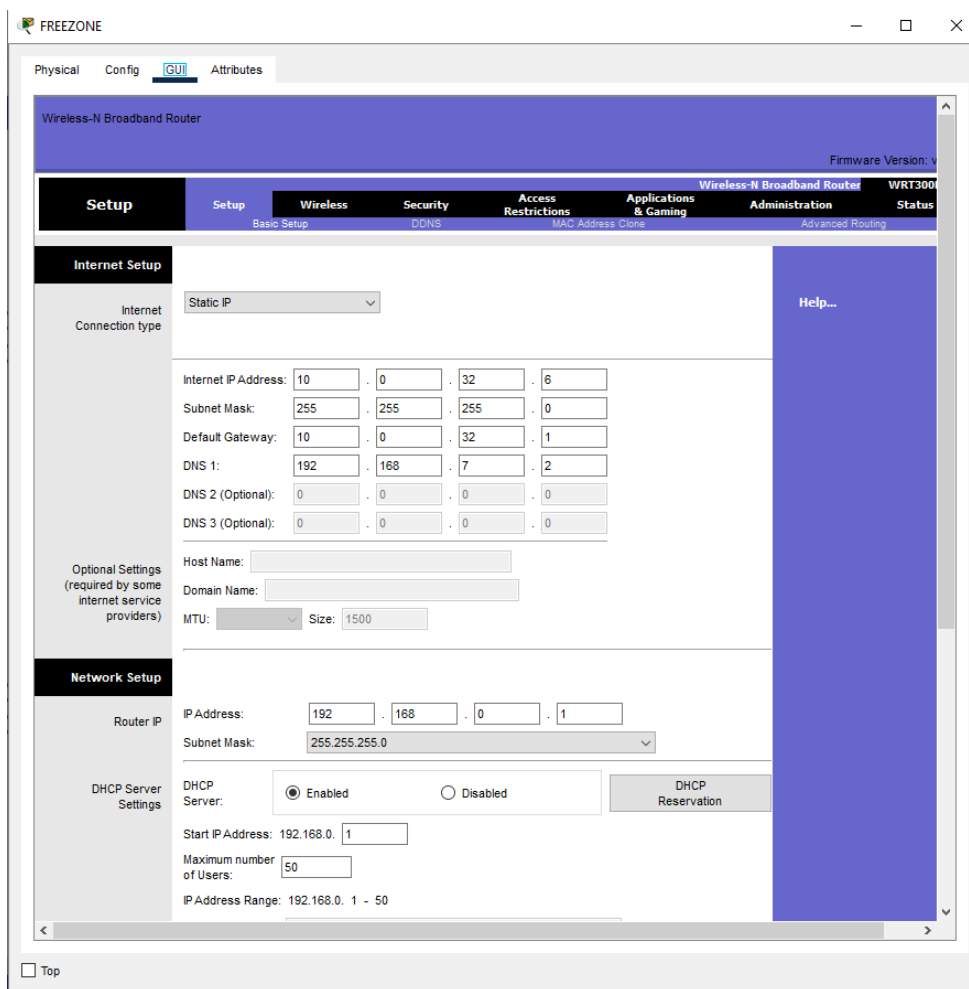


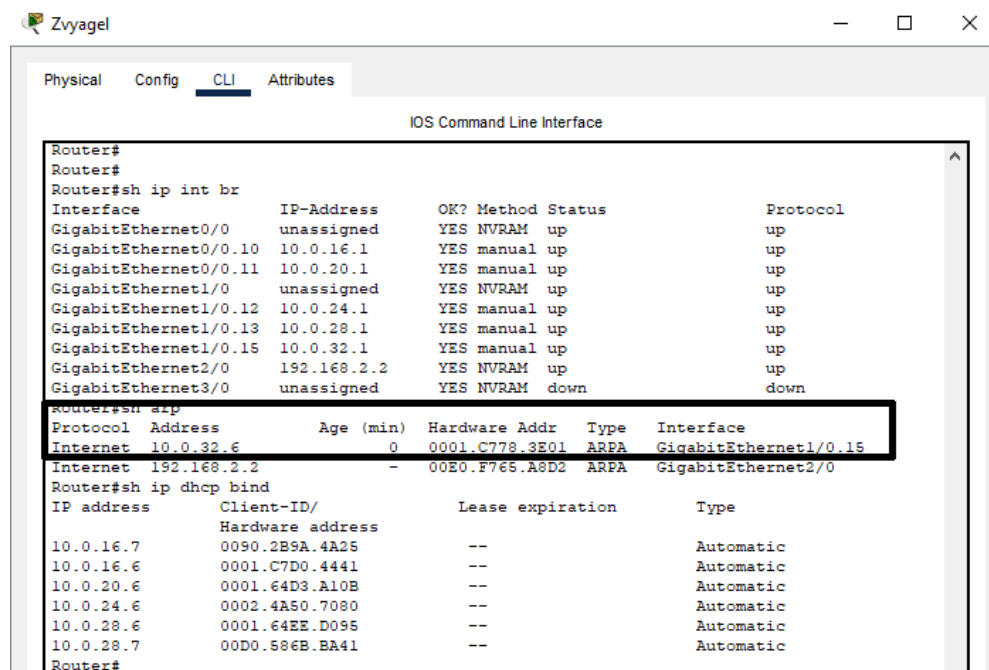
Рисунок 2.18 – Налаштування вільної точки доступу

Конфігурація точки доступу у VLAN 15 з використанням статичної IP-адреси 10.0.32.3 дозволяє забезпечити стабільне підключення до мережі та гарантує доступність інтернет-з'єднання для користувачів у зоні Wi-Fi. Налаштування DHCP з діапазоном 192.168.8.1 - 192.168.8.50 дозволяє автоматично надавати IP-адреси підключеним пристроям, що забезпечує зручність користування та ефективне управління підключеннями. Це налаштування також відповідає вимогам для вільної точки доступу, надаючи клієнтам легкий доступ до інтернету у віддаленому сегменті р-н Дружба.

2.3.7. Контроль коректності налаштувань

Для перевірки доступності всіх VLAN, клієнтів та точки доступу з маршрутизатора Zvyagel було здійснено огляд налаштувань IP-інтерфейсів та

перевірку DHCP-виданих адрес. Результати, наведені на рис. 2.19, підтверджують коректність налаштувань та забезпечують стабільне з'єднання у всіх налаштованих VLAN.



```
Router#
Router#
Router#sh ip int br
Interface                IP-Address      OK? Method Status      Protocol
GigabitEthernet0/0       unassigned      YES NVRAM  up          up
GigabitEthernet0/0.10    10.0.16.1       YES manual  up          up
GigabitEthernet0/0.11    10.0.20.1       YES manual  up          up
GigabitEthernet1/0       unassigned      YES NVRAM  up          up
GigabitEthernet1/0.12    10.0.24.1       YES manual  up          up
GigabitEthernet1/0.13    10.0.28.1       YES manual  up          up
GigabitEthernet1/0.15    10.0.32.1       YES manual  up          up
GigabitEthernet2/0       192.168.2.2     YES NVRAM  up          up
GigabitEthernet3/0       unassigned      YES NVRAM  down        down

Router#sh arp
Protocol Address          Age (min)  Hardware Addr  Type   Interface
Internet  10.0.32.6          0          0001.C778.3E01  ARPA   GigabitEthernet1/0.15
Internet  192.168.2.2        -          00E0.F765.A8D2  ARPA   GigabitEthernet2/0

Router#sh ip dhcp bind
IP address      Client-ID/
                Hardware address
                Lease expiration
                Type
10.0.16.7       0090.2B9A.4A25   --           Automatic
10.0.16.6       0001.C7D0.4441   --           Automatic
10.0.20.6       0001.64D3.A10B   --           Automatic
10.0.24.6       0002.4A50.7080   --           Automatic
10.0.28.6       0001.64EE.D095   --           Automatic
10.0.28.7       00D0.586B.BA41   --           Automatic
Router#
```

Рисунок 2.19 – Перевірка коректності налаштування

Огляд конфігурації інтерфейсів та виданих DHCP-адрес підтверджує доступність і правильність налаштувань для кожного VLAN, клієнтських підключень та точки доступу. Кожен інтерфейс перебуває у стані up і обслуговує відповідний VLAN, забезпечуючи стабільне з'єднання і доступ до мережі для клієнтів. Це свідчить про успішне налаштування мережевої інфраструктури та готовність до стабільної роботи в усіх сегментах мережі.

2.3.8. Налаштування DNS-серверу

Для забезпечення коректної роботи мережевих сервісів та спрощення доступу до ресурсів мережі було виконано налаштування DNS-сервера (рис. 2.20). Додано записи прямої трансляції для доменних імен, які забезпечують перетворення доменних назв на IP-адреси, що дозволяє користувачам звертатися до мережевих ресурсів за допомогою зручних URL.

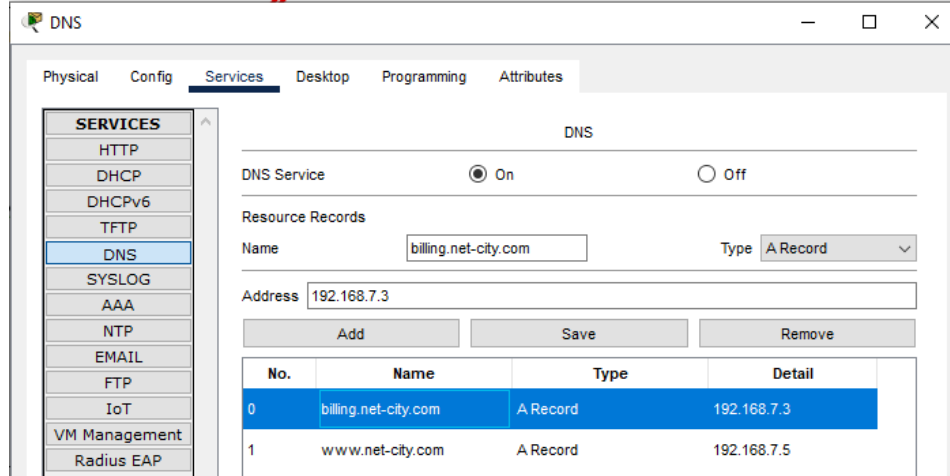


Рисунок 2.20 – Пряма трансляція ресурсів мережі

Налаштовані А-записи забезпечують зручний доступ до внутрішніх ресурсів компанії за допомогою доменних імен, що полегшує навігацію для користувачів та покращує загальну зручність користування мережею. Пряма трансляція DNS забезпечує правильне маршрутизацію запитів, що надходять до доменних імен, на відповідні IP-адреси мережевих ресурсів. Це налаштування є важливим для забезпечення стабільної роботи веб-сервісів та білінгової системи в мережі.

Для перевірки коректності налаштування DNS-сервісу та доступу до веб-сервера провайдера було здійснено тестовий запит з клієнтського комп'ютера PC4 до веб-сайту компанії за URL <http://www.net-city.com> (рис. 2.21).

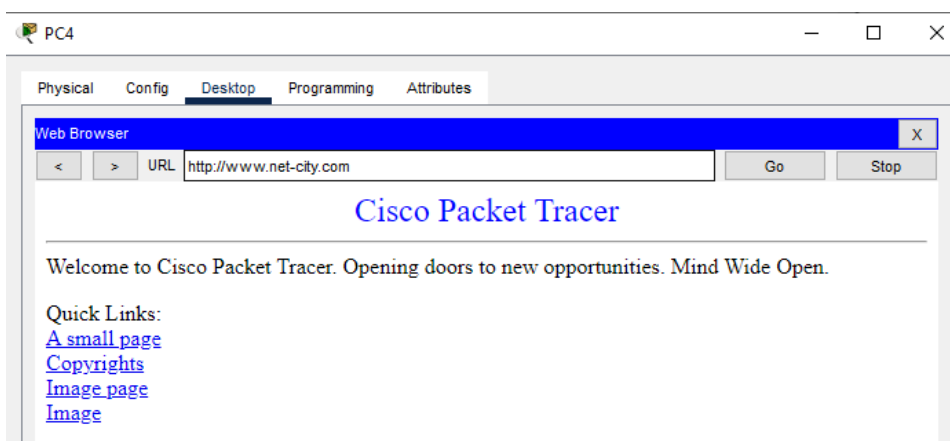


Рисунок 2.21 – Служба DNS у роботі

Результат перевірки підтверджує, що налаштований DNS-сервер коректно перетворює доменне ім'я www.net-city.com на IP-адресу веб-сервера, забезпечуючи доступ до ресурсу через мережу. Це свідчить про стабільну роботу DNS-сервісу та правильність налаштувань, виконаних на DNS-сервері.

Доступність веб-сайту провайдера через доменне ім'я підвищує зручність користування для клієнтів, дозволяючи їм легко отримувати доступ до необхідних мережевих сервісів та інформації.

2.3.9. Налаштування маршрутизації по EIGRP

На маршрутизаторі Zhitomir та маршрутизаторах точок доступу Korosten і Zvyagel було налаштовано протокол маршрутизації EIGRP (Enhanced Interior Gateway Routing Protocol) для забезпечення динамічного обміну маршрутною інформацією між пристроями (рис. 2.22). Ця конфігурація сприяє покращенню управління трафіком і підвищує гнучкість мережі, дозволяючи маршрутизаторам автоматично обмінюватися маршрутами для підключених мереж.

```
!  
router eigrp 10  
  redistribute static  
  redistribute connected  
  passive-interface GigabitEthernet0/0  
  passive-interface GigabitEthernet1/0  
  passive-interface GigabitEthernet9/0  
  network 192.168.1.0  
  network 192.168.2.0  
  auto-summary  
!  
ip route 0.0.0.0 0.0.0.0 88.88.88.1
```

Рисунок 2.22 – Налаштування EIGRP Zhitomir

Дана конфігурація дозволить маршрутизатору Zhitomir та маршрутизаторам точок доступу автоматично обмінюватися інформацією про мережі через протокол EIGRP, що сприяє покращенню управління трафіком і підвищенню гнучкості мережі. Виключення пасивних інтерфейсів запобігає зайвому обміну маршрутами через інтерфейси, які не беруть участь у маршрутизації. Встановлений статичний маршрут за замовчуванням надає можливість направляти невідомий трафік до зовнішнього шлюзу, забезпечуючи вихід до Інтернету для внутрішньої мережі.

Для налаштування протоколу EIGRP на маршрутизаторі Korosten було застосовано конфігурацію, яка забезпечує динамічний обмін маршрутною інформацією з іншими маршрутизаторами в мережі (рис. 2.23). Налаштування

включають розповсюдження підключених мереж, виключення пасивних інтерфейсів та оголошення мережі, через яку працюватиме протокол EIGRP.

```
!  
router eigrp 10  
redistribute connected  
passive-interface GigabitEthernet0/0  
passive-interface GigabitEthernet1/0  
network 192.168.1.0  
auto-summary  
!
```

Рисунок 2.23 – Налаштування EIGRP Korosten

Ця конфігурація дозволяє маршрутизатору Korosten брати участь в обміні маршрутами через EIGRP, забезпечуючи автоматичне додавання підключених мереж у маршрутну таблицю інших маршрутизаторів. Виключення пасивних інтерфейсів обмежує обмін маршрутами на інтерфейсах, де він не потрібен, що підвищує безпеку і стабільність мережі. Оголошена мережа 192.168.1.0 забезпечує доступність цього сегмента для всіх інших маршрутизаторів, підключених через EIGRP, а автоматичне сумування спрощує управління маршрутами у межах підмереж.

На маршрутизаторі Zvyagel було налаштовано протокол EIGRP для автоматичного обміну маршрутною інформацією в межах мережі, що забезпечує стабільне з'єднання між мережевими сегментами (рис. 2.24).

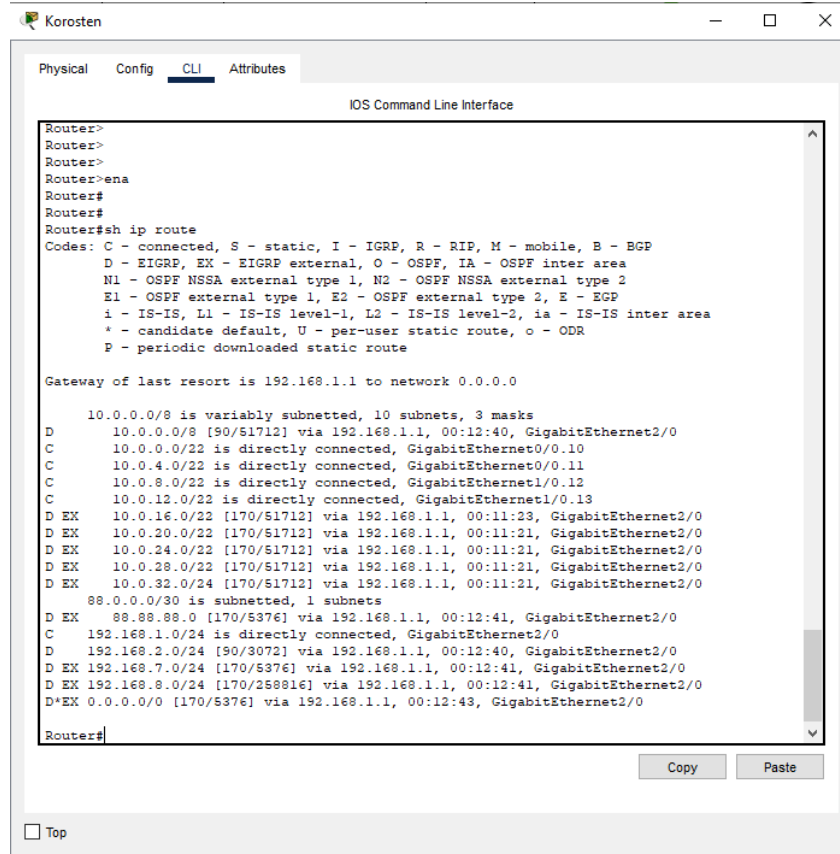
```
!  
router eigrp 10  
redistribute connected  
passive-interface GigabitEthernet0/0  
passive-interface GigabitEthernet1/0  
network 192.168.2.0  
auto-summary  
!
```

Рисунок 2.24 – Налаштування EIGRP Zvyagel

Конфігурація забезпечує ефективний обмін маршрутною інформацією через EIGRP на маршрутизаторі Zvyagel, автоматично додаючи підключені мережі до EIGRP для їх розповсюдження. Виключення пасивних інтерфейсів зменшує обмін маршрутами на інтерфейсах, де він не потрібен, що підвищує безпеку мережі. Оголошення мережі 192.168.2.0 дозволяє маршрутизатору забезпечувати зв'язок з іншими маршрутизаторами для доступу до цього сегмента мережі. Налаштування auto-summary спрощує управління маршрутами,

автоматично виконуючи сумування для класових меж, що робить мережу більш керованою та ефективною.

