Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

факультет прикладних інформаційних технологій та електроінженерії
(повна назва факультету)
кафедра радіотехнічних систем
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр
(назва освітнього ступеня)

на тему: **Автоматизована система синфазної класифікації рівнів мережевого навантаження**

Виконав: студент II курсу, групи РАм-61
спеціальності 172 Телекомунікації та радіотехніка
(шифр і назва спеціальності)

(підпис) Герасимчук Ю.Ю. (прізвище та ініціали)

Керівник (підпис) Хвостівська Л.В. (прізвище та ініціали)

Нормоконтроль (підпис) Хвостівська Л.В. (прізвище та ініціали)

Завідувач кафедри (підпис) Дунець В.Л. (прізвище та ініціали)

Рецензент (підпис) Дедів Л.Є. (прізвище та ініціали)

Тернопіль
2025

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет прикладних інформаційних технологій та електроінженерії
(повна назва факультету)

Кафедра радіотехнічних систем
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Дунець В.Л.
(підпис) (прізвище та ініціали)
« 26 » листопада 2025 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістр
(назва освітнього ступеня)

за спеціальністю 172 Телекомунікації та радіотехніка
(шифр і назва спеціальності)

студенту Герасимчук Юрій Юрійович
(прізвище, ім'я, по батькові)

1. Тема роботи Автоматизована система синфазної класифікації рівнів мережевого навантаження

Керівник роботи Хвостівська Лілія Володимирівна, к.т.н., доц.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 26 » листопада 2025 року № 4/7-1038 .

2. Термін подання студентом завершеної роботи

3. Вихідні дані до роботи : Об'єкт дослідження: процес формування та зміни рівнів навантаження комп'ютерних мереж у часі. Предмет дослідження: математична модель, метод та алгоритм обробки ПКВП для бінарної та багатокласової класифікації рівнів навантаження

4. Зміст роботи (перелік питань, які потрібно розробити)
РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ КЛАСИФІКАЦІЇ МЕРЕЖЕВОГО НАВАНТАЖЕННЯ
РОЗДІЛ 2. МАТЕМАТИЧНА МОДЕЛЬ ТА МЕТОД СИНФАЗНОЇ КЛАСИФІКАЦІЇ РІВНІВ МЕРЕЖЕВОГО НАВАНТАЖЕННЯ
РОЗДІЛ 3. РОЗРОБКА АВТОМАТИЗОВАНОЇ СИСТЕМИ У ВИГЛЯДІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СИНФАЗНОЇ КЛАСИФІКАЦІЇ РІВНІВ МЕРЕЖ. НАВАНТАЖЕННЯ

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1. Актуальність роботи
2. Математична модель навантаження трафіку
3. Метод синфазної класифікації рівнів мережевого навантаження
4. Алгоритм синфазної класифікації рівнів мережевого навантаження
5. Автоматизована система синфазної класифікації рівнів мережевого навантаження
6. Результати синфазної класифікації рівнів мережевого навантаження
7. Наукова новизна
8. Публікації за напрямом дослідження
9. Загальні висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці та	Окіпний І.Б., зав. каф. МТ		
безпека в надзвичайних	Стручок В.С., ст. викл. каф. ОХ		
ситуаціях			

7. Дата видачі завдання 03.09.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	**Примітка**
1	Отримання завдання на виконання роботи	03.09.2025	
2	Аналіз отриманого завдання	13.09.2025	
3	Виконання розділу 1	23.09.2025	
4	Виконання розділу 2	10.10.2025	
5	Виконання розділу 3	25.10.2025	
6	Виконання розділу 4	15.11.2025	
7	Оформлення пояснювальної записки	20.11.2025	
8	Оформлення презентаційного матеріалу	26.11.205	
9	Перевірка кваліфікаційної роботи на антиплагіат	28.11.2025	
10	Попередній захист кваліфікаційної роботи	09.12.2025	
11	Захист кваліфікаційної роботи	23.12.2025	

Студент _____ Герасимчук Ю.Ю.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Хвостівська Л.В.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Автоматизована система синфазної класифікації рівнів мережевого навантаження» // Кваліфікаційна робота // Герасимчук Юрій Юрійович // ТНТУ, факультет прикладних інформаційних технологій та електроінженерії, група РАм-61 // Тернопіль, 2025 // с. – 84, рис. – 21, табл. – 1, додат. – 3, бібліогр. – 27.

Ключові слова: мережеве навантаження, синфазний метод, періодично-корельований випадковий процес, критерій Неймана–Пірсона, бінарна класифікація, багаторівнева класифікація, MATLAB, автоматизована система моніторингу.

В роботі досліджено процеси формування та зміни мережевого навантаження у комп'ютерних мережах, проаналізовано існуючі методи класифікації трафіку та визначено їхні обмеження. На основі апарату періодично-корельованих випадкових процесів розроблено математичну модель та метод синфазної класифікації рівнів навантаження, що включають побудову енергетичних компонент, емпіричних розподілів та застосування критерію Неймана–Пірсона для визначення оптимальних порогів.

Запропоновано алгоритм бінарної та багатокласової класифікації станів мережі (мінімальний, нормальний, критичний). Розроблено програмне забезпечення у середовищі MATLAB, яке реалізує математичний апарат, здійснює синфазну обробку даних, автоматичне визначення порогів класифікації, контроль ймовірностей хибних рішень та візуалізацію результатів.

Проведене тестування підтвердило ефективність системи: отримано достовірну класифікацію станів трафіку, визначено частки перебування мережі у різних режимах, встановлено порогові рівні для мінімального, нормального та критичного навантаження. Розроблена система забезпечує адаптивність, стійкість до флуктуацій та можливість роботи в режимі реального часу.

ANNOTATION

Theme of qualification work: «Automated System for Synphase Classification of Network Load Levels» // Herasumchyk Yu. // Ternopil Ivan Puluj National Technical University, Faculty of Applied Information Technologies and Electrical Engineering group RAm-61 // Ternopil, 2025 // p. – 84, fig. – 21, tab. - 1, add. – 3, bibliography -27.

Keywords: network load, synphase method, periodically correlated random process, Neyman–Pearson criterion, binary classification, multi-level classification, MATLAB, automated monitoring system.

The work investigates the processes of formation and variation of network load in computer networks and analyzes existing traffic classification approaches along with their limitations. A mathematical model and a synphase classification method based on periodically correlated random processes are developed. The method includes construction of energy components, empirical distributions and the use of the Neyman–Pearson criterion to determine optimal classification thresholds.