На маршрутизаторі Korosten було виконано перевірку таблиці маршрутів для оцінки коректності отриманих маршрутів від інших мережевих пристроїв через протокол EIGRP. Результати, відображені на рис. 2.25, підтверджують наявність маршрутів до всіх внутрішніх мереж, а також наявність маршруту за замовчуванням, що забезпечує вихід до зовнішньої мережі.



```
Router>
Router>
Router>
Router>ena
Router#
Router#
Router#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 192.168.1.1 to network 0.0.0.0

    10.0.0.0/8 is variably subnetted, 10 subnets, 3 masks
D    10.0.0.0/8 [90/51712] via 192.168.1.1, 00:12:40, GigabitEthernet2/0
C    10.0.0.0/22 is directly connected, GigabitEthernet0/0.10
C    10.0.4.0/22 is directly connected, GigabitEthernet0/0.11
C    10.0.8.0/22 is directly connected, GigabitEthernet1/0.12
C    10.0.12.0/22 is directly connected, GigabitEthernet1/0.13
D EX  10.0.16.0/22 [170/51712] via 192.168.1.1, 00:11:23, GigabitEthernet2/0
D EX  10.0.20.0/22 [170/51712] via 192.168.1.1, 00:11:21, GigabitEthernet2/0
D EX  10.0.24.0/22 [170/51712] via 192.168.1.1, 00:11:21, GigabitEthernet2/0
D EX  10.0.28.0/22 [170/51712] via 192.168.1.1, 00:11:21, GigabitEthernet2/0
D EX  10.0.32.0/24 [170/51712] via 192.168.1.1, 00:11:21, GigabitEthernet2/0
88.0.0.0/30 is subnetted, 1 subnets
D EX   88.88.88.0 [170/5376] via 192.168.1.1, 00:12:41, GigabitEthernet2/0
C    192.168.1.0/24 is directly connected, GigabitEthernet2/0
C    192.168.2.0/24 [90/3072] via 192.168.1.1, 00:12:40, GigabitEthernet2/0
D EX  192.168.7.0/24 [170/5376] via 192.168.1.1, 00:12:41, GigabitEthernet2/0
D EX  192.168.8.0/24 [170/258816] via 192.168.1.1, 00:12:41, GigabitEthernet2/0
D*EX  0.0.0.0/0 [170/5376] via 192.168.1.1, 00:12:43, GigabitEthernet2/0

Router#
```

Рисунок 2.25 – Маршрути на усі мережі та маршрут за замовчуванням

Таблиця маршрутів підтверджує, що маршрутизатор Korosten отримав всі необхідні маршрути до внутрішніх сегментів мережі через EIGRP. Крім того, наявність маршруту за замовчуванням забезпечує вихід до зовнішньої мережі через інтерфейс з IP-адресою 192.168.1.1. Це налаштування забезпечує стабільну та надійну маршрутизацію всередині мережі, дозволяючи ефективно обмінюватися даними між різними сегментами та забезпечуючи доступ до Інтернету.

Для перевірки коректності налаштувань маршрутизації між різними сегментами мережі було виконано трасування маршруту від клієнтського

комп'ютера PC4 до IP-адреси 10.0.16.7, яка належить іншому клієнтському пристрою у віддаленому сегменті мережі. Результати трасування, відображені на рис. 2.26, підтверджують стабільний і правильний маршрут, що дозволяє передавати дані між виносками через кілька проміжних маршрутизаторів.

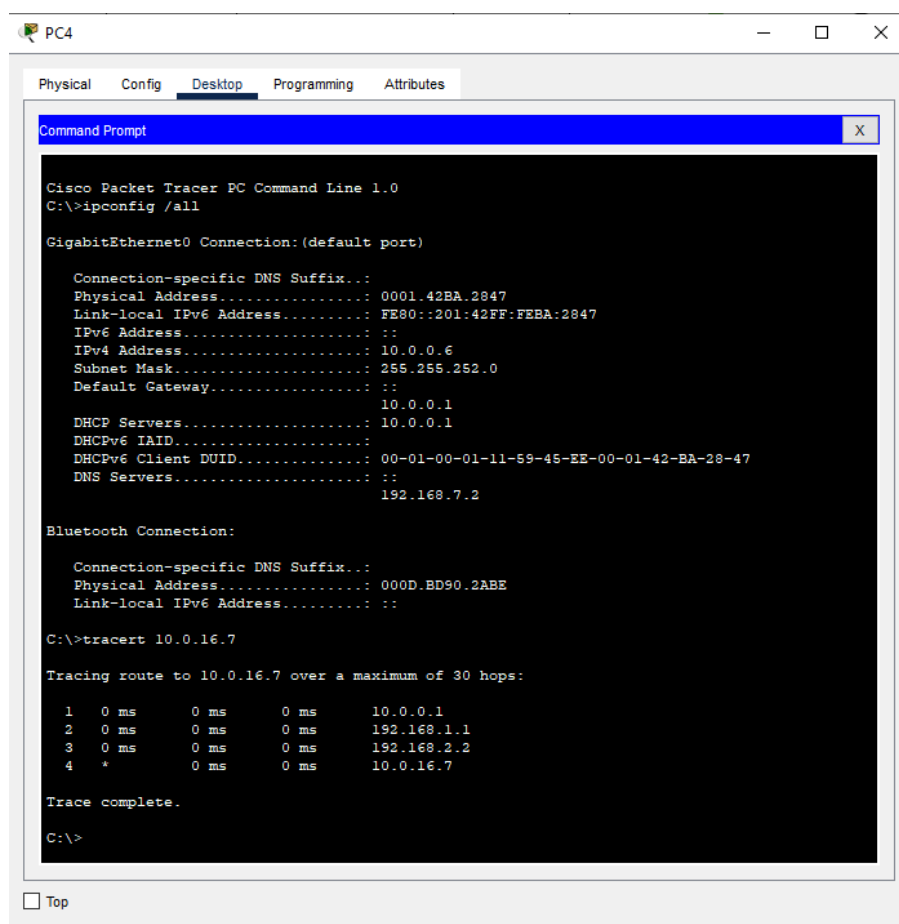


Рисунок 2.26 – Перевірка маршрутизації між виносками

Результати трасування підтверджують, що маршрутизація між виносками працює коректно, а всі проміжні маршрутизатори успішно передають трафік до цільового пристрою. Наявність стабільного маршруту через кілька проміжних вузлів свідчить про правильне налаштування EIGRP, яке забезпечує обмін маршрутною інформацією між різними сегментами мережі. Висока швидкість з'єднання та відсутність затримок вказують на ефективну роботу мережевої інфраструктури та надійність обміну даними між клієнтськими пристроями в різних VLAN.

2.3.10. Налаштування NAT на головному маршрутизаторі

Для забезпечення трансляції внутрішніх IP-адрес у публічні адреси було налаштовано NAT (Network Address Translation) на головному маршрутизаторі Zhitomir. Ця конфігурація дозволяє внутрішнім підмережам використовувати один зовнішній IP для виходу в Інтернет. На рис. 2.27 наведено лістинг налаштувань NAT, що включає директиви для вхідних і вихідних інтерфейсів, створення списку доступу для локальних підмереж, а також команду NAT із перевантаженням (overload).

```
!
interface GigabitEthernet0/0
 ip address 88.88.88.2 255.255.255.252
 ip nat outside
 duplex auto
 speed auto
!
interface GigabitEthernet1/0
 ip address 192.168.7.1 255.255.255.0
 ip nat inside
 duplex auto
 speed auto
!
interface GigabitEthernet2/0
 ip address 192.168.1.1 255.255.255.0
 ip nat inside
!
interface GigabitEthernet3/0
 ip address 192.168.2.1 255.255.255.0
 ip nat inside
!
...
interface GigabitEthernet9/0
 ip address 192.168.8.1 255.255.255.0
 ip nat inside
 duplex auto
 speed auto
!
!
ip nat inside source list 100 interface GigabitEthernet0/0 overload
!
!
access-list 100 permit ip 192.168.0.0 0.0.255.255 any
access-list 100 permit ip 10.0.0.0 0.255.255.255 any
!
```

Рисунок 2.27 – Налаштувань NAT на Zhitomir

Ця конфігурація NAT забезпечує стабільний і ефективний доступ внутрішніх підмереж до Інтернету, використовуючи один зовнішній IP-адрес через перевантажений NAT. Всі внутрішні підмережі, визначені у списку доступу, мають можливість виходу в Інтернет, а також отримують захист від зовнішніх з'єднань, оскільки NAT приховує внутрішні IP-адреси.

Для перевірки коректності налаштування NAT і доступу до Інтернету для клієнтів, підключених через вільну точку доступу, було виконано тестування з пристрою Smartphone0. Як показано на рис. 2.28, перевірка доступу до Інтернету була успішною, що підтверджує стабільність мережевого підключення та коректність налаштувань NAT на головному маршрутизаторі Zhitomir.

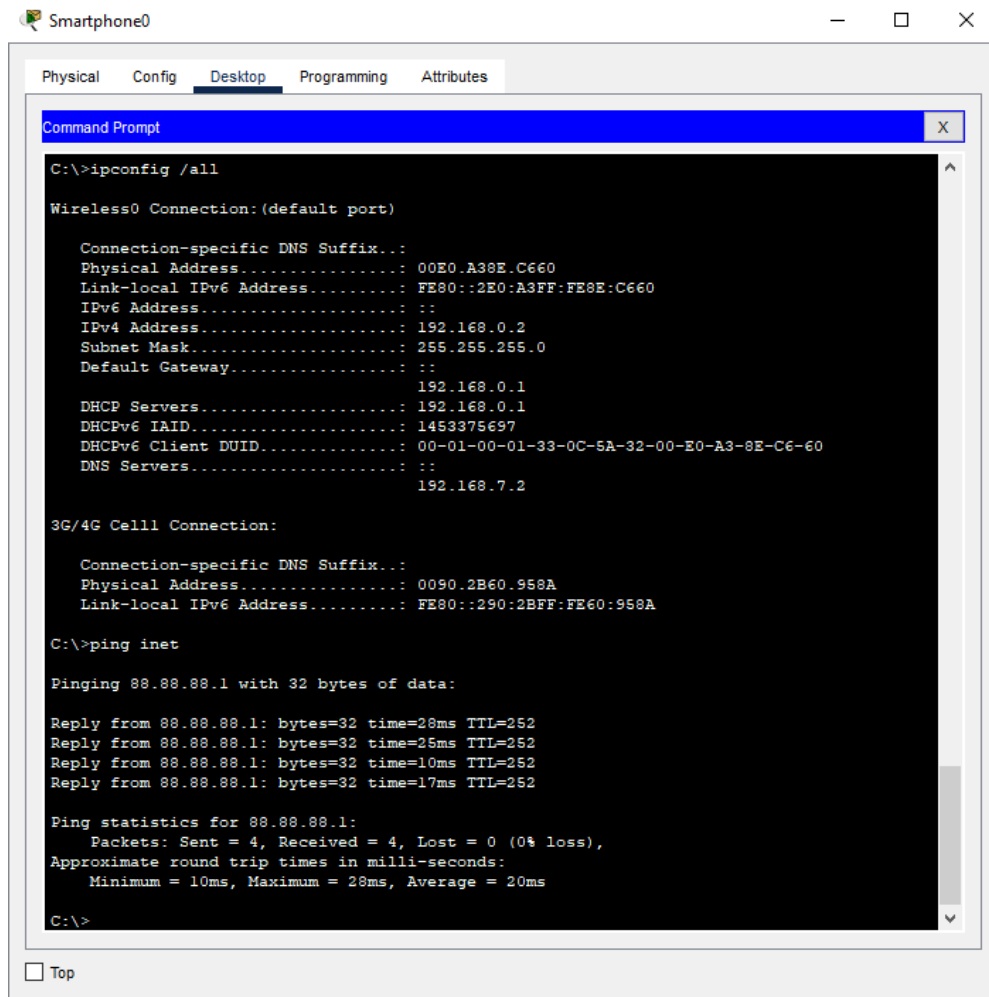


Рисунок 2.28 – Перевірка доступу до мережі інтернет

Результати перевірки підтверджують, що клієнт, підключений до вільної точки доступу через мережу Wi-Fi, має стабільний доступ до Інтернету завдяки коректним налаштуванням NAT на маршрутизаторі Zhitomir. Доступність зовнішньої IP-адреси та низький час відповіді свідчать про правильну роботу мережевої інфраструктури та відповідність налаштувань NAT вимогам мережевого доступу. Це забезпечує користувачам вільної зони Wi-Fi легкий і стабільний доступ до зовнішніх мереж.

2.4. Тестування працездатності мережі

Для перевірки надійності гіперканалу між комутатором точки доступу Switch-Zv-1 та комутатором виносу GPON-OLT-Zv-1-0 на виносі «Дружба» було виконано імітацію відмови одного з каналів оптичного з'єднання (рис. 2.29). Це тестування дозволяє оцінити, як система обробляє відмову одного з каналів і забезпечує безперервність з'єднання для клієнтів.

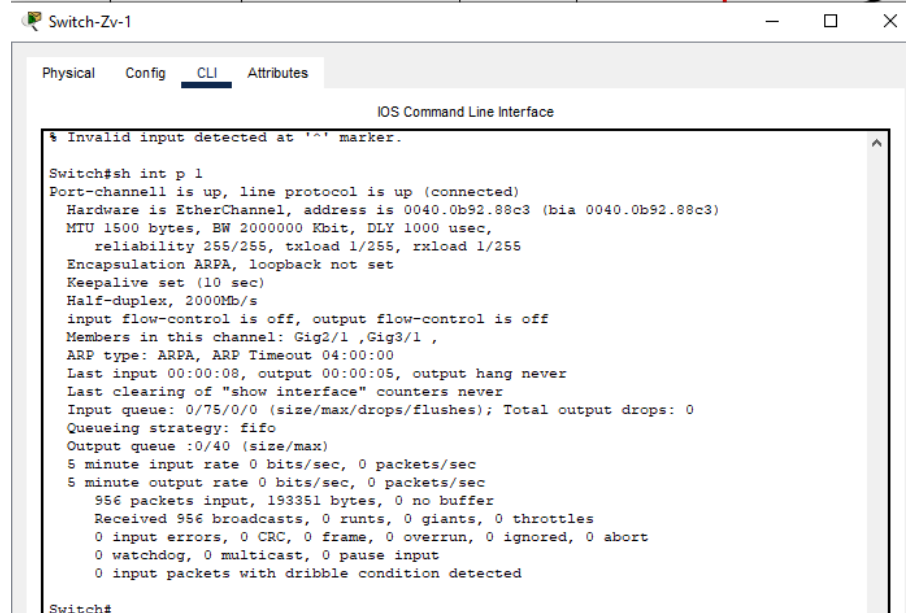


Рисунок 2.29 – Стан гіперканалу та його вміст до блокування (2 канали)

Даний сценарій тестування підтверджує, що мережеве з'єднання зберігається завдяки налаштуванням гіперканалу. Навіть у разі відмови одного з каналів система автоматично переключає трафік на активний інтерфейс, забезпечуючи безперервність з'єднання. Це гарантує, що клієнти на виносі «Дружба» зберігають доступ до внутрішніх ресурсів та Інтернету без помітних затримок чи втрат даних.

Щодо тестування стійкості гіперканалу на виносі «Дружба» було виконано імітацію відмови одного з каналів шляхом адміністративного відключення інтерфейсу Gig3/1 на комутаторі Switch-Zv-1. На рис. 2.30 показано, як інтерфейс Gig3/1 був переведений у стан administratively down, що імітує фізичний або програмний збій каналу з'єднання.

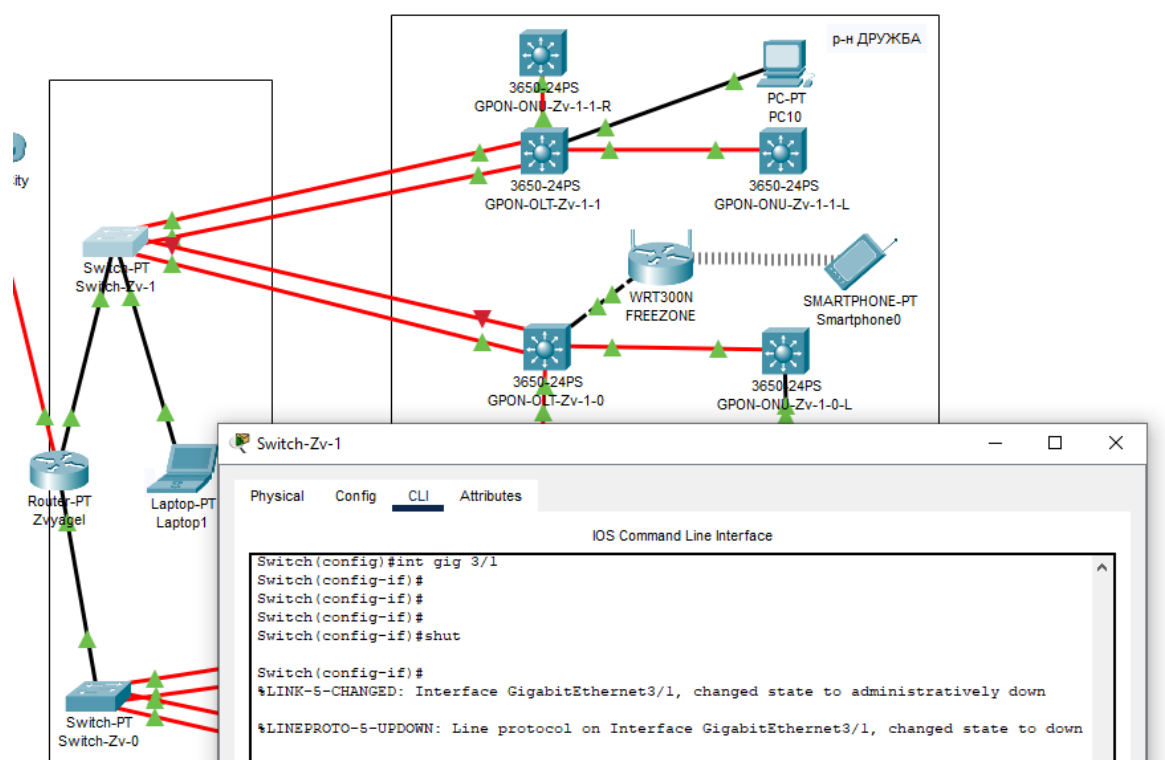
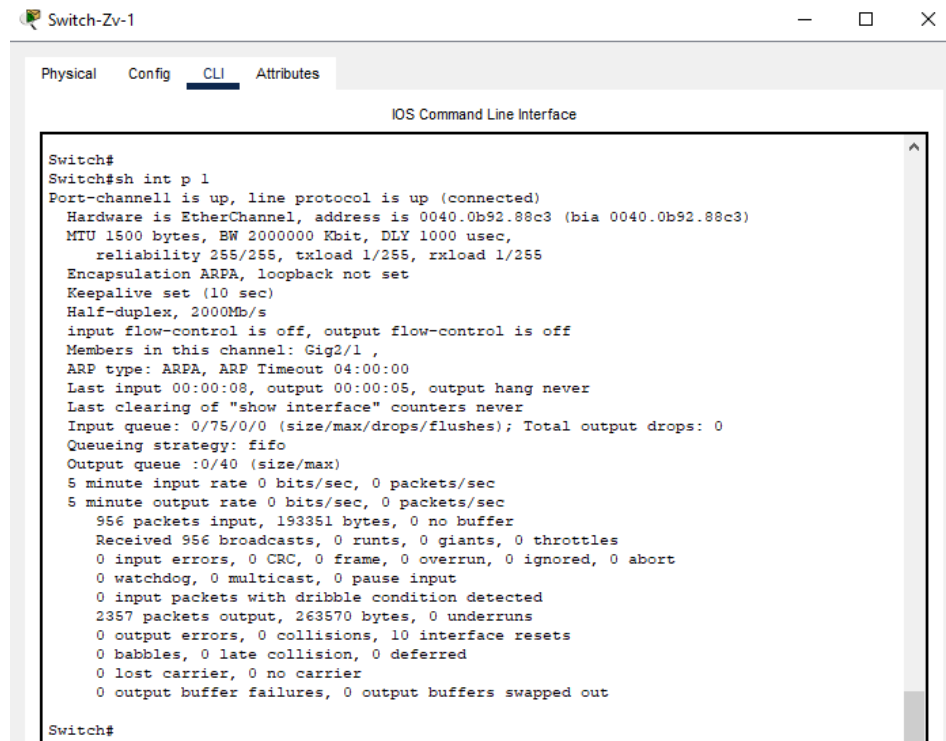


Рисунок 2.30 – Блокування інтерфейсу Gig3/1

Команди `interface GigabitEthernet3/1` та `shutdown` були застосовані для переведення інтерфейсу Gig3/1 у стан down. Після виконання цієї команди інтерфейс одразу перейшов у стан `administratively down`, що підтверджується повідомленнями `LINEPROTO-5-UPDOWN` та `LINK-3-CHANGED`.