An algorithm for binary and multi-level classification of network states (minimal, normal, critical) is proposed. Software implementation in MATLAB has been developed, which performs synphase processing, automatic threshold detection, control of false-decision probabilities and visualization of the obtained results.

Experimental testing confirmed the efficiency of the system. The software provides reliable identification of traffic states, determines the proportions of time spent in each load level and establishes threshold values for minimal, normal and critical network conditions. The developed automated system ensures adaptability, robustness to fluctuations and the ability to operate in real-time monitoring environments.

ЗМІСТ

ВСТУП

Актуальність роботи

Сучасні комп’ютерні мережі характеризуються високою складністю, масштабністю та динамічністю процесів обміну даними. Підвищення інтенсивності трафіку, зміна характеру навантажень і необхідність підтримання якості обслуговування (QoS) зумовлюють потребу у створенні ефективних засобів контролю станів мережі. Одним із ключових завдань при цьому є класифікація рівнів мережевого навантаження – визначення поточного стану системи для своєчасного реагування на перевантаження та запобігання деградації її роботи.

Існує низка підходів до аналізу та класифікації мережевого трафіку:

– Статистичні методи базуються на аналізі інтенсивності, дисперсії та коефіцієнтів варіації потоків даних. Вони прості у реалізації, проте малоефективні за умов нестаціонарності або наявності шумів.

– Спектральні методи орієнтовані на дослідження енергетичного розподілу частотних компонент трафіку. Дозволяють виявляти зростання високочастотної активності при перевантаженнях, однак чутливі до короткочасних флуктуацій.

– Фрактальні та мультифрактальні підходи враховують самоподібність і довготривалу кореляцію у трафіку, що дає змогу описувати довгострокові закономірності мережевих процесів. Недоліком є складність реалізації в адаптивних системах реального часу.

– Методи машинного навчання використовують алгоритми класифікації та нейронні мережі (CNN, LSTM) для розпізнавання складних багатовимірних залежностей. Забезпечують високу точність, але потребують значних обчислювальних ресурсів і великих навчальних вибірок.

– Марковські моделі описують послідовність переходів між станами навантаження з урахуванням ймовірностей зміни режимів роботи мережі. Дають змогу прогнозувати майбутній стан системи, але вимагають ретельного налаштування параметрів.

– Адаптивні ансамблеві методи поєднують декілька класифікаторів, що оцінюють різні аспекти трафіку (статистичні, спектральні, фазові), підвищуючи стійкість до шумів та нестаціонарності, хоча й ускладнюють реалізацію системи.

Попри значний розвиток зазначених підходів, їх спільним недоліком залишається недостатня адаптивність до нестаціонарних процесів **і** низька точність у зашумлених середовищах, де характеристики трафіку змінюються у часі. Для адекватного опису таких процесів необхідно враховувати як періодичну структуру навантаження, так і стохастичні відхилення, що потребує застосування більш гнучкого математичного апарату.

Перспективним у цьому контексті є застосування моделі періодично-корельованого випадкового процесу (ПКВП), яка дає змогу описувати трафік, що поєднує регулярні циклічні зміни та стохастичні складові. На основі ПКВП формуються методи обробки сигналів, що забезпечують виявлення інформативних ознак для класифікації станів мережі. Одним із таких є синфазний метод обробки/аналізу, який дозволяє визначати енергетичні характеристики сигналу, узгоджені за фазою, та формувати стійкі до шуму ознаки.

Для прийняття рішень щодо поточного стану мережі доцільно використовувати статистичні критерії у поєднанні з пороговими пристроями, що забезпечують контроль ймовірностей помилкових класифікацій. Такий підхід дозволяє реалізувати формалізовану систему прийняття рішень з передбачуваними характеристиками точності.

Враховуючи різноманітність режимів роботи мережі, важливо забезпечити як бінарну класифікацію (розпізнавання нормального або аномального стану), так і багатокласову класифікацію (розмежування рівнів навантаження: мінімальний, середній, критичний). Це підвищує інформативність оцінювання та дозволяє адаптивно реагувати на зміну умов функціонування мережі.

Отже, поєднання апарату ПКВП, синфазної обробки та статистичних критеріїв прийняття рішень створює науково обґрунтовану основу для розроблення автоматизованої системи синфазної класифікації рівнів мережевого навантаження,

здатної забезпечити достовірність, стійкість та адаптивність класифікації у реальному часі.

Метою дослідження є розроблення автоматизованої системи синфазної класифікації рівнів мережевого навантаження, що базується на моделі періодично-корельованого випадкового процесу, методах синфазного аналізу та статистичних критеріях прийняття рішень.

Задачі дослідження:

1. Проаналізувати існуючі методи класифікації мережевого навантаження та визначити їхні обмеження.
2. Побудувати математичну модель трафіку на основі апарату періодично-корельованих випадкових процесів.
3. Розробити методи обробки сигналів навантаження трафіку з використанням синфазної обробки.
4. Сформулювати статистичні критерії прийняття рішень і визначити порогові умови класифікації.
5. Реалізувати методи та алгоритми бінарної та багатокласової класифікації рівнів навантаження.
6. Створити програмний модуль автоматизованої системи та оцінити його ефективність.

Об’єкт дослідження: процес формування та зміни рівнів навантаження комп’ютерних мереж у часі.

Предмет дослідження: математична модель, метод та алгоритм обробки ПКВП для бінарної та багатокласової класифікації рівнів мережевого навантаження.

Методи дослідження: методи теорії випадкових процесів, спектрального та кореляційного аналізу, математичної статистики, перевірки статистичних гіпотез, а також методи чисельного моделювання у середовищі MATLAB.

Наукова новизна:

– Обґрунтовано доцільність використання апарату періодично-корельованих випадкових процесів у поєднанні зі статистичними критеріями прийняття рішень для класифікації рівнів мережевого навантаження.

– Запропоновано концепцію автоматизованої системи синфазної класифікації, у якій реалізовано принципи бінарної та багатокласової класифікації станів, що забезпечує адаптивне, стійке й достовірне визначення режимів роботи комп’ютерних мереж.

РОЗДІЛ 1
АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ КЛАСИФІКАЦІЇ МЕРЕЖЕВОГО НАВАНТАЖЕННЯ

1.1 Особливості процесів формування навантаження у комп’ютерних мережах

Процес формування навантаження у комп’ютерних мережах є складним багатофакторним явищем, що зумовлюється взаємодією апаратних, програмних, топологічних та поведінкових компонент системи.

Навантаження характеризує інтенсивність потоків даних, які проходять через вузли мережі, і визначається як функція часу, тобто є динамічним процесом. Його опис потребує урахування як закономірних періодичних змін (циклів активності), так і випадкових флуктуацій, що пов’язані з особливостями користувацьких дій, протоколів обміну та технічного стану мережевих елементів.

1.1 Джерела формування навантаження

Основними джерелами формування трафіку в комп’ютерних мережах є:

– користувацькі запити до серверів і сервісів (HTTP, FTP, DNS, SMTP, VoIP тощо);

– фоновий трафік, зумовлений системними процесами, оновленням програм, резервним копіюванням або синхронізацією даних;

– автоматизовані додатки та IoT, які генерують короткі пакети даних з високою частотою;

– службовий трафік мережевих протоколів, що забезпечує маршрутизацію, моніторинг і керування потоками;

– аномальний або атакувальний трафік – результат дії шкідливих програм, ботнетів чи DDoS-атак.

Кожен тип джерела має власну часову структуру, інтенсивність і характер зміни у часі, тому сукупний трафік являє собою суму детермінованих і випадкових складових, які взаємодіють у межах загальної мережевої системи.

1.1.2 Характеристика мережевого навантаження

Мережеве навантаження описується низкою статистичних та енергетичних характеристик, серед яких найважливішими є:

– інтенсивність трафіку (λ) — середня кількість пакетів, що надходять до вузла за одиницю часу;

– дисперсія потоку — відображає варіативність трафіку;

– автокореляційна функція ($R(\tau)$), яка визначає наявність зв’язків між значеннями навантаження на різних часових інтервалах;

– спектральна густина потужності, що характеризує частотну структуру сигналу навантаження;

– коефіцієнт самоподібності (індекс Херста H) — показує наявність довготривалих залежностей.

Для сучасних мереж (зокрема Ethernet, Wi-Fi, LTE, 5G, оптичних транспортних систем) характерним є високий ступінь нестаціонарності процесу. Значення інтенсивності трафіку $\lambda(t)$ змінюється в часі не лише за рахунок випадкових подій, а й унаслідок добових циклів активності користувачів: у денний час відбуваються пікові навантаження, вночі — зниження інтенсивності, а в ранкові години — різкі стрибки при відновленні роботи систем.

1.1.3 Циклічність і стохастичність трафіку

У структурі мережевого трафіку спостерігається поєднання регулярності та хаотичності. Регулярність зумовлена повторюваними добовими, тижневими або сезонними патернами, що відповідають поведінці користувачів і режимам роботи організацій. Стохастична складова, навпаки, виникає через випадкові події — збої в каналах, коливання пропускної здатності, колективні оновлення програмного забезпечення, сплески внаслідок кібератак тощо.

Такі властивості зумовлюють складну структуру часових рядів трафіку, які не можна описати простою стаціонарною моделлю. У багатьох дослідженнях підтверджено, що мережеве навантаження має фрактальну природу і відображає довготривалу кореляцію, що притаманна системам із пам’яттю.

Тому для адекватного опису трафіку застосовуються ПКВП, які враховують як періодичні цикли, так і стохастичні відхилення. Такий підхід дозволяє моделювати сигнал у вигляді суперпозиції детермінованої складової $S(t)$, що повторюється з певним періодом, та випадкової складової $X(t)$, яка описує шумові або короткочасні флуктуації. Саме така форма є найпридатнішою для аналізу та подальшої класифікації станів системи.

1.1.4 Класифікаційні стани мережевого навантаження

Залежно від рівня використання ресурсів каналу або пропускної здатності мережі, виділяють три основні стани:

– Нормальний стан (A_2) — навантаження перебуває в межах допустимих значень, забезпечується гарантована якість обслуговування (QoS).

– Підвищене або граничне навантаження (A_3) — система наближається до насичення, зростає час затримки пакетів, спостерігаються коливання затримок і втрата пакетів.

– Мінімальне навантаження (A_1) — низька активність користувачів або часткове відключення вузлів, коли ресурси мережі використовуються неефективно.

Класифікація між цими станами дозволяє своєчасно реагувати на перевантаження або оптимізувати розподіл ресурсів. Для цього необхідно не лише фіксувати зміни середнього рівня навантаження, а й враховувати енергетичні та фазові характеристики сигналу трафіку, що відображають його внутрішню структуру.

1.1.5 Необхідність адаптивної обробки

Особливістю сучасних телекомунікаційних мереж є висока мінливість середовища передачі даних. Традиційні методи аналізу (зокрема статистичні або спектральні) не здатні повністю описати поведінку таких процесів, оскільки:

– зміна користувацької активності призводить до зміщення середнього рівня трафіку;

– поява нових сервісів (стримінгові платформи, хмарні додатки, штучний інтелект) суттєво впливає на частотний спектр навантаження;

– мережеві атаки або збої можуть раптово змінювати форму сигналу, створюючи аномальні піки енергії.

Це вимагає використання методів, здатних адаптивно оцінювати часово-частотні властивості трафіку, виявляти зміни у фазових співвідношеннях гармонічних компонент та визначати статистично обґрунтовані порогові рівні для класифікації станів системи.

1.1.6 Узагальнення

Отже, процес формування навантаження у комп'ютерних мережах має гібридну природу – поєднує періодичну структуру, стохастичні флуктуації та кореляційні зв'язки, що змінюються у часі.

Його моделювання потребує врахування періодичності, самоподібності та нестаціонарності, а для адекватної класифікації необхідно використовувати енергетичні та синфазні ознаки, які забезпечують підвищену стійкість до шумів.

Таким чином, для ефективного контролю та управління станами комп'ютерних мереж доцільно створювати адаптивні автоматизовані системи, що базуються на синфазному аналізі сигналів трафіку та статистичних критеріях прийняття рішень. Ці принципи закладають основу для подальшої побудови математичної моделі та методу синфазної класифікації, розглянутого в наступному розділі.

1.2 Методи класифікації мережевого трафіку

Класифікація мережевого навантаження є одним із ключових завдань сучасного інтелектуального моніторингу мереж. Вона полягає у визначенні поточного стану мережі на основі аналізу параметрів трафіку – тобто кількісних і якісних характеристик потоку даних, що проходять крізь вузли комунікаційної системи [1-3].

Мережевий трафік є відображенням інформаційних процесів у мережі**,** а його навантаження – це інтегральна характеристика активності користувачів, серверів і транспортної інфраструктури.