Результати тестування підтверджують, що мережа стабільно функціонує навіть у разі виходу з ладу одного з інтерфейсів гіперканалу. Завдяки використанню LACP і можливості автоматичного перемикання на інший канал, забезпечується висока стійкість мережевої інфраструктури. Це рішення дозволяє уникнути збоїв у зв'язку та підтримує стабільну роботу мережевих сервісів для клієнтів навіть у разі відмови одного з фізичних з'єднань.

Після адміністративного відключення інтерфейсу Gig3/1 гіперканал на комутаторі Switch-Zv-1 продовжує працювати завдяки активному інтерфейсу Gig1/1, що залишився в роботі. На рис. 2.31 відображено поточний стан гіперканалу Port-channell, який працює з одним каналом після блокування одного з інтерфейсів.



```
Switch-Zv-1
Physical Config CLI Attributes
IOS Command Line Interface

Switch#
Switch#sh int p 1
Port-channel1 is up, line protocol is up (connected)
Hardware is EtherChannel, address is 0040.0b92.88c3 (bia 0040.0b92.88c3)
MTU 1500 bytes, BW 2000000 Kbit, DLY 1000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
Keepalive set (10 sec)
Half-duplex, 2000Mb/s
input flow-control is off, output flow-control is off
Members in this channel: Gig2/1 ,
ARP type: ARPA, ARP Timeout 04:00:00
Last input 00:00:08, output 00:00:05, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue :0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
    956 packets input, 193351 bytes, 0 no buffer
    Received 956 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    0 watchdog, 0 multicast, 0 pause input
    0 input packets with dribble condition detected
    2357 packets output, 263570 bytes, 0 underruns
    0 output errors, 0 collisions, 10 interface resets
    0 babbles, 0 late collision, 0 deferred
    0 lost carrier, 0 no carrier
    0 output buffer failures, 0 output buffers swapped out

Switch#
```

Рисунок 2.31 – Стан гіперканалу та його вміст після блокування (1 канал)

Port-channel1 перебуває в стані up, що означає, що гіперканал продовжує функціонувати і передавати дані, незважаючи на відключення одного з каналів. Активним учасником гіперканалу залишається тільки інтерфейс Gig1/1, який забезпечує стабільне з'єднання для клієнтів.

Результати моніторингу підтверджують, що гіперканал Port-channel1 зберігає стабільність навіть після втрати одного з фізичних каналів. Активний інтерфейс Gig1/1 успішно забезпечує зв'язок, а протокол LACP ефективно адаптував гіперканал до нових умов. Це свідчить про надійність і відмовостійкість мережевої архітектури, яка може витримувати збої без втрати з'єднання та збереження доступу до мережевих ресурсів для клієнтів.

Для перевірки доступності зовнішньої мережі після імітації відмови одного з каналів гіперканалу на виносі «Дружба» було виконано команду `tracert` з клієнтського комп'ютера РС9 до зовнішньої IP-адреси 88.88.88.1. Результати тестування, наведені на рис. 2.32, підтверджують, що доступ до мережі Інтернет збережений.

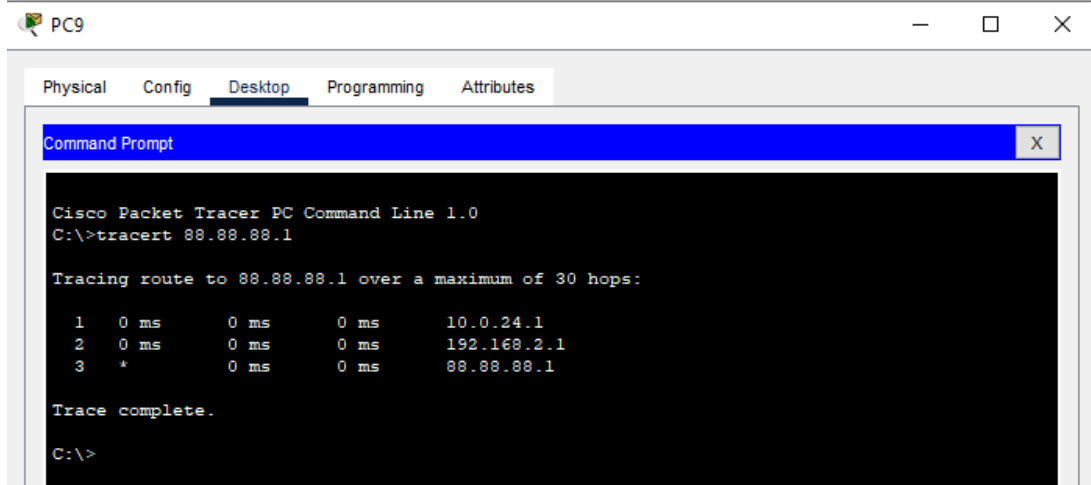


Рисунок 2.32 – Тестування доступу до мережі (доступ збережений)

Результати трасування підтверджують, що після відмови одного з каналів гіперканалу мережа успішно перенаправляє весь трафік через активний інтерфейс, зберігаючи стабільний доступ до зовнішньої мережі для клієнтів. Це доводить надійність налаштувань гіперканалу, який продовжує працювати навіть у зменшеній конфігурації, забезпечуючи безперервний зв'язок для кінцевих користувачів.

Висновок до другого розділу

У рамках даного розділу здійснено комплексне моделювання мережевої інфраструктури ТОВ «Net City» для абонентського підключення, що забезпечує надійність і гнучкість з використанням технологій GPON та DWDM. Виконано аналіз технічного завдання, де визначено вимоги до побудови мережі з інтеграцією сучасних технологій для міст Коростень і Звягель. Проведено планування структури мережі, зокрема створено логічні та функціональні схеми для інтернет-провайдера та адміністративної мережі, а також розроблено схеми гіперканалів через DWDM та логічних сегментів мережі для віддалених підключень.

Розроблено схему адресації для всіх точок доступу, зокрема виконано розподіл адресації для вузлів у містах Коростень і Звягель та резервування адрес для управління мережею. Для захисту та безпеки мережі на всіх пристроях встановлено паролі для доступу до консолі та режиму конфігурації. Налаштування VLAN та DHCP дозволило ізолювати трафік у різних сегментах

мережі, а конфігурація VLAN на комутаторах точок доступу та виносів забезпечила відповідність мережевої архітектури до вимог клієнтської ізоляції та управління.

Особлива увага приділена налаштуванню гіперканалів, що забезпечують стабільне та швидке з'єднання, а також гарантовану відмовостійкість завдяки дублюванню каналів. Було перевірено працездатність гіперканалів при імітації відмови одного з каналів, що продемонструвало збереження доступу до внутрішньої мережі та Інтернету завдяки автоматичному перемиканню на інший канал. Також налаштовано адресацію для клієнтів через DHCP, що дозволяє автоматизувати процес підключення та управління адресами.

Для забезпечення стабільного інтернет-з'єднання налаштовано DNS-сервер, який виконує пряму трансляцію мережевих ресурсів, що підтверджено успішною роботою служби DNS. Протокол маршрутизації EIGRP налаштовано на головному маршрутизаторі та точках доступу, що забезпечує автоматичний обмін маршрутами між мережевими вузлами та підтримує доступність усіх сегментів мережі, включно з маршрутом за замовчуванням. Окрім того, реалізовано NAT для виходу клієнтів у зовнішню мережу через загальну IP-адресу, що дозволило ефективно використовувати адресний простір та забезпечити безпечний доступ до Інтернету.

На етапі тестування перевірено коректність усіх налаштувань та стабільність роботи мережі при різних сценаріях, зокрема при імітації відмови одного з каналів гіперканалу. Результати тестування підтвердили, що розроблена мережа відповідає заданим вимогам і забезпечує стабільний зв'язок для клієнтів, зокрема у віддалених сегментах, зберігаючи доступ до внутрішніх ресурсів та Інтернету.

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ОЦІНКА ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ ВПРОВАДЖЕННЯ ЕНЕРГОНЕЗАЛЕЖНОЇ МЕРЕЖНОЇ ІНФРАСТРУКТУРИ

3.1. Аналіз вимог до програмного забезпечення для моніторингу мережі

Програмне забезпечення для моніторингу мережі, враховуючи тематику побудови енергонебезпечної інфраструктури, має забезпечувати високу надійність, швидкодію та точність аналізу даних. У контексті розробки системи на основі машинного навчання, головними функціональними вимогами є:

- тренування моделей (система повинна підтримувати можливість навчання моделей машинного навчання для прогнозування несправностей);
- забезпечення розділення даних на тренувальні та тестові набори (обробка відсутніх значень і створення набору ознак для побудови моделі);
- зберігання моделей (збереження натренованих моделей у локальному сховищі з можливістю їх подальшого використання; реалізація механізмів версіонування моделей для підтримки історії змін);
- моніторинг стану обладнання (отримання даних від сенсорів та обладнання в реальному часі; відображення поточного стану ключових параметрів, таких як температура, швидкість обертання, крутний момент, тощо);
- прогнозування несправностей (можливість здійснювати класифікацію стану вузлів як «норма» або «відмова»; обчислення ймовірності несправності для кожного мережного елемента);
- виведення рекомендацій для запобігання можливим відмовам.

Нефункціональні вимоги до ПЗ складаються із:

- продуктивність (система повинна обробляти вхідні дані в режимі реального часу без значних затримок; час прогнозування для кожного вузла має бути мінімізований);
- надійність (гарантія коректної роботи навіть у разі некоректного або неповного введення даних; резервування даних моделей і журналів операцій для запобігання втратам);

- енергоефективність (зниження енергоспоживання під час роботи для підтримки автономності мережної інфраструктури; використання оптимізованих алгоритмів машинного навчання);
- масштабованість система повинна легко адаптуватися до зростання кількості вузлів у мережі без втрати продуктивності; підтримка паралельної обробки даних із декількох джерел);
- зручність використання (інтуїтивний інтерфейс для налаштування параметрів системи та перегляду результатів; автоматизація ключових процесів, таких як навчання моделей та оновлення прогнозів).

Даний перелік вимог забезпечує основу для побудови функціонального та надійного ПЗ, яке відповідатиме потребам моніторингу мережної інфраструктури.

3.2. Вибір алгоритмів машинного навчання для аналізу та прогнозування стану мережі

Розробка програмного забезпечення для аналізу та прогнозування стану мережі передбачає використання методів машинного навчання, здатних ефективно працювати з великими обсягами даних та забезпечувати високу точність результатів. Алгоритми мають враховувати особливості даних, такі як багатовимірність ознак, можливість наявності пропущених значень та нерівномірний розподіл класів. Вибір підходящого алгоритму базується на балансі між продуктивністю, простотою реалізації та точністю прогнозів. У даному контексті найбільш доцільним є використання деревоподібних методів, які добре справляються із задачами класифікації та регресії, навіть за умов високої варіативності даних.

Швидке дерево (Fast Tree, ШД) – це метод машинного навчання, який базується на побудові дерев рішень із застосуванням ефективного алгоритму пошуку оптимального поділу даних [26]. Цей метод оптимізує функцію втрат за допомогою градієнтного спуску, що дозволяє досягати високої швидкості навчання. Основною перевагою є його здатність працювати з великими наборами

даних, забезпечуючи добру масштабованість. ШД зазвичай використовується для класифікації та регресії, особливо коли потрібна швидка генерація прогнозів.

Градiєнтний бустинг дерев рішень (Gradient Boosted Decision Trees, ГБДР) – це ансамблевий метод, який поєднує кілька дерев рішень для побудови потужної моделі з високою узагальнюючою здатністю [27]. Кожне наступне дерево навчається на залишкових помилках попередніх, поступово мінімізуючи функцію втрат. Цей підхід забезпечує високу точність і стійкість до перенавчання навіть у складних задачах із нерівномірним розподілом класів. Градiєнтний бустинг добре підходить для аналізу складних залежностей у даних і використовується у задачах класифікації та регресії.

Випадковий ліс (Random Forest, ВЛ) є ансамблевим методом, який використовує велику кількість дерев рішень, що будуються на різних підмножинах даних і ознак [28]. Основною перевагою цього методу є його здатність знижувати ризик перенавчання за рахунок використання випадковості при побудові кожного дерева. Випадковий ліс забезпечує високу точність, зберігаючи стійкість до шуму в даних і пропущених значень. Метод ефективно працює у задачах, де потрібна інтерпретованість результатів і стабільність класифікації.

Для вибору найкращого алгоритму необхідно враховувати специфіку задачі та особливості кожного методу. Кожен із розглянутих алгоритмів має свої переваги та обмеження, що впливають на точність, продуктивність та здатність працювати з різними типами даних. Для наочного аналізу методів проведено їх порівняння у табл. 3.1 за ключовими характеристиками, які важливі для аналізу та прогнозування стану мережі.

Таблиця 3.1

Порівняльний аналіз алгоритмів машинного навчання

Характеристика	ШД	ГБДР	ВЛ
Швидкість навчання	Висока	Середня	Середня
Швидкість прогнозування	Висока	Середня	Висока
Стійкість до перенавчання	Висока	Висока	Висока
Масштабованість	Висока	Середня	Середня
Вимоги до обчислювальних ресурсів	Низькі	Високі	Середні

Точність класифікації	Висока	Дуже висока	Висока
Робота з нерівномірними класами	Ефективна	Дуже ефективна	Ефективна
Простота інтерпретації	Середня	Низька	Середня

Вибір методу ШД для розробки системи зумовлений його високою швидкістю навчання та прогнозування, що є критично важливим для аналізу даних у реальному часі. Завдяки низьким вимогам до обчислювальних ресурсів, ШД добре підходить для масштабованих систем, які повинні ефективно працювати з великими наборами даних. Крім того, метод забезпечує високу точність класифікації при збереженні стійкості до перенавчання, що є ключовим фактором для прогнозування стану мережі. Висока продуктивність та здатність працювати в умовах обмежених ресурсів роблять ШД оптимальним вибором для розробки енергоефективної інфраструктури.

3.3. Попередня обробка даних та особливості роботи з датасетом

Ефективність моделі машинного навчання значною мірою залежить від якості даних, які використовуються для її навчання та тестування. Попередня обробка даних є невід'ємним етапом, що включає очищення, інтеграцію та формування основного набору характеристик, які впливають на прогнозування. Збір даних із різних джерел може супроводжуватися шумами, відсутніми значеннями чи надмірною кількістю ознак, що потребує відповідної обробки. Для забезпечення точності та узагальнюючої здатності моделі необхідно ретельно структурувати процес роботи з датасетом, охоплюючи всі етапи підготовки та побудови моделі.

На рис. 3.1 наведена структура побудови моделі машинного навчання, яка відображає основні кроки створення ефективної системи аналізу та прогнозування.

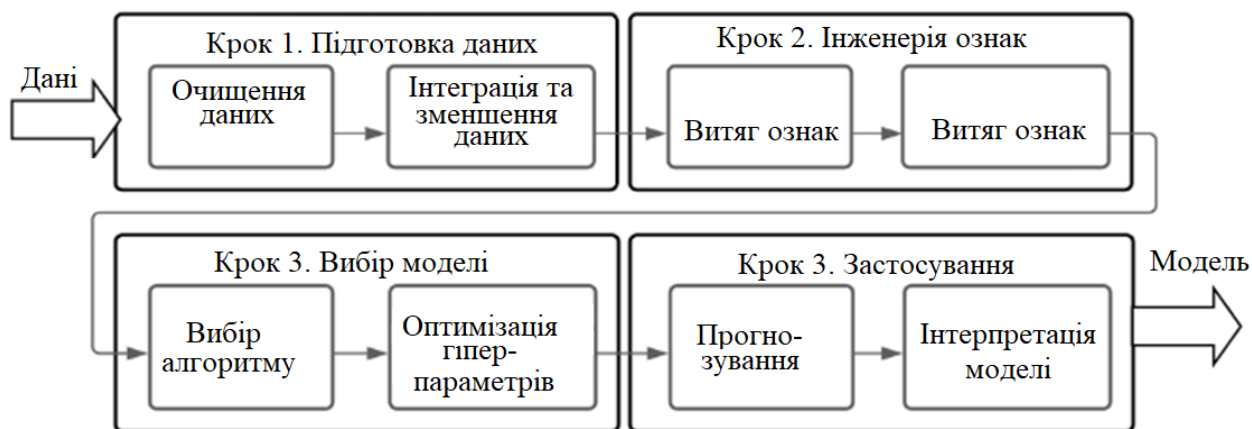


Рисунок 3.1 – Структура побудови моделей МН

На початку здійснюється підготовка даних, яка включає очищення та зменшення обсягу даних за рахунок їхньої інтеграції, що допомагає забезпечити їх якість і релевантність. Далі відбувається інженерія ознак, яка охоплює вилучення ключових характеристик і формування набору ознак для подальшого навчання. Третій етап присвячено вибору моделі: визначення найбільш придатного алгоритму машинного навчання та налаштування його гіперпараметрів для досягнення оптимальних результатів. Завершальний етап передбачає застосування моделі для прогнозування та аналізу отриманих результатів із подальшою їхньою інтерпретацією.

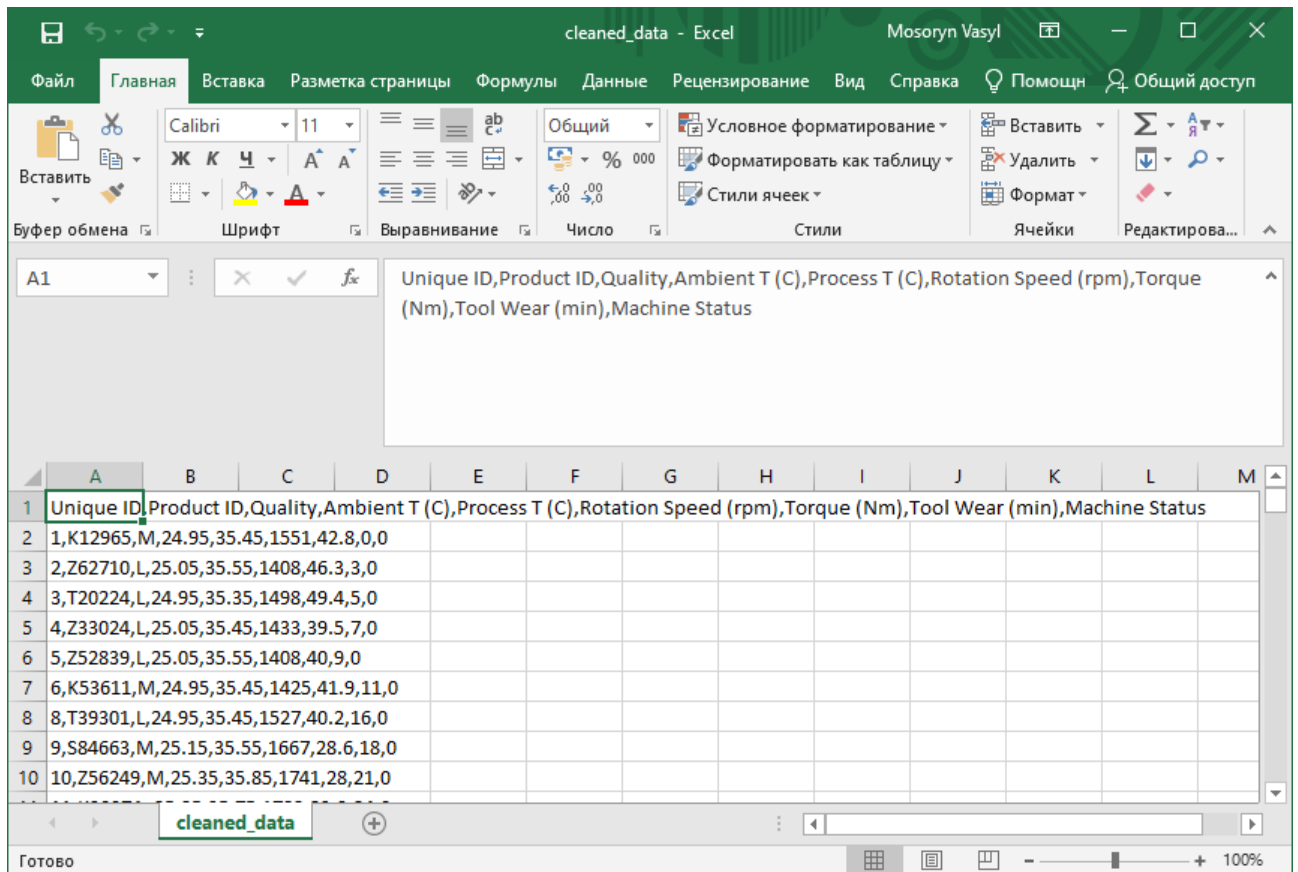
Ця структура забезпечує систематичний підхід до створення моделей машинного навчання, де кожен етап доповнює попередній, створюючи основу для надійного та точного прогнозування стану мережі.