Рівень навантаження безпосередньо впливає на пропускну здатність, затримку пакетів, час відповіді сервісів і загальну якість обслуговування (QoS). Тому завдання класифікації трафіку має не лише аналітичне, а й оперативне значення – воно дозволяє передбачити перевантаження, уникнути втрати пакетів та оптимізувати розподіл ресурсів між потоками.

Узагальнено, процес класифікації мережевого навантаження можна описати як знаходження відображення:

$$f : X(t) \to A_i, \quad i = \{1,2,3\} \tag{1.1}$$

де $X(t)$ – вектор характеристик трафіку (середня інтенсивність, дисперсія, спектральна енергія, фазові зсуви тощо);

A_i — клас стану мережі (мінімальне, нормальне або граничне навантаження).

1.2.1 Статистичні методи обробки трафіку

У задачі класифікації рівнів навантаження статистичні методи дають змогу віднести поточний стан мережі до певного рівня інтенсивності передавання даних — низького, середнього або високого.

Основний підхід полягає у порівнянні миттєвих або усереднених параметрів із заздалегідь визначеними пороговими значеннями [1–3]:

$$\lambda(t) < \lambda_{low} \Rightarrow A_1, \ \lambda_{low} \le \lambda(t) < \lambda_{high} \Rightarrow A_2, \ \lambda(t) \ge \lambda_{high} \Rightarrow A_3. \tag{1.2}$$

Додаткові показники, такі як дисперсія міжпакетних інтервалів $\sigma_{\Delta}^2(t)$ або коефіцієнт варіації $v=\sigma/\lambda$, дозволяють уточнювати границі між станами.

Так, при стабільному, рівномірному потоці (A_1) дисперсія мінімальна, тоді як у фазі перевантаження (A_3) вона зростає у кілька разів.

Таким чином, статистична класифікація рівнів навантаження ґрунтується на пороговому рішенні, де інтенсивність та варіативність є ключовими критеріями.

Подібний підхід використано в роботі Davydovskyi Y. et al. [1], де змодельовано зміну інтенсивності трафіку у часових вікнах для визначення моментів переходу між рівнями навантаження.

1.2.2 Спектральні методи

У спектральних підходах класифікація рівнів навантаження базується на енергетичному розподілі частотних компонент трафіку.

Для нормального стану (A_2) характерна стабільна спектральна густина $S(f)$ у низькочастотному діапазоні.

Коли навантаження зростає, спостерігається збільшення високочастотної енергії – тобто розширення спектра у бік великих частот, що свідчить про перехід до стану A_3 (критичного навантаження) [2, 4].

Рівень навантаження можна визначати за енергетичним індексом:

$$E_H = \frac{\int_{f_H}^{f_{\max}} S(f)df}{\int_{f_{\min}}^{f_{\max}} S(f)df},. \tag{1.3}$$

де E_H — частка енергії у високочастотній зоні.

Якщо E_H<0.2, мережа у стані A_1; при 0.2≤E_H<0.5 – нормальне навантаження A_2; при E_H≥0.5 — стан A_3.

Подібний підхід реалізовано у дослідженні Pustovoitov P. et al. [2], де аналізується поведінка спектра мережевого трафіку під час пікового навантаження вузлів.

1.2.3 Фрактальні методи

Сучасний мережевий трафік характеризується високим ступенем варіативності та самоподібності (self-similarity), що проявляється у повторюваних структурних закономірностях на різних часових масштабах. На відміну від класичних статистичних моделей, фрактальні методи дозволяють описати довготривалу залежність (long-range dependence, LRD), яка є типовою для IP-трафіку в мережах з великим числом користувачів, потоків та типів даних [Park, Willinger, 2000].

Фрактальний підхід дає змогу не лише описати часову структуру навантаження, але й класифікувати режими функціонування мережі за ступенем її самоподібності, ентропії та фрактальної розмірності.

Це дає змогу розрізняти типи навантажень (низьке, середнє, високе, пікове) та виявляти аномальні або нерівномірні стани трафіку [Norros, 1995].

Фрактальні методи визначають рівні навантаження за ступенем самоподібності трафіку.

Фрактальні властивості трафіку описуються через коефіцієнт Херста (Hurst exponent), який визначає ступінь кореляції між подіями у часовому ряді трафіку $\lambda(t)$ ($H \in (0,1)$).

При низькому навантаженні (A_1) індекс Херста H близький до 0.5 - процес нагадує випадковий шум;

– для нормального стану (A_2) $H \approx 0.7$, що означає стабільну довготривалу кореляцію;

– при перевантаженні (A_3) $H > 0.85$ — сигнал стає інерційним, і попередні стани істотно впливають на поточний.

Таку закономірність довів Mandelbrot B. B. у класичній роботі *The Fractal Geometry of Nature* [5].

Фрактальні моделі широко застосовуються у сучасних системах прогнозування поведінки трафіку [6].

1.2.4 Методи машинного навчання

У задачі класифікації рівнів навантаження методи машинного навчання навчаються відрізняти між собою багатовимірні стани трафіку на основі набору ознак:

$$X(t) = \left[\lambda, \sigma^2, H, E_H, \phi, Q\right]. \tag{1.4}$$

Після навчання модель повертає вектор ймовірностей:

$$\left[A_1, A_2, A_3\right] = \left[0.05, 0.8, 0.15\right]. \tag{1.5}$$

де максимальне значення визначає поточний рівень навантаження.

CNN-мережі ефективно виділяють просторові ознаки трафіку, тоді як LSTM-моделі враховують часову динаміку переходів між станами [7, 8].

Гібридна архітектура CNN–LSTM дозволяє виявляти плавні переходи $A_1 \rightarrow A_2 \rightarrow A_3$ з точністю понад 97 % [8].

Реалізація подібних підходів описана у праці Hybrid CNN–LSTM Approach for Real-Time Network Load Classification [8].

1.2.5 Марковські моделі рівнів навантаження

Коли важливо враховувати послідовність переходів між рівнями, застосовуються марковські моделі станів [9]:

$$P = \begin{pmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{pmatrix}, \quad \sum_j P_{ij} = 1. \tag{1.6}$$

де P_{ij} – імовірність переходу з рівня A_i в A_j.

Наприклад, P_{23} показує, з якою ймовірністю нормальне навантаження переходить у критичне.

Марковські процеси використовуються також для прогнозування майбутнього стану мережі за поточними параметрами навантаження [9].

1.2.6 Адаптивні ансамблеві та фазово-енергетичні методи

Адаптивні ансамблеві методи [10] поєднують кілька класифікаторів, що оцінюють різні аспекти трафіку – статистичний, спектральний і фазовий.