Для задачі аналізу стану мережі та прогнозування несправностей було обрано датасет «Factory Data (Classification)», доступний на платформі Kaggle [29]. Цей набір даних містив показники, що відігравали ключову роль у визначенні функціонального стану обладнання, такі як температура, швидкість обертання, крутний момент та інші параметри. Завдяки структурованості та наявності чітко визначених міток, цей датасет підходив для задач класифікації та був основою для навчання моделі.

Після вибору датасету було проведено очищення даних, яке включало видалення шумів, обробку пропущених значень і забезпечення коректного форматування полів. Цей процес дозволив усунути потенційні проблеми, що могли вплинути на якість тренування моделі, і підготувати дані до подальшого

аналізу. У результаті сформувавши новий структурований датасет, який містив ключові ознаки, необхідні для побудови надійної моделі прогнозування.

На рис. 3.2 наведено фрагмент вікна із сформованим датасетом, що відображає фінальний вигляд очищених і підготовлених даних, готових для використання в навчанні моделі. Цей результат підтверджує ретельну роботу на етапі підготовки даних та їхнє успішне формування для подальших етапів аналізу.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Unique ID,Product ID,Quality,Ambient T (C),Process T (C),Rotation Speed (rpm),Torque (Nm),Tool Wear (min),Machine Status												
2	1,K12965,M,24.95,35.45,1551,42.8,0,0												
3	2,Z62710,L,25.05,35.55,1408,46.3,3,0												
4	3,T20224,L,24.95,35.35,1498,49.4,5,0												
5	4,Z33024,L,25.05,35.45,1433,39.5,7,0												
6	5,Z52839,L,25.05,35.55,1408,40,9,0												
7	6,K53611,M,24.95,35.45,1425,41.9,11,0												
8	8,T39301,L,24.95,35.45,1527,40.2,16,0												
9	9,S84663,M,25.15,35.55,1667,28.6,18,0												
10	10,Z56249,M,25.35,35.85,1741,28,21,0												

Рисунок 3.2 – Фрагмент сформованого датасету

Цей датасет містив численні записи про роботу промислового обладнання, представлені у вигляді числових і категоріальних ознак, що відображають стан системи. Основною особливістю цього набору даних була наявність міток, які чітко ідентифікували стан кожної машини, що дозволяло використовувати їх для задач класифікації. Усі параметри були структуровані таким чином, щоб забезпечити інформативність і повноту даних для навчання моделі машинного навчання. У табл. 3.2 наведено детальний опис ключових атрибутів датасету, які використовувалися для побудови моделі. Кожен із цих атрибутів має своє

значення у процесі аналізу й моделювання, забезпечуючи можливість виявлення закономірностей та прогнозування несправностей.

Таблиця 3.2

Атрибути датасету

№	Атрибут	Опис
1	Unique ID	Унікальний ідентифікатор кожного запису в наборі даних, який використовується для ідентифікації конкретного екземпляра.
2	Product ID	Ідентифікатор конкретного продукту або обладнання, що генерує дані, який допомагає співвідносити записи з джерелами.
3	Quality	Категоріальний атрибут, що відображає якість виробленої продукції, наприклад, "добра" чи "несправна".
4	Ambient T (C)	Температура навколишнього середовища, яка фіксується у градусах Цельсія і впливає на роботу обладнання.
5	Process T (C)	Температура процесу в градусах Цельсія, яка є одним із критичних параметрів стану машини під час виконання задач.
6	Rotation Speed (rpm)	Швидкість обертання механізму, вимірювана в обертах за хвилину, що характеризує динамічні властивості обладнання.
7	Torque (Nm)	Крутний момент у Ньютон-метрах, який відображає навантаження на механізм під час виконання операцій.
8	Tool Wear (min)	Час зношування інструменту в хвилинах, що дозволяє оцінити ступінь деградації робочих частин.
9	Machine Status	Мітка класу, яка вказує на стан машини: нормальний (0) або несправний (1), використовується як цільова змінна для задачі класифікації.

Ці атрибути забезпечували необхідну інформацію для аналізу робочих параметрів обладнання, дозволяючи будувати точні моделі машинного навчання для прогнозування несправностей.

3.4. Розробка коду для тренування моделей для прогнозування стану мережі

Розробка програмного забезпечення для прогнозування стану мережі базується на створенні ефективних алгоритмів, які враховують ключові параметри обладнання та їхній вплив на результат [32]. Повний код наведено у

додатку А. Для реалізації цього підходу необхідно забезпечити коректну обробку вхідних даних і структурування результатів, що дозволяє не лише навчати модель, але й ефективно аналізувати її прогнози. Основою для цього є визначення класів, які слугують інтерфейсом між даними та моделлю машинного навчання. На рис. 3.3 наведено структуру класів, які були розроблені для роботи із вхідними параметрами обладнання та результатами прогнозування.

```
public class RawMachineData {
    // Температура навколишнього середовища в градусах Цельсія
    // Значення для введення користувачем
    [LoadColumn(3)]
    5 references
    public float AmbientTemperature { get; set; }

    // Температура процесу в градусах Цельсія
    // Значення для введення користувачем
    [LoadColumn(4)]
    5 references
    public float ProcessTemperature { get; set; }

    // Швидкість обертання в оборотах за хвилину (rpm)
    // Значення для введення користувачем
    [LoadColumn(5)]
    5 references
    public float RotationSpeed { get; set; }

    // Крутий момент в Ньютон-метрах (Nm)
    // Значення для введення користувачем
    [LoadColumn(6)]
    5 references
    public float Torque { get; set; }

    // Зношування інструменту в хвилинали
    // Значення для введення користувачем
    [LoadColumn(7)]
    5 references
    public float ToolWear { get; set; }

    // Стан машини: true (1) - відмова, false (0) - нормальний стан
    // Це поле буде ігноруватися під час прогнозування, але використовується під час тренування моделі
    [LoadColumn(8)]
    2 references
    public bool Label { get; set; }
}

// Прогнозований стан машини: true - відмова,
// false - нормальний стан
[ColumnName("PredictedLabel")]
2 references
public bool Prediction { get; set; }

// Ймовірність передбаченого результату
// Відображає впевненість моделі у своєму прогнозі
2 references
public float Probability { get; set; }

// Оцінка моделі, яка вказує на ступінь відповідності вхідним даним
0 references
public float Score { get; set; }
}
```

Рисунок 3.3 – Класи для роботи із вхідними параметрами обладнання

Клас `RawMachineData` визначає структуру вхідних даних, які використовуються для тренування моделі та подальшого прогнозування. Кожен атрибут цього класу відповідає окремій характеристиці обладнання. Температура навколишнього середовища та температура процесу є ключовими показниками, що відображають умови експлуатації системи. Швидкість обертання та крутий момент характеризують динамічні аспекти роботи обладнання, тоді як параметр зношування інструменту дозволяє оцінити ступінь його деградації. Останнє поле, яке представляє стан машини, використовується виключно для навчання, вказуючи, чи відбулася відмова в конкретному випадку.

`MachinePrediction` у свою чергу призначений для зберігання результатів прогнозування, які повертає модель. Основним його елементом є поле, що представляє прогнозований стан машини, вказуючи на ймовірність відмови чи

нормального функціонування. Додатково, клас містить інформацію про ймовірність передбаченого результату, яка відображає впевненість моделі у зробленому прогнозі. Поле оцінки моделі забезпечує додаткову інформацію для аналізу, дозволяючи більш детально зрозуміти поведінку алгоритму та відповідність його прогнозів вхідним даним.

Фрагмент коду наведений на рис. 3.4 відповідає за обробку події натискання кнопки, яка відкриває діалогове вікно для вибору файлу.

```
private async void OpenBtn_Click(object sender, EventArgs e) {  
    // Створення діалогового вікна для відкриття файлу  
    OpenFileDialog openFileDialog = new OpenFileDialog();  
    // Налаштування властивостей діалогового вікна  
    openFileDialog.Filter = "Файли CSV (*.csv)|*.csv|Усі файли (*.*)|*.*";  
    openFileDialog.FilterIndex = 2;  
    openFileDialog.RestoreDirectory = true;  
  
    // Відображення діалогового вікна та обробка результату  
    if (openFileDialog.ShowDialog() == DialogResult.OK) {  
        _Path = openFileDialog.FileName;  
        FileNameTBox.Text = openFileDialog.FileName;  
    }  
}
```

Рисунок 3.4 – Код для відкриття файлу із тренувальними даними

При натисканні кнопки створюється об'єкт діалогового вікна `OpenFileDialog`, який дозволяє користувачу обирати файл із файлової системи. Параметри діалогового вікна налаштовуються таким чином, щоб обмежити вибір користувача лише файлами формату CSV, а також дозволити перегляд інших типів файлів. Після налаштування діалогового вікна викликається метод `ShowDialog`, який відкриває вікно вибору файлу. Якщо користувач успішно обирає файл і підтверджує свій вибір, шлях до вибраного файлу зберігається у змінну `_Path`, а також відображається у текстовому полі `FileNameTBox`. Цей механізм забезпечує зручність інтеграції зовнішніх даних у систему.

На рис. 3.5 наведено фрагмент коду, який ініціалізує контекст машинного навчання, створюючи об'єкт `MLContext`, який забезпечує всі необхідні функції для роботи з ML.NET.

```
// Створення контексту ML  
mlContext = new MLContext(seed: 0);  
  
// Завантаження даних як RawMachineData  
dataView = mlContext.Data.LoadFromTextFile<RawMachineData>(  
    _Path,  
    separatorChar: ',',  
    hasHeader: true);
```

Рисунок 3.5 – Код для відкриття файлу із тренувальними даними

Для забезпечення відтворюваності результатів при роботі з випадковістю використовується фіксоване значення насіння (seed: 0). Після цього виконується завантаження даних із файлу, шлях до якого визначається змінною `_Path`. Дані завантажуються у форматі `RawMachineData`, що дозволяє зіставити структуру вхідного файлу зі структурою класу, визначеного для обробки даних. У процесі завантаження враховується розділювач (кома) та наявність заголовка, що дозволяє правильно інтерпретувати дані з файлу.

Код на рис. 3.6 відповідає за виконання асинхронного завдання тренування моделі машинного навчання. На початковому етапі відбувається розділення набору даних на дві частини: одну для навчання моделі, іншу для її тестування. Для цього використовується метод `TrainTestSplit`, який забезпечує розподіл даних таким чином, щоб 80% записів використовувалися для тренування, а решта 20% – для оцінки ефективності моделі. Після виконання цього поділу дані для тренування зберігаються у змінній `trainingData`, а тестовий набір – у змінній `testData`. Такий підхід дозволяє моделі навчитися на одній частині даних і перевірити свою точність на іншій, забезпечуючи об’єктивну оцінку її роботи.

```
private async Task TrainModelAsync() {  
    // Розділення даних на тренувальні та тестові  
    var trainTestData = mlContext.Data.TrainTestSplit(dataView, testFraction: 0.2);  
    var trainingData = trainTestData.TrainSet;  
    var testData = trainTestData.TestSet;
```

Рисунок 3.6 – Виконання асинхронного завдання для тренування моделі

Фрагмент коду на рис. 3.7 визначає процес підготовки даних і навчання моделі машинного навчання. На першому етапі виконується обробка відсутніх значень у наборах даних, зокрема для таких ознак, як температура навколишнього середовища, температура процесу, швидкість обертання, крутний момент і ступінь зношування інструменту. Для кожної з них відсутні значення замінюються середніми значеннями, що дозволяє зберегти цілісність даних. Після цього відбувається об’єднання всіх ознак у єдиний вектор, який отримує назву «Features», для спрощення використання в моделі.

```

var trainingPipeline = mlContext.Transforms.ReplaceMissingValues("AmbientTemperature",
replacementMode: MissingValueReplacingEstimator.ReplacementMode.Mean)
.Append(mlContext.Transforms.ReplaceMissingValues("ProcessTemperature",
replacementMode: MissingValueReplacingEstimator.ReplacementMode.Mean))
.Append(mlContext.Transforms.ReplaceMissingValues("RotationSpeed",
replacementMode: MissingValueReplacingEstimator.ReplacementMode.Mean))
.Append(mlContext.Transforms.ReplaceMissingValues("Torque",
replacementMode: MissingValueReplacingEstimator.ReplacementMode.Mean))
.Append(mlContext.Transforms.ReplaceMissingValues("ToolWear",
replacementMode: MissingValueReplacingEstimator.ReplacementMode.Mean))
.Append(mlContext.Transforms.Concatenate("Features",
    nameof(RawMachineData.AmbientTemperature),
    nameof(RawMachineData.ProcessTemperature),
    nameof(RawMachineData.RotationSpeed),
    nameof(RawMachineData.Torque),
    nameof(RawMachineData.ToolWear)))
.Append(mlContext.BinaryClassification.Trainers.FastTree(
    numberOfLeaves: 20,
    numberOfTrees: 100,
    minimumExampleCountPerLeaf: 10));

```

Рисунок 3.7 – Процес підготовки даних тренування моделі

Фінальним етапом є застосування алгоритму «Швидке дерево», який використовується для задачі двійкової класифікації. У процесі налаштування моделі задається кількість листків, дерев та мінімальна кількість прикладів для кожного листка, що дозволяє адаптувати її до специфіки даних. У результаті формується конвеєр обробки та навчання моделі, який об'єднує етапи трансформації даних із процесом класифікації.

Код на рис. 3.8 вимірює тривалість тренування моделі, щоб оцінити швидкість її навчання. Для цього використовується таймер, ініційований через об'єкт Stopwatch, який запускається перед початком тренування. Перш ніж розпочати навчання, виконується перевірка наявності достатньої кількості даних у тренувальному наборі. Якщо дані доступні, модель тренується за допомогою обраного конвеєра обробки даних trainingPipeline, причому тренування виконується асинхронно для оптимізації ресурсоемності операції. Після завершення процесу навчання таймер зупиняється, дозволяючи зафіксувати точну тривалість виконання цієї операції.

```

// Вимірювання часу тренування
var stopwatch = System.Diagnostics.Stopwatch.StartNew();

// Перевірка наявності даних для тренування
if (mlContext.Data.CreateEnumerable<RawMachineData>(trainingData,
reuseRowObject: false).Any()) {
    trainedModel = await Task.Run(() => trainingPipeline.Fit(trainingData));
    stopwatch.Stop();

    ReportTBox.Text += $"Модель успішно натреновано за" +
        $" {stopwatch.Elapsed.TotalSeconds:N2} секунд.\r\n";
}

```

На рис. 3.9 наведено код, який виконує оцінку продуктивності навченої моделі, використовуючи тестовий набір даних.

```
// Оцінка моделі
var predictions = trainedModel.Transform(testData);
var metrics = mlContext.BinaryClassification.Evaluate(predictions, "Label");

// Основні метрики
ReportTextBox.Text += $"Точність: {metrics.Accuracy:P2}\r\n";
ReportTextBox.Text += $"Площа під ROC-кривою (AUC): {metrics.AreaUnderRocCurve:P2}\r\n";
ReportTextBox.Text += $"F1-міра: {metrics.F1Score:P2}\r\n";
```

Рисунок 3.9 – Оцінка продуктивності навченої моделі

На початку модель застосовується до тестових даних, створюючи передбачення, які аналізуються далі. Для обчислення якості моделі використовується метод оцінювання задачі двійкової класифікації, що надає різні метрики. Серед них відображаються точність класифікації, площа під ROC-кривою, яка вказує на загальну ефективність моделі, та F1-міра, що є комбінованим показником точності та повноти. Результати цих метрик записуються в текстове поле, забезпечуючи інформативне представлення якості роботи моделі для подальшого аналізу.

Фрагмент коду приведений на рис. 3.10 розраховує та виводить матрицю помилок для оцінки роботи моделі. Спочатку результати передбачень моделі перетворюються у список об'єктів, які містять як передбачені значення, так і реальні класи. Далі виконується ітерація через кожен запис, де порівнюються реальний стан та прогноз моделі. На основі цього підраховуються кількість істинно позитивних, істинно негативних, хибно позитивних і хибно негативних результатів.

```

//Матриця помилок
var scoredData = mlContext.Data.CreateEnumerable<FailurePrediction>(
    predictions, reuseRowObject: false).ToList();
int truePositives = 0;
int trueNegatives = 0;
int falsePositives = 0;
int falseNegatives = 0;
foreach (var prediction in scoredData) {
    bool actual = prediction.Label; // Реальний клас
    bool predicted = prediction.PredictedLabel; // Передбачений клас
    if (actual && predicted)
        truePositives++;
    else if (!actual && !predicted)
        trueNegatives++;
    else if (!actual && predicted)
        falsePositives++;
    else if (actual && !predicted)
        falseNegatives++;
}

ReportTextBox.Text += "\r\n=== Confusion Matrix ===\r\n";
ReportTextBox.Text += $"True Positives: {truePositives}\r\n";
ReportTextBox.Text += $"True Negatives: {trueNegatives}\r\n";
ReportTextBox.Text += $"False Positives: {falsePositives}\r\n";
ReportTextBox.Text += $"False Negatives: {falseNegatives}\r\n";

```

Рисунок 3.10 – Розрахунок та виведення матриці помилок

Зібрані дані про розподіл результатів моделі формують основу для побудови матриці помилок. Усі значення, які ілюструють продуктивність класифікації, записуються у текстове поле для подальшого аналізу. Такий підхід дозволяє зрозуміти, в яких аспектах модель працює добре, а де можуть бути слабкі місця, наприклад, надмірна кількість хибних позитивів чи негативів.

На рис. 3.11 наведено фрагмент коду реалізує функціонал для збереження натренованої моделі. Під час натискання кнопки перевіряється коректність введених даних, і якщо всі умови виконані, процес збереження розпочинається. Спершу генерується унікальний шлях до файлу моделі, який буде збережений у вигляді архіву з розширенням .zip у вказаній директорії. Локальний шлях до виконаного файлу визначається динамічно, щоб забезпечити правильне розташування файлу.

```

private void SaveBtn_Click(object sender, EventArgs e) {
    if (IsDataEnteringCorrect()) {
        //Зберігання моделі
        string pathName = @"teach\" + GenerateFileName() + ".zip";
        string localProj =
            System.IO.Path.GetDirectoryName(System.Reflection.Assembly.GetExecutingAssembly().Location);
        _ModelsProvider.InsertModels(ModelsNamesTBox.Text, pathName);
        mlContext.Model.Save(trainedModel, dataView.Schema, localProj + pathName);
        ClearAllData();
        _LogsProvider.InsertLogs(LoginForm.CurrentUser.UsersId,
            "Було навчено модель " +
            ModelsNamesTBox.Text, DateTime.Now);
        MessageBox.Show("Дані успішно збережено!");
    }
}

```

Рисунок 3.11 – Збереження натренованої моделі

Нова модель додається до бази даних моделей через об'єкт `_ModelsProvider`, а потім зберігається разом зі схемою даних за допомогою функцій ML.NET. Після цього всі тимчасові дані очищуються, а інформація про створення нової моделі записується до журналу через `_LogsProvider`, включаючи час і назву моделі. Завершальним кроком користувач отримує повідомлення про успішне збереження даних, що робить процес інтерактивним і зрозумілим.

3.5. Інтеграція моделі машинного навчання у програмне забезпечення моніторингу

Інтеграція моделі машинного навчання у програмне забезпечення моніторингу є важливим етапом для забезпечення повного циклу роботи системи: від збору даних і тренування моделі до її використання для прогнозування. Цей процес передбачає створення необхідних об'єктів та механізмів, які забезпечують безперервну взаємодію між інтерфейсом користувача, базою даних і моделлю машинного навчання. Вбудовані компоненти дозволяють не лише завантажувати та використовувати модель, а й зберігати журнали операцій та здійснювати валідацію даних.

У наведеному коді на рис. 3.12 створюються основні об'єкти, які забезпечують функціонування системи моніторингу. Клас `ValidationMy` використовується для перевірки коректності введених даних, запобігаючи помилкам на етапі підготовки інформації для прогнозування. Об'єкт `_SelectedModels` представляє поточну вибрану модель, яка буде

використовуватися для аналізу, тоді як список `_ModelsList` забезпечує доступ до всіх доступних моделей. `MLContext` є ядром роботи з ML.NET, яке створює середовище для завантаження, використання та прогнозування моделей. Для взаємодії з моделлю створюється `PredictionEngine`, що дозволяє виконувати прогнозування на основі введених даних.

```
private ValidationMy _Validation = new ValidationMy();
private Models _SelectedModels = new Models();
private MLContext context = new MLContext();
private PredictionEngine<RawMachineData, MachinePrediction> predictionEngine;

private ModelsProvider _ModelsProvider = new ModelsProvider();
private List<Models> _ModelsList = new List<Models>();
private bool _IsModelsLoad = false;
private LogsProvider _LogsProvider = new LogsProvider();
```

Рисунок 3.12 – Основні об’єкти для забезпечення функціонування системи

Компоненти `_ModelsProvider` і `_LogsProvider` відповідають за управління моделями та ведення журналу відповідно. Перший забезпечує доступ до збережених моделей і їх завантаження, тоді як другий записує всі ключові події, такі як вибір моделі чи виконання прогнозу, для аудиту та відстеження роботи системи. У результаті, цей код створює основу для інтеграції моделі машинного навчання в загальну архітектуру програмного забезпечення, забезпечуючи її ефективне використання та підтримку.

На рис. 3.13 зображено метод, який виконує завантаження раніше створеної моделі машинного навчання та забезпечує її підготовку до використання в програмному забезпеченні. Спочатку визначається шлях до файлу моделі, що базується на директорії запуску програми. Цей шлях формується шляхом доповнення базової директорії заданим значенням у змінній `FilePath`. Для роботи з моделлю ініціалізується об’єкт `DataViewSchema`, який забезпечує зберігання інформації про структуру моделі. Цей об’єкт використовується для інтерпретації схеми даних, необхідної для коректної роботи з моделлю.

```

private void LoadData(string FilePath) {
    string localProj = Application.StartupPath + FilePath;
    // Визначте DataViewSchema для конвеєра підготовки даних і навченої моделі
    DataViewSchema modelSchema;
    // Завантаження моделі
    ITransformer model = context.Model.Load(localProj, out modelSchema);
    //Створення механізму прогнозування
    predictionEngine =
        context.Model.CreatePredictionEngine<RawMachineData,
            MachinePrediction>(model);
}

```

Рисунок 3.13 – Завантаження створеної моделі та підготовка її до використання

Модель завантажується за допомогою context.Model.Load, що дозволяє відновити треновану модель із файлу. Це забезпечує її готовність до подальшого використання для аналізу та прогнозування. Після завантаження створюється механізм прогнозування у вигляді PredictionEngine, який дає змогу застосовувати модель для виконання прогнозів на основі даних у форматі RawMachineData. Така реалізація створює простий і зручний інтерфейс для інтеграції моделі машинного навчання у робочий процес програми. Вона також забезпечує безперебійну роботу з моделлю, спрощуючи її використання.

Метод на рис. 3.14 відповідає за симуляцію моніторингу роботи системи в реальному часі. Це досягається завдяки генерації даних і виконанню прогнозування на їх основі.

```

private void MonitoringTimer_Tick(object sender, EventArgs e) {
    // Створення об'єкта випадкових даних для моделювання
    var monData = MonitiringData();
    // Виведення згенерованих даних у TextBox
    MonitoringTBox.Invoke((MethodInvoker)((() => {
        var dataInfo = new StringBuilder();
        dataInfo.AppendLine($"Температура навколишнього середовища: {monData.AmbientTemperature:F2} °C");
        dataInfo.AppendLine($"Температура процесу: {monData.ProcessTemperature:F2} °C");
        dataInfo.AppendLine($"Швидкість обертання: {monData.RotationSpeed:F1} об/хв");
        dataInfo.AppendLine($"Крутний момент: {monData.Torque:F1} Нм");
        dataInfo.AppendLine($"Зношення інструменту: {monData.ToolWear} хв");
        // Виведення інформації в текстове поле
        MonitoringTBox.Text = dataInfo.ToString();
        // Прогнозування на основі згенерованого об'єкта randomData
        var prediction = predictionEngine.Predict(monData);
        var answer = new StringBuilder();
        answer.AppendLine("\r\n--- Прогнозування ---");
        answer.AppendLine($"Рекомендований клас: {(prediction.Prediction ? "Відмова" : "Норма")}");
        answer.AppendLine($"Ймовірність відмови: {prediction.Probability:P2}");
        // Додавання результату прогнозування до текстового поля
        MonitoringTBox.Text += answer.ToString();
    }));
}

```

Рисунок 3.14 – Симуляція даних та прогнозування

На кожному циклі таймера створюється об'єкт із випадковими параметрами, що імітують характеристики обладнання, такі як температура,

швидкість обертання, крутний момент та зношення інструменту. Згенеровані дані форматуються у вигляді текстової інформації та виводяться у відповідне текстове поле інтерфейсу для користувача. Далі з отриманих даних виконується прогнозування, де модель визначає, чи перебуває система в нормальному стані, чи є загроза відмови. Результат прогнозу разом із розрахованою ймовірністю відмови також додається до текстового поля, створюючи повну картину стану системи. Завдяки асинхронній обробці метод забезпечує інтерактивність і дозволяє користувачам в режимі реального часу бачити результати прогнозів, що полегшує моніторинг та аналіз роботи системи.

Розроблене програмне забезпечення може бути адаптоване для використання мережевого обладнання організації, зокрема комп'ютерній техніці, ББЖ чи комутаційному обладнанні, шляхом налаштування вхідних параметрів у конфігураторі. Наприклад, при заміні компонента, такого як кулер, користувач може вручну оновити відповідні дані, використовуючи інформацію з технічного паспорту. Усі параметри, що не передаються автоматично, також можуть бути задані вручну, що забезпечує більшу гнучкість та точність моделі. Такий підхід дозволяє застосовувати систему в умовах реальних мереж із різноманітним обладнанням та специфічними вимогами.

3.6. Тестування програмного забезпечення та оцінювання точності прогнозів

Процес тестування програмного забезпечення для прогнозування стану технічних систем є одним із ключових етапів його розробки та впровадження. Він дозволяє перевірити здатність моделі коректно ідентифікувати стан машини (нормальний або аварійний) на основі реальних даних, а також оцінити точність прогнозів у контексті задачі класифікації. Для цього було використано добре структурований датасет, що включає дані про робочі параметри обладнання та стан його працездатності.

Датасет складався з 624 записів про відмови обладнання (позитивні зразки) та 17,821 запису про нормальний стан (негативні зразки). Такий розподіл є типовим для задач діагностики, де позитивні зразки трапляються значно рідше,

ніж негативні. Це зумовлює необхідність ретельного балансування даних, оскільки дисбаланс може спричинити зміщення моделі на користь більш поширеного класу. Попередня обробка даних включала видалення дублікатів і пропусків, нормалізацію числових ознак і перевірку відповідності кожного запису заданій структурі.

Під час тестування моделі було виконано оцінку її точності, чутливості та специфічності, що дало змогу визначити здатність моделі як до виявлення відмов, так і до мінімізації помилкових тривог. Особлива увага приділялася аналізу роботи моделі з незбалансованими даними, оскільки неправильно налаштована модель могла б суттєво втратити ефективність у прогнозуванні рідкісних подій, таких як відмови. Подальші етапи тестування передбачали перевірку моделі на нових даних, відмінних від тих, що використовувалися для навчання, а також аналіз сценаріїв, які імітують типові умови експлуатації обладнання.

Наступні етапи тестування включали перевірку моделі на даних, відмінних від тих, на який відбувалося її навчання, а також аналіз сценаріїв, що імітують типові умови експлуатації технічного обладнання.

На рис. 3.15 відображено процес навчання моделі у розробленому програмному забезпеченні.

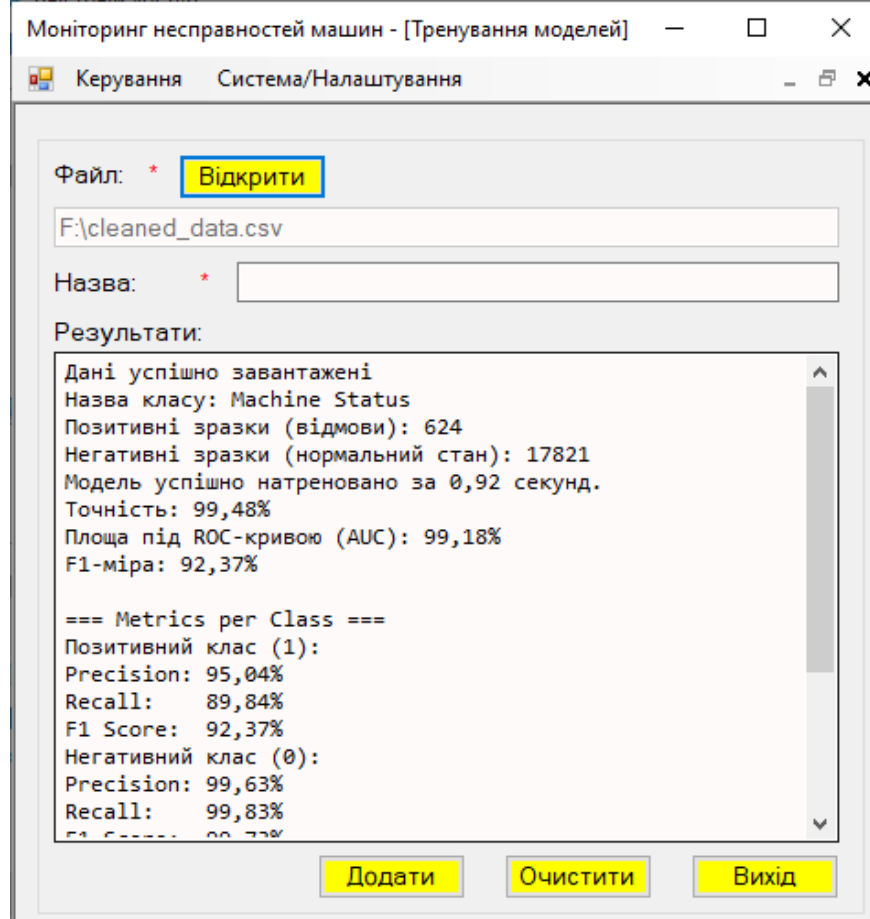


Рисунок 3.15 – Результат проведеного навчання моделі

Для детальної оцінки роботи моделі були розраховані ключові метрики для кожного класу окремо, що дозволяє глибше проаналізувати її продуктивність та ефективність у вирішенні задачі класифікації (рис. 3.16).

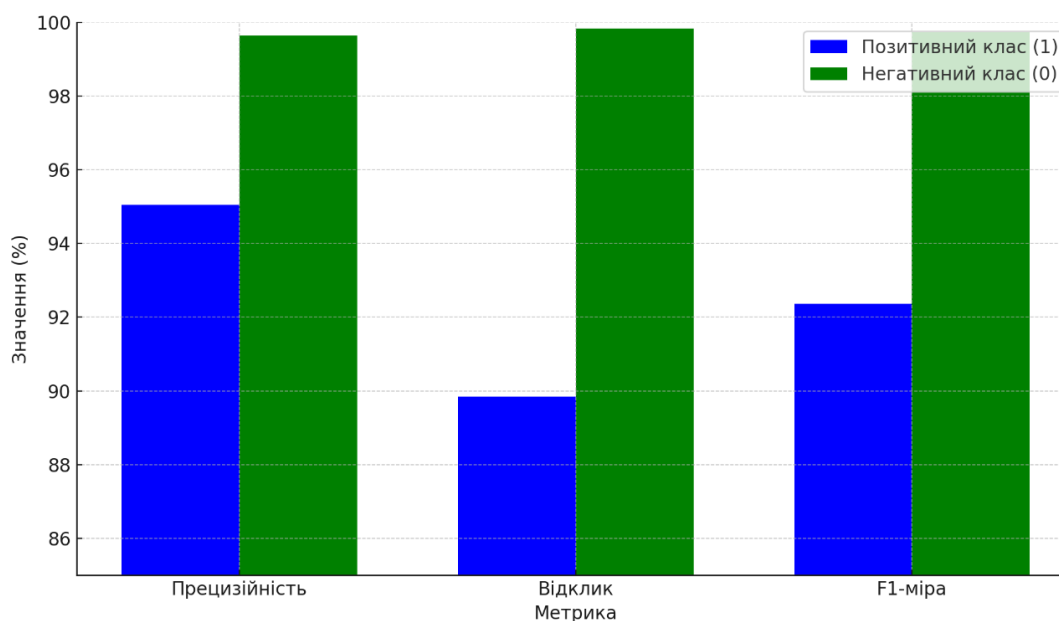


Рисунок 3.16 – Метрики по кожному класу (Прогнозування несправності)

Рис. 3.16 демонструє, що модель ефективно справляється з класифікацією обох класів, забезпечуючи високі показники прецизійності та відклику. Для позитивного класу (стан "відмова") модель досягає F1-міри 92,37%, що свідчить про добре збалансовану точність і здатність виявляти відмови. У випадку негативного класу (стан "норма") модель демонструє ще вищу ефективність, забезпечуючи F1-міру на рівні 99,73%. Це вказує на високу здатність системи уникати помилкових сигналів про відмову, що критично для запобігання необґрунтованих дій.

Для візуалізації роботи моделі було побудовано матрицю помилок, яка деталізує правильні та неправильні класифікації (рис. 3.17).

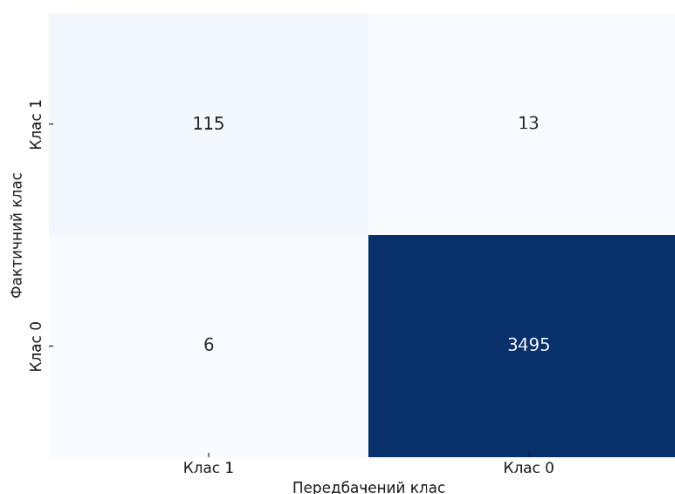


Рисунок 3.17 – Матриця помилок

Матриця помилок свідчить про високу точність моделі в класифікації обох класів. Модель успішно ідентифікувала 115 випадків фактичних відмов (істинно позитивні), що є важливим для запобігання потенційним аваріям. Лише 13 випадків залишилися не виявленими (хибно негативні), що вказує на необхідність додаткових покращень для зменшення пропущених відмов. У той же час хибно позитивних результатів (6 випадків) майже немає, що свідчить про здатність моделі уникати зайвих сповіщень. Висока кількість істинно негативних результатів (3495) підкреслює надійність моделі у визначенні нормального стану системи. Загалом матриця демонструє, що модель ефективно справляється зі своїм завданням, зберігаючи баланс між точністю та мінімізацією помилок.

На рис. 3.18 зображено інтерфейс програмного забезпечення моніторингу, який демонструє результати тестування моделі в реальному часі на першому

сценарії. Після введення вихідних параметрів, таких як температура навколишнього середовища, температура процесу, швидкість обертання, крутний момент і ступінь зношування інструменту, модель виконала прогнозування стану машини. У результаті система відобразила ключову інформацію про вхідні дані та рекомендації моделі щодо стану обладнання.

Вхідні дані:	
Модель: *	Модель 2
Температура навколишнього середовища: *	27,21 (°C)
Температура процесу: *	28,07 (°C)
Швидкість обертання: *	2712 (об./хв. (rpm))
Крутний момент: *	12,1 (Ньютон-метри)
Зношування інструменту: *	152

Прогнозувати

Моніторити

Результати прогнозування:	
Температура навколишнього середовища:	27,21 °C
Температура процесу:	27,21 °C
Швидкість обертання:	2712,0 об/хв
Крутний момент:	12,1 Нм
Зношення інструменту:	152 хв
--- Прогнозування ---	
Рекомендований клас:	Відмова
Ймовірність відмови:	94,06%

Рисунок 3.18 – Результати тестування моделі на 1-му сценарії

Результати прогнозування показують, що при заданих параметрах модель класифікувала стан як "відмова" із ймовірністю 94,06%. Це свідчить про високу впевненість алгоритму у зробленому прогнозі, що підтверджує її здатність ефективно виявляти потенційні несправності. Виведені значення також дозволяють оператору швидко оцінити критичні параметри, які могли призвести до такого прогнозу, наприклад, значне зношування інструменту (152 хвилини) або висока швидкість обертання (2712 об/хв).

Аналіз цього сценарію підтверджує ефективність розробленої системи у виявленні критичних станів обладнання. Завдяки інтерактивному інтерфейсу користувач отримує не лише результат прогнозу, а й доступ до детальної інформації про вхідні дані, що полегшує прийняття рішень щодо подальших дій для запобігання відмовам. Це підтверджує придатність моделі для впровадження в реальні умови моніторингу стану машин.

На рисунку 3.19 наведено результати тестування моделі на другому сценарії, який демонструє інший набір вхідних параметрів.

Моніторинг несправностей машин - [Моніторинг стану машин]

Керування Система/Налаштування

Вхідні дані:

Модель: *

Температура навколишнього середовища: * (°C)

Температура процесу: * (°C)

Швидкість обертання: * (об./хв. (rpm))

Крутний момент: * (Ньютон-метри)

Зношування інструменту: *

Температура навколишнього середовища: 24,90 °C
 Температура процесу: 24,90 °C
 Швидкість обертання: 2172,0 об/хв
 Крутний момент: 14,8 Нм
 Зношення інструменту: 34 хв

--- Прогнозування ---

Рекомендований клас: Норма

Ймовірність відмови: 11,08%

Рисунок 3.19 – Результати тестування моделі на 2-му сценарії

Модель, після обробки даних, класифікувала стан машини як "норма" з низькою ймовірністю відмови – лише 11,08%. Це свідчить про те, що при зазначених значеннях ключових параметрів, таких як температура навколишнього середовища (24,90 °C), температура процесу (25,74 °C), швидкість обертання (2172 об/хв), крутний момент (14,8 Нм) та зношування інструменту (34 хвилини), обладнання перебуває у стабільному стані.

Результат прогнозу показує впевненість моделі у відсутності загроз для нормального функціонування системи. Порівняно з першим сценарієм, у цьому випадку параметри значно менше наближаються до критичних значень, що підтверджує логічність висновків моделі. Наприклад, рівень зношування інструменту є низьким, а швидкість обертання знаходиться у допустимих межах.

Цей сценарій підкреслює здатність моделі точно визначати безпечний стан системи, що є важливим для запобігання непотрібним діям або зайвій тривозі з боку оператора. Висока точність прогнозу в умовах нормальних параметрів підтверджує надійність розробленої системи для моніторингу стану машин у режимі реального часу. Також це свідчить про збалансованість моделі в обробці як потенційно критичних, так і стабільних ситуацій.