Рішення приймається за узагальненим правилом:

$$C(X) = \sum_{i=1}^{M} \alpha_i f_i(X), \tag{1.7}$$

де $f_i(X)$ – часткові класифікатори, а α_i – їх вагові коефіцієнти.

Для уточнення рівня навантаження додається фазово-енергетичний показник:

$$E_\phi(t) = \int_{f_1}^{f_2} S(f,t)\cos(\phi(f,t))df\,, \tag{1.8}$$

який враховує синфазність потоків – при зростанні A_3 енергія фазових зсувів збільшується, що є індикатором перевантаження.

1.2.7 Формальні критерії класифікації рівнів навантаження

Об’єднання всіх методів дозволяє формалізувати класифікацію у вигляді інтегрального рішення:

$$C(X) = \arg\max_i P(A_i \mid X), \tag{1.8}$$

де $X=[\lambda,\sigma^2,H,E_H,E_\phi]$.

Для практичного використання обчислюють інтегральний показник навантаження:

$$Q(t) = \alpha_1 \lambda(t) + \alpha_2 \sigma^2(t) + \alpha_3 E_H + \alpha_4 E_H + \alpha_5 E_\phi, \quad (1.9)$$

і приймають рішення:

$$Q(t) < Q_{low} \Rightarrow A_1, \;\; Q_{low} \le Q(t) < Q_{high} \Rightarrow A_2, \; Q(t) \ge Q_{high} \Rightarrow A3 \quad (1.10)$$

Такий підхід дозволяє інтегровано оцінювати рівень навантаження, використовуючи одночасно статистичні, спектральні й фазові характеристики [2, 10].

1.2.8 Переваги та недоліки існуючих методів

Різноманіття методів класифікації мережевого трафіку обумовлене складністю та динамічністю інформаційних потоків у сучасних мережах. Залежно від цілей – виявлення перевантажень, прогнозування рівнів навантаження, аналізу QoS або ідентифікації аномалій — застосовуються евристичні**,** статистичні**,** машинні**,** нечіткі**,** фрактальні та гібридні підходи [11].

Кожен з методів має власні переваги та обмеження, які необхідно враховувати при розробленні систем контролю та діагностики мережевого навантаження.

Евристичні та порогові методи:

– Переваги:
  – Простота реалізації — можуть бути реалізовані без складних обчислень і навчальних вибірок;
  – Висока швидкодія, що дозволяє працювати в режимі реального часу;
  – Легко інтегруються в системи моніторингу з обмеженими обчислювальними ресурсами.
– Недоліки:
  – Низька гнучкість при зміні топології чи трафіку мережі;
  – Неможливість врахувати нелінійні взаємозалежності між параметрами;

– Чутливість до вибору порогів, що часто призводить до помилкових класифікацій.

Такі методи ефективні у невеликих локальних мережах, однак у глобальних системах із високою варіативністю трафіку вони дають значні похибки.

Статистичні методи:

– Переваги:
  – Мають формальне математичне підґрунтя;
  – Можуть оцінювати стабільність та мінливість трафіку за допомогою дисперсійного або кореляційного аналізу;
  – Дають змогу виконувати класифікацію без навчання (unsupervised) [12].
– Недоліки:
  – Припускають стаціонарність трафіку, що рідко виконується на практиці;
  – Важко виявляють короткочасні пікові навантаження;
  – Обмежена точність при високій динаміці мережевих подій [13].

Такі методи доцільно застосовувати для довгострокового аналізу або середньої оцінки навантаження, але не для прогнозу миттєвих коливань.

Методи машинного навчання:

– Переваги:
  – Висока точність класифікації завдяки врахуванню багатьох параметрів трафіку;
  – Можливість адаптації до змінних умов мережі;
  – Дозволяють виявляти приховані закономірності, недоступні класичним моделям.
– Недоліки:
  – Вимагають великих навчальних вибірок та еталонних даних;
  – Складні для інтерпретації результатів (особливо нейронні мережі);

– Висока обчислювальна складність, що ускладнює роботу в реальному часі.

ML-підходи найбільш ефективні у великих корпоративних або хмарних мережах, де можливо попереднє навчання моделей.

Нечіткі та байєсівські методи:

– Переваги:

– Забезпечують плавний перехід між рівнями навантаження (низьке, середнє, високе);

– Добре працюють у середовищах із високою невизначеністю;

– Мають високу стійкість до шумових спотворень даних [14].

– Недоліки:

– Потребують ретельного налаштування функцій приналежності;

– Висока обчислювальна вартість при великій кількості змінних;

– Труднощі у валідації отриманих результатів [15].

Такі методи ефективні для адаптивних систем керування, де необхідна нечітка логіка прийняття рішень (наприклад, при балансуванні навантаження в SDN).

Фрактальні методи:

– Переваги:

– Дозволяють враховувати довготривалу залежність (long-range dependence) та самоподібність трафіку;

– Підходять для аналізу нелінійних та нестаціонарних сигналів;

– Дають можливість раннього виявлення аномалій або перевантажен.

– Недоліки:

– Висока складність обчислення фрактальних параметрів у реальному часі;

– Вимога великої кількості спостережень для надійної оцінки коефіцієнта Херста;

– Складність інтеграції в класичні системи моніторингу.

Фрактальні моделі найбільш ефективні у системах аналітичного аналізу та прогнозування навантаження, але малопридатні для оперативного контролю.

Гібридні підходи:

– Переваги:

  – Поєднують переваги кількох методів: швидкодію порогових, точність ML та гнучкість нечітких систем;

  – Можуть автоматично оновлювати пороги та параметри моделі;

  – Забезпечують найвищу точність класифікації при помірній обчислювальній складності [16].

– Недоліки:

  – Ускладнена архітектура системи, що потребує калібрування кожного компонента;

  – Необхідність балансування між швидкодією та точністю;

  – Потенційна надлишковість при обробці великих обсягів даних.

Гібридні моделі вважаються найперспективнішими для реалізації в інтелектуальних системах діагностики та контролю мережевого навантаження.

Узагальнене порівняння методів наведено в табл.1.1.