3.7. Оцінка економічної доцільності впровадження запропонованого енергонезалежного рішень

Мета економічної частини роботи полягає у виконанні розрахунків для оцінки економічної доцільності впровадження енергонезалежної мережної інфраструктури на базі підприємства ТОВ «Net City». Це підприємство спеціалізується на управлінні та обслуговуванні корпоративних мереж, і розроблене рішення має забезпечити енергетичну стабільність, зменшення витрат на електроенергію та підвищення загальної ефективності інфраструктури. Економічна оцінка включає аналіз витрат на впровадження, експлуатацію та обслуговування системи, а також визначення потенційної вигоди для підприємства.

Основними етапами оцінки є аналіз існуючої інфраструктури підприємства та виявлення ключових проблем, пов'язаних із витратами на енергозабезпечення. Після цього виконуються розрахунки матеріальних витрат на впровадження енергонезалежної інфраструктури, включно з вартістю обладнання, інсталяційними роботами та обслуговуванням. Окремо оцінюються витрати на електроенергію в існуючій системі та прогнозуються економічні вигоди від її модернізації. Особлива увага приділяється обчисленню амортизаційних витрат, які забезпечують фінансовий облік довготривалої експлуатації обладнання.

У процесі аналізу розраховується загальна собівартість проекту, враховуючи накладні витрати, заробітну плату персоналу та інші статті витрат. Крім того, проводиться оцінка економічної ефективності проекту через показники окупності та рентабельності. Такий підхід дозволяє виявити реальний економічний ефект від впровадження енергонезалежного рішення та визначити, наскільки доцільним є цей проект для ТОВ «Net City». Результати аналізу також дадуть змогу підприємству спланувати подальшу оптимізацію своїх ресурсів для забезпечення довготривалої фінансової вигоди.

3.7.1 Визначення стадій техпроцесу для побудови енергонезалежної мережної інфраструктури організації

Успішне впровадження енергонезалежної мережної інфраструктури на підприємстві ТОВ «Net City» потребує ретельного визначення ключових стадій технологічного процесу. Це сприяє не лише систематизації необхідних операцій, але й дозволяє провести детальну оцінку трудозатрат та забезпечити оптимальний розподіл ресурсів. Завчасне структурування етапів робіт допомагає знизити ризики, пов'язані із затримками, та гарантує виконання проекту у встановлені терміни. Окрім цього, розподіл задач між членами команди дозволяє зосередитися на пріоритетних завданнях і уникнути дублювання зусиль. Визначення чіткої відповідальності кожного виконавця підвищує рівень координації та контроль виконання робіт. У процесі реалізації особливу увагу приділяють підготовчим та організаційним заходам, які є фундаментом для успішного впровадження проекту. Для досягнення мети були виділені основні стадії, які охоплюють весь цикл – від планування до впровадження та фінального тестування. У табл. 3.3 наведено середній час, необхідний для виконання кожної стадії, а також список виконавців, відповідальних за їх реалізацію.

Таблиця 3.3

Стадії техпроцесу для побудови енергонезалежної мережної інфраструктури

№	Назва операції (стадії)	Виконавець	Сер. час виконання, год.
1	Підготовчий етап	Керівник	8
2	Розробка проекту енергонезалежної інфраструктури	Інженер-проектувальник	40
3	Монтаж і налаштування відновлюваних джерел енергії	Технічний персонал	35
4	Інсталяція активного мережного обладнання	Технік	20
5	Інтеграція серверів та енергонезалежних модулів	Інженер систем	15
6	Тестування та оптимізація роботи системи	Технік	10
Разом			128

Перший етап – підготовчий – включає аналіз потреб підприємства, узгодження технічного завдання та попереднє планування. Основні витрати часу на цьому етапі припадають на керівника проекту, який координує загальний процес. Другий етап передбачає створення детального проекту інфраструктури, що враховує всі аспекти енергозабезпечення, резервування та мережної взаємодії. Це найбільш трудомістка стадія, що потребує значної участі інженерів.

Монтаж відновлюваних джерел енергії, таких як сонячні панелі чи акумуляторні системи, є ключовим етапом для забезпечення енергонезалежності мережі. Наступні стадії включають встановлення та налаштування активного мережного обладнання, інтеграцію серверів і енергетичних модулів. Завершальним етапом є тестування та оптимізація системи для забезпечення її безперебійної роботи.

Це структурування техпроцесу дозволяє не лише точно оцінити час виконання проекту, але й оптимізувати витрати підприємства на кожному етапі, забезпечуючи надійну реалізацію енергонезалежного рішення.

3.7.2 Визначення загальних витрат на оплату праці та відрахувань

Згідно з чинним законодавством України, зокрема Законом «Про оплату праці», заробітна плата працівників підприємств складається із основної та додаткової частин, які забезпечують справедливу винагороду за виконану роботу. Основна заробітна плата є фіксованою і залежить від посадових окладів, тарифних ставок або відрядних розцінок, а також від фактичного часу, витраченого працівником на виконання робіт. Додаткова частина включає премії, надбавки та інші заохочувальні виплати, які формуються з фонду матеріального стимулювання та залежать від результатів роботи підприємства.

Для проекту з впровадження енергонезалежної мережної інфраструктури на базі ТОВ «Net City» було розраховано загальні витрати на оплату праці команди, яка виконує роботи. Основою для таких розрахунків є погодинні тарифні ставки для кожної категорії працівників, які беруть участь у реалізації проекту. Формула для обчислення основної заробітної плати має вигляд:

$$Z_{\text{осн.}} = T_c \cdot K_r \quad (3.1)$$

де T_c – погодинна тарифна ставка (грн/год), K_r – кількість відпрацьованих годин.

$$Z_{o.k.} = 180 \cdot 8 = 1440 \text{ грн.}$$

$$Z_{o.i.} = 140 \cdot 40 = 5600 \text{ грн.}$$

$$Z_{o.l.} = 100 \cdot 35 = 3500 \text{ грн.}$$

$$Z_{o.t.} = 130 \cdot 45 = 5850 \text{ грн.}$$

Сумарна основна заробітна плата всіх працівників:

$$Z_{осн.} = Z_{o.k.} + Z_{o.i.} + Z_{o.l.} + Z_{o.t.} \quad (3.2)$$

$$Z_{осн.} = 1440 + 5600 + 3500 + 5850 = 16390 \text{ грн.}$$

Додаткова заробітна плата була визначена як 15% від основної заробітної плати кожного працівника. Формула для обчислення додаткової зарплати:

$$Z_{дод.} = Z_{осн.} \cdot K_{допл.} \quad (3.3)$$

де $K_{допл.} = 0,15$. Розрахунки для кожної категорії працівників виглядають наступним чином:

$$Z_{д.к.} = 1440 \cdot 0,15 = 216 \text{ грн.}$$

$$Z_{д.i.} = 5600 \cdot 0,15 = 840 \text{ грн.}$$

$$Z_{д.л.} = 3500 \cdot 0,15 = 525 \text{ грн.}$$

$$Z_{д.т.} = 5850 \cdot 0,15 = 877,5 \text{ грн.}$$

Сумарна додаткова заробітна плата становить:

$$Z_{дод.} = Z_{д.к.} + Z_{д.i.} + Z_{д.л.} + Z_{д.т.} \quad (3.4)$$

$$Z_{дод.} = 216 + 840 + 525 + 877,5 = 2458,5 \text{ грн.}$$

Загальні витрати на оплату праці розраховуються як сума основної та додаткової заробітної плати:

$$B_{o.п.} = Z_{осн.} + Z_{дод.} \quad (3.5)$$

$$B_{o.п.} = 16390 + 2458,5 = 18848,5 \text{ грн.}$$

Для обчислення витрат на соціальні відрахування використовується формула:

$$B_{з.с.} = B_{o.п.} \cdot 0,22, \quad (3.6)$$

$$B_{з.с.} = 18848,5 \cdot 0,22 = 4146,67 \text{ грн.}$$

Таким чином, загальні витрати на оплату праці з урахуванням усіх виплат і соціальних відрахувань становлять:

$$B_{\text{заг.}} = B_{\text{о.п.}} + B_{\text{з.с.}}, \quad (3.6)$$

$$B_{\text{заг.}} = 18848,5 + 4146,67 = 22995,17 \text{ грн.}$$

Усі ці розрахунки зведені в табл. 3.4, що деталізує витрати для кожної ролі, основну та додаткову заробітну плату, а також відрахування на фонд оплати праці.

Таблиця 3.4

Стадії техпроцесу для побудови енергонезалежної мережної інфраструктури

№	Категорія праців - ників	Основна заробітна плата, грн			Додаткова заробітна плата, грн	Нарахування на ФОП	Всього витрати
		Тар. ставка	К–сть відпра- цьованих год	Фактично нарахована з/пл			
1	Керівник	180	8	1440	216	-	-
2	Інженер	140	40	5600	840	-	-
3	Лаборант	100	35	3500	525	-	-
4	Технік	130	45	5850	877,5	-	-
Разом				16390	2458,5	4146,67	22995,17

Загальні витрати на оплату праці для проекту склали 22995,17 грн, включаючи основну та додаткову заробітну плату, а також соціальні відрахування.

3.7.3 Розрахунок матеріальних витрат

Матеріальні витрати для проекту побудови енергонезалежної мережної інфраструктури ТОВ «Net City» обчислюються на основі потреб у конкретних ресурсах та їх вартості. Для визначення витрат було використано формулу:

$$M_{\text{Ві}} = q_i \cdot p_i, \quad (3.7)$$

де q_i – кількість використаних одиниць матеріалу i -го виду, а p_i – ціна за одиницю цього матеріалу. Загальні матеріальні витрати визначаються як сума витрат на всі види матеріалів:

$$З_{М.В.} = \sum M_{Ві}, \quad (3.8)$$

де n – кількість видів матеріалів.

На основі проведених розрахунків було визначено необхідні матеріали та їх обсяги для виконання проекту (табл. 3.5).

Таблиця 3.5

Розрахунки матеріальних витрат

№	Найменування матеріальних ресурсів	Од. вим.	К- сть	Ціна за 1-цю, грн	Загальна сума витрат, грн
1	ZYXEL GPON OLT 8-port OLT1408A (1000 абонентів)	шт.	4	45 000	180 000
2	ZYXEL GPON OLT 4-port OLT1404A (500 абонентів)	шт.	4	35 000	140 000
3	GPON ONU для абонента (Netlink C-DATA FD600-521G-HZ660)	шт.	4	1 200	4800
4	S5860-20SQ, 24-Port Ethernet L3 Switch	шт.	2	25 000	50 000
5	Huawei NE20E серії у комплектації з модулями	шт.	2	85 000	170 000
6	CBS250-24T-4X	шт.	1	10 000	10 000
7	Huawei NE40E серії у комплектації з модулями	шт.	1	120 000	120 000
8	Cisco ASA5505 (мережевий екран)	шт.	1	35 000	35 000
9	SuperMicro SYS-620P-TR (16 ТБ SATA)	шт.	4	150 000	600 000
10	Asus Vivobook 15	шт.	5	20 000	100 000
11	ПК (ПК+Монітор) ARTLINE Business B27+B24F75plus-IPS	шт.	4	25 000	100 000
12	Оптичний кабель (16 волокон)	м	2000	20	40 000
13	Кабель живлення NYM 3x2,5 мм ²	м	300	18	5 400
14	Конектори RJ-45	шт.	200	1,5	300
15	Оптичні конектори SC/APC	шт.	100	25	2 500
16	Монтажні коробки	шт.	50	30	1 500
17	Сонячні панелі (модулі 500 Вт)	шт.	12	9200	110 400
18	Акумуляторні батареї LiFePO4 (100 А·год)	шт.	8	15 000	120 000
19	Кабель живлення NYM 3x2,5 мм ²	м	300	18	5400
20	Комунікаційний кабель UTP Cat6	м	400	15	6000
21	Конектори RJ-45 Cat6	шт.	100	1,5	150

22	Розетки RJ-45	шт.	40	50	2000
23	Комутаційний блок (24 порти)	шт.	1	4500	4500
Разом:					1807950

Розрахунки показують, що найбільші витрати припадають на сонячні панелі, акумуляторні батареї та інвертори, що є основою енергонезалежної інфраструктури. Вартість комунікаційних матеріалів, таких як кабелі, конектори та розетки, є значно нижчою, але також відіграє важливу роль у загальній структурі витрат.

Отримані результати свідчать про значну частку вартості, що пов'язана з енергетичними компонентами. Загальна сума матеріальних витрат становить 1807950 грн, що буде включено в загальний кошторис проекту. Ці дані дозволяють оцінити фінансову доцільність проекту та спланувати його реалізацію з урахуванням доступного бюджету підприємства.

3.7.4 Розрахунок витрат на електроенергію

Витрати на електроенергію для обладнання в рамках проекту побудови енергонезалежної мережної інфраструктури ТОВ «Net City» розраховуються з урахуванням потужності обладнання, часу його роботи та поточної вартості однієї кіловат-години електроенергії. Формула для розрахунку витрат на електроенергію має вигляд:

$$Z_e = W \cdot T \cdot S, \quad (3.9)$$

де:

W – необхідна потужність обладнання, кВт;

T – час роботи обладнання, годин;

S – вартість однієї кіловат-години електроенергії, грн.

Для розрахунків прийнято:

- потужність ПК (W) = 0,3 кВт;
- тривалість використання (T) = 150 годин;
- вартість електроенергії (S) = 6,00 грн./кВт·год.

Підставляючи значення у формулу, отримано:

$$T_b = 0,3 \cdot 150 \cdot 6,00 = 270,00 \text{ грн.}$$

Таким чином, витрати на електроенергію для використання ПК під час виконання кваліфікаційної роботи становлять 270,00 грн. Цей розрахунок дозволяє оцінити вплив експлуатації обладнання на загальні витрати, пов'язані з підготовкою роботи.

3.7.5 Визначення транспортних затрат

У рамках реалізації проекту побудови енергонезалежної мережної інфраструктури для ТОВ «Net City» важливо врахувати транспортні витрати, пов'язані з доставкою обладнання, матеріалів та супутніх ресурсів до місць їхнього використання. Транспортні витрати обчислюються як частка від загальної суми матеріальних витрат, що є типовою практикою при плануванні бюджету інфраструктурних проектів. Формула для розрахунку транспортних витрат має вигляд:

$$T_b = Z_{м.в.} \cdot K_{тр.}, \quad (3.10)$$

де:

T_b – транспортні витрати, грн;

$Z_{м.в.}$ – загальні матеріальні витрати, грн;

$K_{тр.}$ – коефіцієнт транспортних витрат, що зазвичай становить від 8% до 10%.

Підставляючи значення у формулу, отримано:

$$T_b = 1807950 \cdot 0,08 = 144636 \text{ грн.}$$

Таким чином, транспортні витрати для реалізації проекту становлять 144636 грн.

3.7.6 Розрахунок суми амортизаційних відрахувань

У процесі експлуатації обладнання в рамках проекту побудови енергонезалежної мережної інфраструктури для ТОВ «Net City» важливим є облік амортизаційних відрахувань. Амортизація дозволяє поступово відшкодувати вартість основних засобів, які використовуються для забезпечення

роботи інфраструктури. Це забезпечує фінансову можливість оновлення обладнання після закінчення його корисного використання. Для розрахунку амортизаційних відрахувань використовується формула:

$$A = \frac{B_B \cdot H_A \cdot T}{150\%}, \quad (3.11)$$

де:

A – амортизаційні відрахування за звітний період, грн;

B_B – балансова вартість основних засобів на початок звітного періоду, грн;

H_A – норма амортизації, виражена у вигляді частки від 1;

T – тривалість використання обладнання, годин.

Загальна сума амортизаційних відрахувань:

$$A = 36800 \cdot 0,04 \cdot 52/150 = 515,2 \text{ грн.}$$

Таким чином, амортизаційні відрахування для обладнання проекту становлять 515,2 грн за звітний період. Ці витрати є необхідними для підтримки фінансової стабільності підприємства та забезпечення оновлення інфраструктури в майбутньому.

3.7.7 Обчислення накладних витрат

Накладні витрати в рамках проекту побудови енергонезалежної мережної інфраструктури для ТОВ «Net City» включають витрати, пов'язані з адміністративним управлінням, організацією робочих процесів та створенням комфортних умов праці для співробітників. Вони також охоплюють комунальні послуги, офісне забезпечення та інші потреби, які супроводжують виконання проекту. Розмір накладних витрат залежить від особливостей підприємства та зазвичай становить від 20% до 60% від загальної суми витрат на оплату праці.

Формула для обчислення накладних витрат виглядає так:

$$H_B = B_{\text{о.п.}} \cdot K_{\text{нв}}, \quad (3.12)$$

де:

H_B – накладні витрати, грн;

$B_{\text{о.п.}}$ – загальні витрати на оплату праці, грн;

$K_{\text{нв}}$ – коефіцієнт накладних витрат (у межах 0,2–0,6).

Для розрахунку беремо коефіцієнт накладних витрат $K_{\text{нв}}=0,2(20\%)$, як оптимальний для поточних умов підприємства. Загальна сума витрат на оплату праці становить 13 377,54 грн. Підставляючи значення у формулу:

Підставляючи значення у формулу:

$$H_{\text{в}} = 18848,5 \cdot 0,2 = 3769,7 \text{ грн.}$$

Таким чином, накладні витрати для реалізації проекту становлять 3769,7 грн. Ця сума включає адміністративні витрати, забезпечення матеріалами загального користування, витрати на утримання приміщень і допоміжні ресурси. Обчислення накладних витрат дозволяє сформувати реалістичний бюджет проекту, враховуючи всі супутні витрати, необхідні для успішної реалізації поставлених завдань.

3.7.8 Складання кошторису витрат та визначення собівартості впровадження

Для повної оцінки фінансових витрат на реалізацію проекту побудови енергонезалежної мережної інфраструктури для ТОВ «Net City» було зведено всі проведені розрахунки у єдиний кошторис. Це дозволяє систематизувати витрати та визначити собівартість впровадження проекту.

Собівартість проекту обчислюється як сума всіх складових витрат, включаючи оплату праці, соціальні відрахування, матеріальні витрати, витрати на електроенергію, транспортні витрати, амортизаційні відрахування та накладні витрати. Розрахунок здійснюється за формулою:

$$C_{\text{в.}} = V_{\text{о.п.}} + V_{\text{с.з.}} + Z_{\text{м.в.}} + Z_{\text{е.}} + T_{\text{в.}} + A + H_{\text{в.}} \quad (3.13)$$

де: $C_{\text{в.}}$ – собівартість проекту, грн; $V_{\text{о.п.}}$ – витрати на оплату праці, грн; $V_{\text{с.з.}}$ – відрахування на соціальні заходи, грн; $Z_{\text{м.в.}}$ – матеріальні витрати, грн; $Z_{\text{е.}}$ – витрати на електроенергію, грн; $T_{\text{в.}}$ – транспортні витрати, грн; A – амортизаційні відрахування, грн; $H_{\text{в.}}$ – накладні витрати, грн.

Розрахунки представлені у табл. 3.7, яка систематизує всі витрати.

Таблиця 3.7

Розрахунок витрат на електроенергію

Зміст витрат	Сума, грн	В % до заг. суми
Витрати на оплату праці	18848,50	0,96
Відрахування на соціальні заходи	4146,67,00	0,21
Матеріальні витрати	1807950,00	92,36
Витрати на електроенергію	270,00	0,01
Транспортні витрати	144636,00	7,39
Амортизаційні відрахування	515,20	0,03
Накладні витрати	3769,70	0,19
Собівартість	1957140,90	100,00

Розрахунок собівартості виглядає наступним чином:

$$C_{\text{в.}} = 18848,50 + 4146,67,00 + 1807950,00 + 270,00 + 144636,00 + 515,20 + 3769,70 = 1957140,90 \text{ грн.}$$

Загальна сума собівартості проекту становить 1957140,90 грн. Вона враховує всі витрати, пов'язані з проектом, і дає змогу оцінити реальну фінансову вартість впровадження енергонезалежної мережної інфраструктури. Ці дані є основою для прийняття управлінських рішень та планування подальшого розвитку проекту.