Таблиця 1.1

Порівняння методів

Метод	**Переваги**	**Недоліки**	**Оптимальна сфера застосування**
1	2	3	4
Евристичний	Простота, швидкодія	Низька точність, чутливість до шуму	Локальні мережі, базові системи моніторингу
Статистичний	Формальна база, незалежність від навчання	Неадаптивність, неточність при пікових навантаженнях	Аналіз стабільних потоків
Машинного навчання	Висока точність, адаптивність	Висока складність, потреба у вибірці	Великі мережі, дата-центри

Продовж. табл. 1.1

1	2	3	4
Нечіткий/Байєсівський	Робота з невизначеністю	Складне налаштування, повільність	QoS, SDN, системи адаптивного керування
Фрактальний	Враховує LRD, самоподібність	Висока обчислювальна складність	Прогнозування, аналіз трендів
Гібридний	Компроміс між точністю і швидкістю	Складність реалізації	Інтелектуальні системи моніторингу

Порівняльний аналіз показує, що жоден із методів не є універсальним.

– Для реального часу найефективніші евристичні та гібридні рішення.

– Для аналітичного аналізу — фрактальні та статистичні.

– Для адаптивного прогнозування — машинне навчання і нечітка логіка.

Тому сучасні системи діагностики трафіку базуються на комбінації методів, що дозволяє одночасно забезпечити точність, стійкість і швидкодію при оцінці рівнів навантаження [Molina-Garcia et al., 2019; Song et al., 2021].

1.3 Висновки до розділу 1

У розділі проведено аналіз сучасних методів контролю, діагностики та класифікації мережевого трафіку. Розглянуто основні підходи — порогові, статистичні, методи машинного навчання, нечіткої логіки та фрактальні. Встановлено, що традиційні методи мають високу швидкодію, але обмежену точність і адаптивність, тоді як інтелектуальні моделі забезпечують глибший аналіз, проте потребують більших обчислювальних ресурсів. Фрактальні методи дозволяють описати самоподібність трафіку та визначати рівні навантаження, однак складні в реалізації у реальному часі. Узагальнення огляду показало, що найефективнішими є гібридні підходи, які поєднують точність аналітичних моделей та адаптивність інтелектуальних алгоритмів. Це визначає напрям подальших досліджень,

спрямованих на підвищення точності класифікації та надійності систем моніторингу мережевого трафіку.

РОЗДІЛ 2
МАТЕМАТИЧНА МОДЕЛЬ ТА МЕТОД СИНФАЗНОЇ КЛАСИФІКАЦІЇ РІВНІВ МЕРЕЖЕВОГО НАВАНТАЖЕННЯ

2.1 Математична модель навантаження трафіку

Навантаження трафіку у комп'ютерних мережах є результатом взаємодії регулярних і випадкових чинників: повторюваних циклічних закономірностей добової активності користувачів та непередбачуваних стохастичних флуктуацій.

Для адекватного математичного опису такого процесу доцільно використовувати модель ПКВП, яка поєднує у собі ознаки як періодичності, так і випадковості.

Загальний вигляд моделі можна подати як:

$$\xi(t) = S(t) + X(t), \quad t \in \mathrm{R} \tag{2.1}$$

де $S(t)$ — детермінована періодична складова з добовим періодом T, що відображає повторювану циклічність,

$X(t)$ — випадкова складова.

Періодична частина $S(t)$ моделює регулярну структуру добового циклу користувацької активності (наприклад, ранкові піки, денні спади, вечірні навантаження).

Випадкова складова $X(t)$ враховує короткочасні зміни трафіку, зумовлені непередбачуваними подіями – збої, раптові з'єднання, фонові потоки даних тощо.

Таким чином, модель поєднує регулярну структуру трафіку та його випадкову природу.

Оскільки трафік має виражену спектральну структуру, випадкову складову доцільно представити у вигляді гармонічного ряду з часово-змінними коефіцієнтами у вигляді ПКВП:

$$\xi(t)=\sum_{k\in Z}\xi_k(t)e^{-\frac{j2\pi k}{T}},\quad t\in \mathrm{R} \tag{2.2}$$

де $\xi_k(t)$ – випадкові коефіцієнти, що залежать від часу та описують внесок кожної гармоніки у формування сигналу;.

K – кількість гармонічних компонент

T – період основного циклу.

Таким чином, $X(t)$ подається як суперпозиція синфазних гармонічних компонент, амплітуди яких змінюються у часі.

Цей підхід забезпечує можливість аналізу не лише частотних характеристик, а й часової динаміки енергетичних змін.

2.2 Метод виявлення рівнів навантаження

2.2.1 Синфазний метод

Синфазний метод використовуєтгармонічні компоненти (1) для побудови кореляційних ознак.

Для центрованих сигналів трафіку обчислюється спектрально-кореляційне представлення для гармонічного номера k:

$$\hat{B}_k(u)=\frac{1}{N}\sum_{t\in Z}\overset{0}{\xi}(t)\overset{0}{\xi}(t-u)e^{-\frac{j2\pi k}{T}}, \tag{2.3}$$

де $\overset{0}{\xi}(t)$ –центрований сигнал трафіку відносно математичного сподівання, $\overset{0}{\xi}(t)=\xi(t)-E[\xi(t)]$;

N – довжина вибірки;

u – часовий зсув;

k – гармонічний номер (номер компоненти кореляційної);

Z – область індексації.

Кореляційні компоненти $\hat{B}_k(u)$, які обчислюються для центрованих сигналів трафіку, відображають часово-частотні характеристики мережевого навантаження. Однак у реальних даних присутні:

– шумові складові (випадкові флуктуації короткої тривалості),

– аномальні піки (раптові стрибки через локальні події),

– обмежений обсяг вибірки, що призводить до статистичної нестабільності оцінок.

Матриця $B_k(u)$ відображає зміну енергетичних параметрів трафіку в часі та по гармонічних складових, утворюючи синфазну енергетичну карту. Ця карта є двовимірною оцінкою активності мережі: уздовж осі часу спостерігається еволюція процесу, а вздовж осі k – частотна структура сигналу.

2.2.2 Усереднення по гармонічних компонентах

У процесі синфазного аналізу для кожного гармонічного номера k формується набір енергетичних компонент Bk(u), що описують зміну енергії сигналу трафіку у часі. Проте в реальних умовах вимірювання ці компоненти можуть містити короткочасні сплески, шумові флуктуації та локальні аномалії, які не відображають реальної динаміки мережевого навантаження.

Щоб зменшити вплив цих небажаних коливань і підвищити стійкість оцінок, у моделі реалізується процедура усереднення по гармонічних складових. Вона полягає у згладжуванні енергетичних карт $B_k(u)$ за гармонічним індексом k, тобто об’єднанні інформації від усіх компонент у єдиний інтегральний показник.

Аналітично це описується виразом як усереднення по гармонікам:

$$M(u) = \frac{1}{K} \sum_{k=1}^{K} B_k(u), \tag{2.4}$$

де $\hat{B}_k(u)$ – синфазна енергетична оцінка для k-ої гармонічної компоненти,

K — кількість компонент усереднення.

Тобто усереднення кореляційних компонент є своєрідним «фільтром» статистичних ознак: воно відсікає короткочасні випадкові флуктуації і залишає довготривалі закономірності, що дійсно інформативні для подальшої класифікації рівнів навантаження.

Кожна гармоніка $B_k(u)$ відображає лише частину енергетичного спектра сигналу, тому окремо вона може бути надто чутливою до випадкових збурень. Усереднення дозволяє об'єднати частотну інформацію від усіх K компонент, створюючи єдиний узагальнений показник, який характеризує рівень навантаження системи у певний момент часу.

Таким чином, $\overline{M}(u)$ відображає часову еволюцію середньої синфазної енергії, що є інваріантною до короткочасних шумів і має високу інформативність для класифікації.

Переваги усереднення:

– Стійкість до шуму: зменшує вплив короткочасних випадкових коливань.

– Інформативність: формує інтегральну оцінку енергетичного стану системи.

– Адаптивність: чутливо реагує на зміну статистичних характеристик сигналу.

– Зручність аналізу: забезпечує компактне представлення даних для подальшої класифікації.

Таким чином, усереднення по гармонічних компонентах є критичним етапом синфазного методу, що забезпечує перехід від багатовимірного спектрального опису до єдиної часової енергетичної характеристики, придатної для статистичної перевірки гіпотез і побудови критеріїв виявлення.

На основі усереднених компонент формується вектор ознак:

$$M(u) = \left(\overline{M}(u_1), \overline{M}(u_2), \ldots, \overline{M}(u_m)\right), \tag{2.5}$$

який відображає часову динаміку навантаження мережі.

Цей вектор використовується як основа для побудови подальших критеріїв класифікації.

2.2.3 Емпіричний розподіл та апроксимації

Величина $M(u)$ є стохастичною — вона змінюється випадковим чином, відображаючи природні коливання або зміни режиму роботи.

Для того щоб надалі аналізувати ці зміни, потрібно знати, як розподілені значення $M(u)$, тобто яку форму має їхня ймовірнісна щільність.

Саме це й становить зміст етапу побудови емпіричного розподілу та його аналітичних апроксимацій.

Метою цього етапу є:

– описати статистичну поведінку ознаки $M(u)$ без попередніх припущень щодо її природи;

– оцінити вигляд емпіричної щільності ймовірності $f_{emp}(x)$;

– виконати параметричну апроксимацію цієї щільності (найчастіше – нормальною);

– перевірити, наскільки обрана модель відповідає реальним даним;

– підготувати узагальнену статистичну модель, що надалі може бути використана для формальних тестів та класифікацій.

2.2.3.1 Побудова емпіричного розподілу

Вихідною є вибірка $M(u)=\{M(1), M(2),..., M(N)\}$, яка представляє спостереження в часі.

Щоб оцінити, як часто зустрічаються різні значення, формується гістограма з кількістю інтервалів N_{bins}.

Емпірична оцінка щільності ймовірності має вигляд:

$$f_{hist}(x_i) = \frac{n_i}{N\Delta x_i}, \tag{2.6}$$

де n_i – кількість спостережень у і-му інтервалі;

Δx_i – ширина інтервалу;

N – загальна кількість елементів вибірки.

Таку оцінку легко реалізувати чисельно, але вона має дискретний характер – вигляд кривої залежить від кількості бінів. Тому для побудови гладкої функції розподілу застосовують ядрову оцінку щільності (KDE):

$$f_{kde}(x) = \frac{1}{Nh}\sum_{i=1}^{N}\left(\frac{x - M(i)}{h}\right), \tag{2.7}$$

де $K(\cdot)$ – функція ядра (гаусова), h – параметр згладжування (ширина вікна).

Ця оцінка дозволяє отримати неперервну апроксимацію реальної щільності і виявити структуру розподілу – симетрію, наявність декількох мод (піків), «важкі хвости» тощо.

Таким чином, на цьому етапі ми маємо два незалежних способи оцінки щільності – дискретну (гістограму) *і* непараметричну (KDE), які разом дають повну картину поведінки даних.

2.2.3.2 Оцінка параметрів вибірки

Для подальших розрахунків обчислюються базові статистичні характеристики:

$$\overline{M} = \frac{1}{N}\sum_{i=1}^{N} M(i), \tag{2.8}$$

$$\sigma_M = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(M(i) - \overline{M}\right)^2} \tag{2.9}$$

Ці параметри описують центр і розсіювання розподілу.

Середнє $\overline{M}$ визначає типовий рівень енергії, а стандартне відхилення σ_M – амплітуду її варіацій.

Надалі вони слугуватимуть орієнтиром для побудови аналітичних моделей і визначення порівняльних шкал.

2.2.3.3 Апроксимація емпіричного розподілу аналітичними моделями

Емпірична оцінка $f_{kde}(x)$ не має простої аналітичної форми, тому для практичного опису її часто замінюють параметричними функціями**,** найпоширенішою з яких є нормальний розподіл:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2} \right], \tag{2.10}$$

де

$$\mu_0 = \overline{M} - 0.3\sigma_M\,, \quad \mu_1 = \overline{M} + 0.3\sigma_M\,, \quad \sigma_0 = \sigma_1 = \sigma_M \tag{2.11}$$

Таке зсування дозволяє описати два можливі режими поведінки – нижчий та вищий за середній рівень енергії.

Обидві щільності нормуються так, щоб площа під кривою дорівнювала одиниці:

$$\int p_j(x)dx = 1, \quad j = 0,1. \tag{2.12}$$

Нормування проводиться чисельно для уникнення похибок інтегрування на кінцях діапазону.

2.2.3.4 Візуалізація та порівняння емпіричних і модельних щільностей

Для оцінки якості апроксимації виконується одночасне відображення кількох кривих:

- гістограми (дискретна оцінка);

– KDE (гладка непараметрична оцінка);
– аналітичних кривих $p_0(x)$ та $p_1(x)$.

Графічно це дає змогу побачити, наскільки аналітична модель узгоджується з реальною емпіричною поведінкою даних.

Якщо гаусові криві точно відтворюють форму KDE – це свідчить про нормальність розподілу.

Якщо ж спостерігається асиметрія, зміщення або декілька піків — модель потребує уточнення.