3.7.9 Розрахунок ціни запропонованого рішення

Ціна запропонованого енергонезалежного рішення для ТОВ «Net City» визначається на основі собівартості проекту, рівня рентабельності, кількості замовлень, вартості супутніх послуг та ставки податку на додану вартість (ПДВ). Формула для обчислення ціни виглядає так:

$$Ц = \frac{C_{\text{в.}} \cdot (1 + P_{\text{рен.}}) \cdot K + B_{\text{н.і.}}}{K} \cdot (1 + \text{ПДВ}), \quad (4.14)$$

де: $C_{\text{в.}}$ – собівартість рішення, грн; $P_{\text{рен.}}$ – рівень рентабельності (30%); K – кількість замовлень (у даному випадку $K=1$);

$B_{\text{н.і.}}$ – вартість супутніх послуг, наприклад, носія інформації (прийнято 0 грн);

ПДВ – ставка податку на додану вартість (20%).

Згідно з розрахованою раніше собівартістю проекту, ціна рішення розраховується наступним чином:

$$Ц = 1957140,9 \cdot (1 + 0,3) \cdot (1 + 0,2) = 3053139,8 \text{ грн.}$$

Отже, ціна запропонованого енергонезалежного рішення для ТОВ «Net City» становить 3053139,8 грн.

3.7.10 Визначення економічної ефективності і терміну окупності капітальних вкладень

Економічна ефективність впровадження енергонезалежної мережної інфраструктури для ТОВ «Net City» оцінює, наскільки раціонально використовуються вкладені ресурси для досягнення бажаних результатів. Це співвідношення між отриманим прибутком та загальною собівартістю проекту. Воно дозволяє визначити доцільність проекту та його вплив на фінансовий стан підприємства.

Економічна ефективність проекту визначається за формулою:

$$E_p = \frac{\Pi}{C_v}, \quad (3.15)$$

де:

E_p – економічна ефективність;

Π – прибуток, грн;

C_v – собівартість проекту, грн.

Прибуток розраховується як різниця між ціною запропонованого рішення та його собівартістю:

$$\Pi = Ц - C_v, \quad (3.16)$$

де:

$Ц$ – ціна запропонованого рішення, грн.

Розрахунок прибутку:

$$\Pi = 3053139,8 - 1957140,9 = 1095998,9 \text{ грн.}$$

Розрахунок економічної ефективності:

$$E_p = 1095998,9 / 1957140,9 = 0,56$$

Термін окупності капітальних вкладень визначається як зворотне значення економічної ефективності:

$$T_{p.} = \frac{1}{E_p}, \quad (3.17)$$

$$T_{p.} = 1/0,56 = 1,8 \text{ р.}$$

Отже, проект характеризується високою економічною ефективністю з коефіцієнтом 0,56 і терміном окупності 1,8 роки, що підтверджує його фінансову доцільність та інвестиційну привабливість.

Таблиця 3.8

Економічні показники проекту

№ п/п	Показник	Значення
1	Собівартість, грн.	1957140,9
2	Плановий прибуток, грн.	1095998,9
3	Ціна, грн.	3053139,8
4	Економічна ефективність	0,56
5	Термін окупності, рік	1,8

Ці розрахунки свідчать про доцільність впровадження проекту, оскільки він швидко окупається та забезпечує значний прибуток для підприємства.

Висновок до третього розділу

У рамках даного розділу було проведено комплексну розробку програмного забезпечення для моніторингу енергонезалежної мережної інфраструктури ТОВ «Net City» та оцінку економічної доцільності його впровадження. Спочатку визначено функціональні та нефункціональні вимоги до програмного забезпечення, що забезпечило чітке формулювання цілей і критеріїв для його розробки. На етапі вибору алгоритмів машинного навчання проаналізовано три методи – швидке дерево, градієнтний бустинг дерев рішень і випадковий ліс. На основі порівняльного аналізу обрано алгоритм швидкого дерева, який демонструє оптимальне співвідношення точності та швидкості навчання.

Виконано попередню обробку даних, зокрема описано структуру датасету Factory Data та його ключові атрибути, що дозволило сформулювати якісний

тренувальний набір для моделі. Розроблено код для тренування моделі, який інтегрує алгоритм машинного навчання у програмне забезпечення, забезпечуючи прогнозування стану мережі в режимі реального часу. Результати тестування програмного забезпечення свідчать про його високу точність: для позитивного класу F1-міра склала 92,37%, а для негативного класу – 99,73%. Побудована матриця помилок показала мінімальну кількість хибно позитивних та хибно негативних прогнозів, що підтверджує ефективність моделі у реальних сценаріях.

Економічний аналіз показав, що розробка є фінансово вигідною: собівартість проекту становить 1 957 140,90 грн, плановий прибуток – 1 095 998,90 грн, а термін окупності – 1,8 року. Отримані результати підтверджують доцільність впровадження розробленого рішення для моніторингу мережі, що дозволяє підвищити ефективність управління ресурсами підприємства та забезпечити економічну стабільність за рахунок зменшення витрат на ремонт і обслуговування мережевого обладнання.

У ході виконання кваліфікаційної роботи було проведено комплексне дослідження актуальності теми побудови енергонезалежної мережної інфраструктури з оптоволоконним підключенням для абонентів. Визначено мету, об'єкт і предмет дослідження, а також сформульовано задачі, які успішно виконані в процесі роботи. Було детально проаналізовано сучасні інфраструктурні та технологічні рішення, зокрема оптоволоконні мережі, альтернативні джерела живлення та системи автоматизованого енергорозподілу, що дозволило визначити ключові вимоги до стабільної роботи мережі.

Досліджено особливості моделювання інфраструктури з абонентським підключенням, зокрема інтеграцію технологій GPON і DWDM, що забезпечують надійне і швидке з'єднання для користувачів у міських умовах. Розробка включала побудову логічної схеми мережі з урахуванням специфіки роботи в різних регіонах, таких як Коростень і Звягель, з використанням сучасних підходів до резервування та управління адресацією. На основі технічного завдання були створені функціональні сегменти мережі, які забезпечують гнучкість і масштабованість. Детально налаштовано параметри VLAN, DHCP, NAT і маршрутизації, що дозволяє оптимізувати використання ресурсів мережі та підвищити її стійкість до навантажень. Проведено тестування працездатності мережі в умовах реального навантаження, включаючи перевірку відмовостійкості та функціонування гіперканалів. Результати тестування підтвердили відповідність технічним вимогам і забезпечення стабільного функціонування навіть у разі відмови окремих елементів.

Розроблено програмне забезпечення для моніторингу мережі з використанням алгоритмів машинного навчання, яке дозволяє автоматизувати процеси аналізу та прогнозування стану інфраструктури. Проведено порівняльний аналіз алгоритмів, таких як швидке дерево, градієнтний бустинг дерев рішень і випадковий ліс, що дало змогу обрати швидке дерево як найоптимальніший метод з точки зору точності прогнозів і швидкості навчання. Виконано попередню обробку даних, включаючи очищення, нормалізацію та формування атрибутів для навчання моделі. Розроблено код для тренування

моделі, який інтегрується з програмним забезпеченням моніторингу, забезпечуючи можливість реального прогнозування несправностей мережі.

Результати тестування моделі підтвердили її високу ефективність. Для позитивного класу отримано Precision на рівні 95,04%, Recall – 89,84%, а F1-міра склала 92,37%. Для негативного класу Precision становить 99,63%, Recall – 99,83%, а F1-міра досягла 99,73%. Проведене тестування на двох сценаріях продемонструвало високу здатність системи ідентифікувати потенційні несправності з мінімальною кількістю помилкових спрацювань. Окрім того, інтеграція системи з інтерфейсом моніторингу спрощує роботу операторів, дозволяючи оперативно виявляти і реагувати на потенційні проблеми. Це забезпечує ефективну підтримку стабільності мережі та мінімізацію часу простоїв.

Оцінено економічну доцільність впровадження розробленого рішення. Собівартість проекту становить 1 957 140,90 грн, при цьому плановий прибуток складає 1 095 998,90 грн, а термін окупності проекту – 1,8 року. Це свідчить про високу економічну ефективність проекту та його привабливість для реалізації.

Зважаючи на все вище сказане, можна зробити висновок, що мета кваліфікаційної роботи успішно досягнута. Розроблене рішення відповідає технічним та економічним вимогам і має значний потенціал для практичного впровадження в умовах реального підприємства.

1. Матвійчук, Л. Ю., & Герасимчук, Б. П. Економічна доцільність використання альтернативних джерел енергії. Економічний форум. 2013. №4, с. 12-16.
2. Адаменко О. Альтернативні палива та інші нетрадиційні джерела енергії, посіб. Івано-Франківськ, 2000. – 270 с
3. Карпуков Л. М., Щекотихін В.О. Волоконно-оптичні лінії зв'язку: навчальний посібник для студентів вищих навчальних закладів– Запоріжжя: НУ «Запорізька політехніка», 2020. – 153 с.
4. Осадчук В.С., Осадчук О.В. Волоконно-оптичні системи передачі. Навчальний посібник — Вінниця: ВНТУ, 2005. – 225 с.
5. Що таке технологія GPON та як користуватися інтернетом без електрики? .URL: https://lambda.net.ua/index_ua.php?id=30 (дата звернення 31.10.2024).
6. Верхотуров Д. С. Джерело безперебійного живлення з мікропроцесорним керуванням. Хмельниц. нац. ун-т. – Хмельницький, 2023. – 74 с.
7. Педяш В.В. Телекомунікаційні системи та мережі наступного покоління: конспект лекцій. – Одеса: ОНАЗ ім. О.С. Попова, 2017.
8. Богацька О. О. Особливості сонячної енергетики та роботи сонячної панелі. Відроджена енергетика та енергоефективність у ХХІ столітті: матеріали ХХІІ міжнародної науково-практичної конференції (Київ, 20-21 травня 2021р.).–К.: Інтерсервіс, 2021.–1104 с.
9. Ребрик Б. Д.; Макаренко М. Є. Сонячні електростанції-переваги та недоліки. 2024. – 36с.
10. Вітрогенератори: принцип дії, типи, застосування, ефективність роботи URL: <https://alterair.ua/stati/vetrogeneratoryi/> (дата звернення 31.10.2024).
11. Бідник Д.С. Огляд та стан сучасних вітрогенераторів. 2019, – 30с.
12. Вітрогенератори: переваги та недоліки використання URL: <https://www.ueex.com.ua/presscenter/news/vitrova-energetika-perevagi-ta-nedoliki-zelenoi-ee-ii-prodazh-na/> (дата звернення 31.10.2024).

13. Про стабілізацію роботи об'єднаної енергетичної системи України.
URL: <https://mon.gov.ua/npa/pro-stabilizatsiiu-roboty-obiednanoi-enerhetychnoi-systemy-ukrainy> (дата звернення 31.10.2024).
14. Sagar Nishant. Powerline Communications Systems: Overview and Analysis. Rutgers University-Graduate School-New Brunswick. 2011. p. 28.
15. MENDELSON Galit. All you need to know about Power over Ethernet (PoE) and the IEEE 802.3 af Standard. Internet Citation,[Online] Jun, 2004, p.13.
16. AMODEI JR Aurelio; COSTA Luís Henrique MK; DUARTE Otto Carlos MB. Increasing the throughput of HomePNA. International Journal of Communication Systems, 2008, 21.7: pp. 695-711.
17. SHEN Lester S. Power over ethernet (poe) technical overview. Center for Energy and Environment, 2019, p.11.
18. EcoStruxure Platform | Schneider Electric Україна URL: <https://www.se.com/ua/uk/work/campaign/innovation/platform.jsp> (date of access: 31.10.2024).
19. Preguntas frecuentes de búsqueda. URL: <https://www.se.com/es/es/faqs/FAQ000223318/> (date of access: 31.10.2024).
20. ABB Ability™ Energy and Asset Manager. URL: <https://appsource.microsoft.com/es-es/product/web-apps/abb-3963258.abb-ability-energy-asset-manager?tab=Overview> (date of access: 31.10.2024).
21. Кулініч Т. В. Industrial management features in the context of abb ability systems application. Науковий вісник Полтавського університету економіки і торгівлі. Серія «Економічні науки», 2023, 4 (110): 36-42 с.
22. Honeywell Forge for Industrial. URL: <https://www.miningmonthly.com/partners/partner-content/1368342/honeywell-forge-industrial> (date of access: 31.10.2024).
23. The Power of Energy Management Software - Honeywell Forge URL: <https://www.honeywellforge.ai/us/en/learn/blog/power-of-energy-management-software> (date of access: 31.10.2024).
24. TOB «Net City» URL: <https://net-city.net/ua/> (дата звернення 31.10.2024).

25. ТОВ "НЕТ-СІТІ": вся інформація про компанію. URL: <https://vkursi.pro/card/rov-net-siti-42613217> (дата звернення: 31.10.2024).

26. Price, M. N., Dehal, P. S., & Arkin, A. P. FastTree: computing large minimum evolution trees with profiles instead of a distance matrix. *Molecular biology and evolution*. 2009. Vol. 26, No7, 1641-1650 p.

27. Zhang, Z., & Jung, C. GBDT-MO: gradient-boosted decision trees for multiple outputs. *IEEE transactions on neural networks and learning systems*. 2020. Vol. 32, No7. pp. 156-167.

28. Boateng, E. Y., Otoo, J., & Abaye, D. A. Basic tenets of classification algorithms K-nearest-neighbor, support vector machine, random forest and neural network: A review. *Journal of Data Analysis and Information Processing*. 2020. Vol. 8, No4. pp. 341-357.

29. Factory Data (Classification). URL: <https://www.kaggle.com/datasets/woonel/factory-data-classification> (date of access: 31.10.2024).

30. Окрушко В.С., Головня О. С. Методи та технології побудови енергонезалежної мережної інфраструктури організації // Тези доповідей VII Всеукраїнської науково-технічної конференції «Комп'ютерні технології: інновації, проблеми, рішення», м. Житомир, 02-03 грудня 2024 року – Житомир: Житомирська політехніка, 2024.

31. Окрушко В.С., Головня О. С. Розробка інтеграція оптоволоконних технологій та автоматизованих систем управління для побудови енергонезалежної мережної інфраструктури // Тези доповідей VII Всеукраїнської науково-технічної конференції «Комп'ютерні технології: інновації, проблеми, рішення», м. Житомир, 02-03 грудня 2024 року – Житомир: Житомирська політехніка, 2024.

32. Окрушко В.С., Головня О. С. Розробка програмного забезпечення для моніторингу несправностей у виробничих системах на основі машинного навчання // Тези доповідей VII Всеукраїнської науково-технічної конференції «Комп'ютерні технології: інновації, проблеми, рішення», м. Житомир, 02-03 грудня 2024 року – Житомир: Житомирська політехніка, 2024.

ДОДАТКИ

Лістинг 1. Код класу «ModelsForm»

```
using MachineFailureApp.AppCode;
using MachineFailureApp.Forms.Systems;
using MachineFailureApp.Providers;
using Microsoft.ML;
using Microsoft.ML.Data;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Globalization;
using System.IO;
using System.Diagnostics;
using Microsoft.ML.Transforms;

namespace MachineFailureApp.Forms.Systems {
    public partial class ModelsForm : Form {
        private MLContext mlContext;
        ITransformer trainedModel;
        private IDataView dataView;
        private string _Path = "";

        private int _selectedRowIndex = 0;
        private ValidationMy _Validation = new ValidationMy();
        private ModelsProvider _ModelsProvider = new ModelsProvider();
        private List<Models> _ModelsList = new List<Models>();
        private LogsProvider _LogsProvider = new LogsProvider();
        private bool _IsModelTrain = false;
        private Random _rand = new Random();

        public ModelsForm() {
            InitializeComponent();
            DataLoad();
        }

        private async void OpenBtn_Click(object sender, EventArgs e) {
            // Створення діалогового вікна для відкриття файлу
            OpenFileDialog openFileDialog = new OpenFileDialog();
            // Налаштування властивостей діалогового вікна
            openFileDialog.Filter = "Файли CSV (*.csv)|*.csv|Усі файли (*.*)|*.*";
            openFileDialog.FilterIndex = 2;
            openFileDialog.RestoreDirectory = true;

            // Відображення діалогового вікна та обробка результату
            if (openFileDialog.ShowDialog() == DialogResult.OK) {
                _Path = openFileDialog.FileName;
            }
        }
    }
}
```

```

FileNameTextBox.Text = openFileDialog.FileName;

// Створення контексту ML
mlContext = new MLContext(seed: 0);

// Завантаження даних як RawMachineData
dataView = mlContext.Data.LoadFromTextFile<RawMachineData>(
    _Path,
    separatorChar: ',',
    hasHeader: true);

// Конвертація DataView у перелік записів для підрахунку
var dataEnumerable = mlContext.Data.CreateEnumerable<RawMachineData>(dataView,
reuseRowObject: false);

// Підрахунок позитивних та негативних зразків
int positiveCount = dataEnumerable.Count(data => data.Label);
int negativeCount = dataEnumerable.Count(data => !data.Label);

// Додавання інформації у текстове поле
ReportTextBox.Text += "Дані успішно завантажені\r\n";
ReportTextBox.Text += $"Назва класу: Machine Status\r\n";
ReportTextBox.Text += $"Позитивні зразки (відмови): {positiveCount} \r\n";
ReportTextBox.Text += $"Негативні зразки (нормальний стан): {negativeCount} \r\n";
Application.DoEvents();

// Запуск асинхронного тренування моделі
await TrainModelAsync();
}
}

```

```

private async Task TrainModelAsync() {
    // Розділення даних на тренувальні та тестові
    var trainTestData = mlContext.Data.TrainTestSplit(dataView, testFraction: 0.2);
    var trainingData = trainTestData.TrainSet;
    var testData = trainTestData.TestSet;

    // Побудова та тренування моделі
    var trainingPipeline = mlContext.Transforms.ReplaceMissingValues("AmbientTemperature",
        replacementMode: MissingValueReplacingEstimator.ReplacementMode.Mean)
        .Append(mlContext.Transforms.ReplaceMissingValues("ProcessTemperature",
            replacementMode: MissingValueReplacingEstimator.ReplacementMode.Mean))
        .Append(mlContext.Transforms.ReplaceMissingValues("RotationSpeed",
            replacementMode: MissingValueReplacingEstimator.ReplacementMode.Mean))
        .Append(mlContext.Transforms.ReplaceMissingValues("Torque",
            replacementMode: MissingValueReplacingEstimator.ReplacementMode.Mean))
        .Append(mlContext.Transforms.ReplaceMissingValues("ToolWear",
            replacementMode: MissingValueReplacingEstimator.ReplacementMode.Mean))
        .Append(mlContext.Transforms.Concatenate("Features",
            nameof(RawMachineData.AmbientTemperature),
            nameof(RawMachineData.ProcessTemperature),
            nameof(RawMachineData.RotationSpeed),
            nameof(RawMachineData.Torque),

```

```

        nameof(RawMachineData.ToolWear)))
        .Append(mlContext.BinaryClassification.Trainers.FastTree(
            numberOfLeaves: 20,
            numberOfTrees: 100,
            minimumExampleCountPerLeaf: 10));

// Вимірювання часу тренування
var stopwatch = System.Diagnostics.Stopwatch.StartNew();

// Перевірка наявності даних для тренування
if (mlContext.Data.CreateEnumerable<RawMachineData>(trainingData,
    reuseRowObject: false).Any()) {
    trainedModel = await Task.Run(() => trainingPipeline.Fit(trainingData));
    stopwatch.Stop();

    ReportTextBox.Text += $"Модель успішно натреновано за" +
        $" {stopwatch.Elapsed.TotalSeconds:N2} секунд.\r\n";

    // Оцінка моделі
    var predictions = trainedModel.Transform(testData);
    var metrics = mlContext.BinaryClassification.Evaluate(predictions, "Label");

    // Основні метрики
    ReportTextBox.Text += $"Точність: {metrics.Accuracy:P2}\r\n";
    ReportTextBox.Text += $"Площа під ROC-кривою (AUC):
{metrics.AreaUnderRocCurve:P2}\r\n";
    ReportTextBox.Text += $"F1-міра: {metrics.F1Score:P2}\r\n";

    // Метрики по кожному класу
    ReportTextBox.Text += "\r\n=== Metrics per Class ===\r\n";
    ReportTextBox.Text += "Позитивний клас (1):\r\n";
    ReportTextBox.Text += $"Precision: {metrics.PositivePrecision:P2}\r\n";
    ReportTextBox.Text += $"Recall: {metrics.PositiveRecall:P2}\r\n";
    double positiveF1Score = 2 * metrics.PositivePrecision * metrics.PositiveRecall /
        (metrics.PositivePrecision + metrics.PositiveRecall);
    ReportTextBox.Text += $"F1 Score: {positiveF1Score:P2}\r\n";

    ReportTextBox.Text += "Негативний клас (0):\r\n";
    ReportTextBox.Text += $"Precision: {metrics.NegativePrecision:P2}\r\n";
    ReportTextBox.Text += $"Recall: {metrics.NegativeRecall:P2}\r\n";
    double negativeF1Score = 2 * metrics.NegativePrecision * metrics.NegativeRecall /
        (metrics.NegativePrecision + metrics.NegativeRecall);
    ReportTextBox.Text += $"F1 Score: {negativeF1Score:P2}\r\n";

    // Матриця помилок
    var scoredData = mlContext.Data.CreateEnumerable<FailurePrediction>(
        predictions, reuseRowObject: false).ToList();
    int truePositives = 0;
    int trueNegatives = 0;
    int falsePositives = 0;
    int falseNegatives = 0;
    foreach (var prediction in scoredData) {
        bool actual = prediction.Label; // Реальний клас
        bool predicted = prediction.PredictedLabel; // Передбачений клас
        if (actual && predicted)

```

```

        truePositives++;
    else if (!actual && !predicted)
        trueNegatives++;
    else if (!actual && predicted)
        falsePositives++;
    else if (actual && !predicted)
        falseNegatives++;
}

RaportTBox.Text += "\r\n=== Confusion Matrix ===\r\n";
RaportTBox.Text += $"True Positives: {truePositives}\r\n";
RaportTBox.Text += $"True Negatives: {trueNegatives}\r\n";
RaportTBox.Text += $"False Positives: {falsePositives}\r\n";
RaportTBox.Text += $"False Negatives: {falseNegatives}\r\n";

_IsModelTrain = true;
} else {
    RaportTBox.Text += "Недостатньо даних для тренування моделі.\r\n";
}
}

private void ModelsGridView_CellClick(object sender, DataGridViewCellEventArgs e) {
    if (e.ColumnIndex == 5 && ModelsGridView[0, e.RowIndex].Value.ToString() !=
        _ModelsList[0].Message) {
        if (MessageBox.Show("Ви дійсно хочете видалити цю модель?", "Видалити",
            MessageBoxButtons.YesNo) == DialogResult.Yes) {
            _ModelsProvider.DeleteModelsByModelsId(Convert.ToInt32(ModelsGridView[0,
                e.RowIndex].Value.ToString()));
            DataLoad();
        }
    }
}

private void SaveBtn_Click(object sender, EventArgs e) {
    if (IsDataEnteringCorrect()) {
        //Зберігання моделі
        string pathName = @"teach\" + GenerateFileName() + ".zip";
        string localProj =

System.IO.Path.GetDirectoryName(System.Reflection.Assembly.GetExecutingAssembly().Location);
        _ModelsProvider.InsertModels(ModelsNamesTBox.Text, pathName);
        mlContext.Model.Save(trainedModel, dataView.Schema, localProj + pathName);
        ClearAllData();
        _LogsProvider.InsertLogs(LoginForm.CurrentUser.UsersId,
            "Було навчено модель " +
            ModelsNamesTBox.Text, DateTime.Now);
        MessageBox.Show("Дані успішно збережено!");
    }
}

private void ClearBtn_Click(object sender, EventArgs e) {

```

```

        ClearAllData();
    }

    private void ExitBtn_Click(object sender, EventArgs e) {
        this.Close();
    }

    public string GenerateFileName() {
        DateTime now = DateTime.Now;
        string fileName = string.Format("{0}_{1}_{2}_{3}_{4}_{5}",
            now.Year, now.Month, now.Day, now.Hour, now.Minute, now.Second);

        return fileName;
    }

    private void ClearAllData() {
        _IsModelTrain = false;
        ModelsNamesTBox.Text = String.Empty;
        RaportTBox.Text = String.Empty;
        DataLoad();
    }

    private bool IsDataEnteringCorrect() {
        bool isCorrect = true;
        if (!_IsModelTrain) {
            MessageBox.Show("Неможливо зберегти дані. \r\nЩе не навчено модель!", "Увага!");
            isCorrect = false;
        }
        if (_Validation.IsDataEntering(ModelsNamesTBox.Text)) {
            ModelsNamesValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
        } else {
            ModelsNamesValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
            isCorrect = false;
        }
        return isCorrect;
    }

    private void DataLoad() {
        int firstRowIndex = 0;
        if (ModelsGridView.FirstDisplayedScrollingRowIndex > 0) {
            firstRowIndex = ModelsGridView.FirstDisplayedScrollingRowIndex;
        }
        try {
            _ModelsList = _ModelsProvider.GetAllModels();
            LoadDataInModelsGridView(_ModelsList);
            if (_selectedRowIndex == ModelsGridView.Rows.Count) {
                _selectedRowIndex = ModelsGridView.Rows.Count - 1;
            }
            if (_selectedRowIndex >= 0) {
                ModelsGridView.FirstDisplayedScrollingRowIndex = firstRowIndex;
                ModelsGridView.Rows[_selectedRowIndex].Selected = true;
            }
        } catch (Exception ex) {
            MessageBox.Show(ex.ToString());
        }
    }

```

```

}

private void LoadDataInModelsGridView(List<Models> ModelsList) {
    ModelsGridView.DataSource = null;
    ModelsGridView.Columns.Clear();
    ModelsGridView.AutoGenerateColumns = false;
    ModelsGridView.RowHeadersVisible = false;

    ModelsGridView.DataSource = ModelsList;

    if (ModelsList.Count > 0) {
        if (ModelsList[0].Message == NamesMy.NoDataNames.NoDataInModels) {
            DataGridViewColumn messageColumn = new DataGridViewTextBoxColumn();
            messageColumn.DataPropertyName = "Message";
            messageColumn.Width = ModelsGridView.Width - NamesMy.SizeOptins.MinusSizePanel;
            ModelsGridView.Columns.Add(messageColumn);
        } else {
            DataGridViewColumn DetailIdColumn = new DataGridViewTextBoxColumn();
            DetailIdColumn.DataPropertyName = "ModelsId";
            ModelsGridView.Columns.Add(DetailIdColumn);
            ModelsGridView.Columns[0].Visible = false;

            DataGridViewColumn numberColumn = new DataGridViewTextBoxColumn();
            numberColumn.HeaderText = "№ ";
            numberColumn.DataPropertyName = "Number";
            numberColumn.DefaultCellStyle.Alignment =
DataGridContentAlignment.MiddleCenter;
            numberColumn.Width = NamesMy.SizeOptins.NumberSize;
            ModelsGridView.Columns.Add(numberColumn);

            DataGridViewColumn ModelsNamesColumn = new DataGridViewTextBoxColumn();
            ModelsNamesColumn.HeaderText = "Назва моделі";
            ModelsNamesColumn.DataPropertyName = "ModelsName";
            ModelsNamesColumn.Width = 150;
            ModelsGridView.Columns.Add(ModelsNamesColumn);
            DataGridViewColumn CreateDateColumn = new DataGridViewTextBoxColumn();
            CreateDateColumn.HeaderText = "Дата створення";
            CreateDateColumn.DefaultCellStyle.Alignment =
DataGridContentAlignment.MiddleCenter;
            CreateDateColumn.DataPropertyName = "CreateDate";
            CreateDateColumn.Width = 150;
            ModelsGridView.Columns.Add(CreateDateColumn);

            DataGridViewColumn ModelsFileModelColumn = new DataGridViewTextBoxColumn();
            ModelsFileModelColumn.HeaderText = "Файл";
            ModelsFileModelColumn.DataPropertyName = "ModelsFileModel";
            ModelsFileModelColumn.Width = 220;
            ModelsGridView.Columns.Add(ModelsFileModelColumn);

            DataGridViewButtonColumn IsResidesBtn = new DataGridViewButtonColumn();
            IsResidesBtn.HeaderText = "Видалити";
            IsResidesBtn.Text = "Видалити";
            IsResidesBtn.UseColumnTextForButtonValue = true;
            IsResidesBtn.ToolTipText = "Видалити";
            IsResidesBtn.Width = NamesMy.SizeOptins.DeleteBtnSize;

```

```

        ModelsGridView.Columns.Add(IsResidesBtn);
    }
    for (int i = 0; i < ModelsGridView.Columns.Count; i++) {
        ModelsGridView.Columns[i].HeaderCell.Style.BackColor = Color.LightGray;
    }
}
}

}
}
public class FailurePrediction {
    public bool PredictedLabel { get; set; } // Передбачений клас
    public float Probability { get; set; } // Ймовірність передбачення
    public float Score { get; set; } // Оцінка передбачення
    public bool Label { get; set; } // Реальний клас
}
Лістинг 2. Код класу «MonitoringForm»
using MachineFailureApp.AppCode;
using MachineFailureApp.Forms.Systems;
using MachineFailureApp.Providers;
using Microsoft.ML;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Reflection;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace MachineFailureApp.Forms.Controls {
    public partial class MonitoringForm : Form {
        private ValidationMy _Validation = new ValidationMy();
        private Models _SelectedModels = new Models();
        private MLContext context = new MLContext();
        private PredictionEngine<RawMachineData, MachinePrediction> predictionEngine;

        private ModelsProvider _ModelsProvider = new ModelsProvider();
        private List<Models> _ModelsList = new List<Models>();
        private bool _IsModelsLoad = false;
        private LogsProvider _LogsProvider = new LogsProvider();

        private Random _Monitoring = new Random();

        public MonitoringForm() {
            InitializeComponent();
            LoadAllData();
        }
    }
}

```

```

private void PredictBtn_Click(object sender, EventArgs e) {
    // Перевіряємо, чи всі введені дані коректні та чи існує модель
    if (IsAllMachineDataCorrect() && IsModelExist()) {
        // Створюємо об'єкт RawMachineData з даними на основі введених значень
        RawMachineData testMachineData = new RawMachineData {
            AmbientTemperature = (float)Convert.ToDouble(AmbientTemperatureTBox.Text),
            ProcessTemperature = (float)Convert.ToDouble(AmbientTemperatureTBox.Text),
            RotationSpeed = (float)Convert.ToDouble(RotationSpeedTBox.Text),
            Torque = (float)Convert.ToDouble(TorqueTBox.Text),
            ToolWear = (int)Convert.ToInt32(ToolWearTBox.Text),
            Label = false // Це поле не використовується для прогнозування, але може знадобитися
            для сумісності
        };

        // Виведення введених даних у TextBox з підказками
        var dataInfo = new StringBuilder();
        dataInfo.AppendLine($"Температура навколишнього середовища:
{testMachineData.AmbientTemperature:F2} °C");
        dataInfo.AppendLine($"Температура процесу: {testMachineData.ProcessTemperature:F2}
°C");
        dataInfo.AppendLine($"Швидкість обертання: {testMachineData.RotationSpeed:F1}
об/хв");
        dataInfo.AppendLine($"Крутний момент: {testMachineData.Torque:F1} Нм");
        dataInfo.AppendLine($"Зношення інструменту: {testMachineData.ToolWear} хв");

        // Додаємо введену інформацію до текстового поля
        MonitoringTBox.Text = dataInfo.ToString();

        // Прогнозування на основі введених даних
        var prediction = predictionEngine.Predict(testMachineData);

        // Формуємо результат прогнозування
        var answer = new StringBuilder();
        answer.AppendLine("\r\n--- Прогнозування ---");
        answer.AppendLine($"ρ\r\nРекомендований клас: {(prediction.Prediction ? "Відмова" :
"Норма")}");
        answer.AppendLine($"ρ\r\nЙмовірність відмови: {prediction.Probability:P2}");

        // Додаємо результат прогнозування до текстового поля
        MonitoringTBox.Text += answer.ToString();
    }
}

private void GenBtn_Click(object sender, EventArgs e) {
    if (IsModelExist()) {
        if (MoniroringTimer.Enabled) {
            MoniroringTimer.Enabled = false;
            GenBtn.Text = "Моніторити";
            _LogsProvider.InsertLogs(LoginForm.CurrentUser.UsersId,
                "Було зупинено моніторинг випадкових даних для моделі " +
                ModelsCBox.Text, DateTime.Now);
        } else {
            MoniroringTimer.Enabled = true;
            GenBtn.Text = "Зупинити";
            _LogsProvider.InsertLogs(LoginForm.CurrentUser.UsersId,

```

```

        "Було запущено моніторинг випадкових даних для моделі " +
        ModelsCBox.Text, DateTime.Now);
    }
}

private void MoniroringTimer_Tick(object sender, EventArgs e) {
    // Створення об'єкта випадкових даних для моделювання
    var monData = MonitiringData();
    // Виведення згенерованих даних у TextBox
    MonitoringTBox.Invoke((MethodInvoker)() => {
        var dataInfo = new StringBuilder();
        dataInfo.AppendLine($"Температура навколишнього середовища:
{monData.AmbientTemperature:F2} °C");
        dataInfo.AppendLine($"Температура процесу: {monData.ProcessTemperature:F2} °C");
        dataInfo.AppendLine($"Швидкість обертання: {monData.RotationSpeed:F1} об/хв");
        dataInfo.AppendLine($"Крутний момент: {monData.Torque:F1} Нм");
        dataInfo.AppendLine($"Зношення інструменту: {monData.ToolWear} хв");
        // Виведення інформації в текстове поле
        MonitoringTBox.Text = dataInfo.ToString();
        // Прогнозування на основі згенерованого об'єкта randomData
        var prediction = predictionEngine.Predict(monData);
        var answer = new StringBuilder();
        answer.AppendLine("\r\n--- Прогнозування ---");
        answer.AppendLine($"ρ\r\nРекомендований клас: {(prediction.Prediction ? "Відмова" :
"Норма")}");
        answer.AppendLine($"ρ\r\nЙмовірність відмови: {prediction.Probability:P2}");
        // Додавання результату прогнозування до текстового поля
        MonitoringTBox.Text += answer.ToString();
    });
}

```

```

private void LoadAllData() {
    _ModelsList = _ModelsProvider.GetAllModels();
    ModelsCBox.DataSource = _ModelsList;
    ModelsCBox.ValueMember = "ModelsId";
    ModelsCBox.DisplayMember = "ModelsName";
    _IsModelsLoad = true;

    ModelsCBox_SelectedValueChanged(ModelsCBox, EventArgs.Empty);
}

```

```

private void ModelsCBox_SelectedValueChanged(object sender, EventArgs e) {
    if (_IsModelsLoad && IsModelExist()) {
        _SelectedModels = _ModelsProvider.SelectedModelsByModelsId(
            Convert.ToInt32(ModelsCBox.SelectedValue));
        LoadData(_SelectedModels.ModelsFileModel);
    }
}

```

```

private void LoadData(string FilePath) {
    string localProj = Application.StartupPath + FilePath;
}

```

```

// Визначте DataViewSchema для конвеєра підготовки даних і навченої моделі
DataViewSchema modelSchema;
// Завантаження моделі
ITransformer model = context.Model.Load(localProj, out modelSchema);
//Створення механізму прогнозування
predictionEngine =
    context.Model.CreatePredictionEngine<RawMachineData,
        MachinePrediction>(model);
}

// Метод для генерації випадкових вхідних даних
private RawMachineData MonitoringData() {
    float ambientTemp = (float)(_Monitoring.NextDouble() * 5 + 24.0); // Температура від 24 до
29 °C
    float processTemp = ambientTemp + (float)(_Monitoring.NextDouble() * 1); // Температура
процесу близька до Ambient T
    float rotationSpeed = (float)(_Monitoring.Next(1200, 3000)); // Швидкість обертання від 1200
до 3000 об/хв
    float torque = (float)(_Monitoring.NextDouble() * 60 + 4); // Крутний момент від 4 до 64 Нм
    int toolWear = _Monitoring.Next(0, 250); // Зношення інструменту від 0 до 250 хвилин

    // Повернення нового екземпляра RawMachineData
    return new RawMachineData {
        AmbientTemperature = ambientTemp,
        ProcessTemperature = processTemp,
        RotationSpeed = rotationSpeed,
        Torque = torque,
        ToolWear = toolWear,
        Label = false // Це поле не використовується для прогнозування, але може бути
необхідним для сумісності
    };
}

private bool IsAllMachineDataCorrect() {
    bool isCorrect = true;
    if (_Validation.IsDataConvertToDouble(AmbientTemperatureTBox.Text)) {
        AmbientTemperatureValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {
        AmbientTemperatureValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
    }
    if (_Validation.IsDataConvertToDouble(ProcessTemperatureTBox.Text)) {
        ProcessTemperatureValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {
        ProcessTemperatureValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
    }
    if (_Validation.IsDataConvertToDouble(RotationSpeedTBox.Text)) {
        RotationSpeedValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {
        RotationSpeedValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
    }
}

```

```

        if (_Validation.IsDataConvertToDouble(TorqueTBox.Text)) {
            TorqueValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
        } else {
            TorqueValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
            isCorrect = false;
        }
        if (_Validation.IsDataConvertToDouble(ToolWearTBox.Text)) {
            ToolWearValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
        } else {
            ToolWearValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
            isCorrect = false;
        }
        return isCorrect;
    }

    private bool IsModelExist() {
        bool isCorrect = true;
        if (Convert.ToInt32(ModelsCBox.SelectedValue) > 0) {
            ModelsValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
        } else {
            ModelsValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
            isCorrect = false;
        }
        return isCorrect;
    }

}
}

```

Лістинг 3. Код класу «RawMachineDataProvider»

```

using Microsoft.ML.Data;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace MachineFailureApp.Providers {
    internal class RawMachineDataProvider {
        private string _ConnString =
            System.Configuration.ConfigurationSettings.AppSettings["CONNECT"];

    }
}

```

// Клас для зберігання вхідних даних, які використовуватимуться для прогнозування та тренування моделі

```

public class RawMachineData {

    // Температура навколишнього середовища в градусах Цельсія
    // Значення для введення користувачем
    [LoadColumn(3)]

```

```

public float AmbientTemperature { get; set; }

// Температура процесу в градусах Цельсія
// Значення для введення користувачем
[LoadColumn(4)]
public float ProcessTemperature { get; set; }

// Швидкість обертання в оборотах за хвилину (rpm)
// Значення для введення користувачем
[LoadColumn(5)]
public float RotationSpeed { get; set; }

// Крутний момент в Ньютон-метрах (Nm)
// Значення для введення користувачем
[LoadColumn(6)]
public float Torque { get; set; }

// Зношування інструменту в хвилинали
// Значення для введення користувачем
[LoadColumn(7)]
public float ToolWear { get; set; }

// Стан машини: true (1) - відмова, false (0) - нормальний стан
// Це поле буде ігноруватися під час прогнозування, але використовується під час
тренування моделі
[LoadColumn(8)]
public bool Label { get; set; }
}

// Клас для зберігання результатів прогнозування, які повертає модель
public class MachinePrediction {

// Прогнозований стан машини: true - відмова, false - нормальний стан
// Це основний результат, який буде виведено користувачеві після прогнозування
[ColumnName("PredictedLabel")]
public bool Prediction { get; set; }

// Ймовірність передбаченого результату
// Відображає впевненість моделі у своєму прогнозі
public float Probability { get; set; }

// Оцінка моделі, яка вказує на ступінь відповідності вхідним даним
// Може використовуватися для додаткового аналізу, але є другорядним результатом
public float Score { get; set; }
}

```