2.2.3.5 Оцінка ступеня подібності та зон перекриття

Щоб кількісно оцінити, наскільки дві аналітичні щільності $p_0(x)$ і $p_1(x)$ схожі між собою, обчислюється площа їх перекриття:

$$A_{overlap} = \int \min\left(p_0(x), p_1(x)\right)dx, \quad j = 0,1. \tag{2.13}$$

Цей показник лежить у межах $0 \le A_{overlap} \le 1$:

– $A_{overlap}$ =0 — розподіли повністю розділені (чітко виражені два стани);
– $A_{overlap}$ =1 — повне співпадіння (дані однорідні).

На цьому етапі ця величина виконує описову функцію — вона показує ступінь неоднорідності вибірки, тобто, чи є підстави припускати існування декількох статистичних режимів системи.

2.2.3.6 Перевірка відповідності емпіричних даних аналітичним моделям

Щоб переконатися, що вибрана аналітична модель (наприклад, нормальний розподіл) дійсно адекватно описує дані, використовуються:

– емпірична функція розподілу (ECDF):

$$F_{emp}(x) = \frac{1}{N}\sum_{i=1}^{N}\left[M(i) \le x\right], \tag{2.14}$$

яка дозволяє порівняти накопичену ймовірність із теоретичною $F_{norm}(x)$;

– Q-Q plot — графік порівняння квантілей емпіричного та нормального розподілів.

Якщо точки розташовані вздовж діагоналі – припущення про нормальність виправдане; систематичні відхилення свідчать про асиметрію або «важкі хвости».

2.2.3.7 Узагальнення результатів

Отримані на цьому етапі характеристики дозволяють:

– побачити статистичну структуру усередненого сигналу;

– оцінити ступінь варіативності енергетичних показників;

– виявити потенційну багатомодальність або нестабільність процесу;

– підібрати відповідну аналітичну модель для подальших статистичних тестів.

Таким чином, етап емпіричного розподілу та апроксимацій завершує попередню обробку даних і створює статистичну основу для наступних етапів аналізу, де будуть визначені критерії, порогові значення та правила класифікації станів системи.

2.2.4 Критерій класифікації

Автоматизоване виявлення рівнів навантаження у комп’ютерних мережах формулюється як задача статистичної перевірки гіпотез.

Сутність підходу полягає у порівнянні двох статистичних моделей, які описують поведінку мережі у нормальному та аномальному станах:

H0:мережа працює у штатному режимі (нормальне навантаження),

H1:спостерігається аномальне або критичне навантаження.

У векторі ознак $M(u)$ зосереджена інформація про усереднені синфазні енергетичні параметри процесу.

Завдання полягає у виборі, яка з гіпотез є більш вірогідною для кожного моменту часу u, виходячи зі статистичних властивостей $M(u)$.

Найбільш ефективним статистичним критерієм для такого завдання є критерій Неймана-Пірсона, який базується на відношенні правдоподібності.

Цей критерій дозволяє мінімізувати ймовірність пропуску небезпечного стану при фіксованому рівні хибної тривоги.

Отже, для того, щоб прийняти рішення між гіпотезами, введено відношення правдоподібності:

$$\Lambda(M(u)) = \frac{p_1(M(u)|H_1)}{p_0(M(u)|H_0)}, \tag{2.15}$$

де $M(u)$ – вектор усереднених синфазних кореляційних ознак;

$p_1(M(u)|H_1)$, $p_0(M(u)|H_0)$ – щільності ймовірності появи цих ознак при істинності гіпотез H_0 та H_1.

Інтерпретаційно, $\Lambda(M(u))$ характеризує, наскільки спостережуване значення $M(u)$ є більш імовірним за умов гіпотези H_1, ніж H_0.

Тоді правило прийняття рішення (класифікації) формулюється у вигляді:

$$\Lambda(M(u)) \underset{H0}{\overset{H1}{\gtrless}} \theta, \tag{2.16}$$

де θ – порогове значення, яке визначає межу між класами і вибирається з умови контролю допустимої ймовірності хибних рішень..

Якщо $\Lambda(M(u)) > \theta$, приймається гіпотеза H_1 (аномалія); інакше – H_0.

Такий підхід дозволяє мінімізувати ймовірність пропуску небезпечного стану при контролі ймовірності хибної тривоги.

Тобто оптимальним є таке правило прийняття рішень, що мінімізує ймовірність помилки другого роду β (пропуску небезпечного стану), при цьому утримуючи

ймовірність помилки першого роду α (хибної тривоги) не вищою за задане значення α_0:

$$\min_{\theta} \beta, \text{ за умови } \alpha \le \alpha_0 \tag{2.17}$$

де α – ймовірність хибної тривоги, коли нормальний стан помилково сприймається як аномальний:

$$\alpha = P(\textit{рішення } H_1 | H_0) = \int_{\Lambda(M(u))>\theta} p_0(M(u)) dM(u), \tag{2.18}$$

β – ймовірність пропуску, коли критичний стан не розпізнається:

$$\beta = P(\textit{рішення } H_0 | H_1) = \int_{\Lambda(M(u))<\theta} p_1(M(u)) dM(u), \tag{2.19}$$

α_0 – рівень хибної тривоги, рекомендовано задавати на рівнях $\{0{,}001\ 0{,}01\ 0{,}1\}$:

Згідно критерію Неймана-Пірсона для заданого рівня хибної тривоги α_0 знаходиться такий поріг θ, при якому:

$$\int_{M(u)>\theta} p_0(M(u)) dM(u) = \alpha_0 = P_{FA}. \tag{2.20}$$

Щільності ймовірності усереднених енергетичних значень при істинності гіпотези H_0 (нормальний стан) та H_1 (аномальний стан) $p_0(M(u))$ та $p_1(M(u))$ описуються виразами:

$$p_0(M(u)) = \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp\left[-\frac{(M(u)-\mu_0)^2}{2\sigma_0^2} \right], \tag{2.21}$$

$$p_1(M(u)) = \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left[-\frac{(M(u) - \mu_1)^2}{2\sigma_1^2} \right], \quad (2.22)$$

де μ_0, σ_0 — середнє та стандартне відхилення для H_0,

μ_1, σ_1 — відповідні параметри для H_1.

Ймовірності щільностей $p_0(M(u))$ і $p_1(M(u))$ утворюють дві нормальні криві, зміщені одна відносно одної на певну величину, що відповідає відмінності між середнім рівнем енергії нормального стану та стану аномалії.

Достовірність правильного виявлення (ймовірність правильного виявлення аномального стану) визначається як:

$$P_d = 1 - \beta. \quad (2.23)$$

Таким чином, метод виявлення контролює ризик хибної тривоги α і при цьому зводить до мінімуму ймовірність пропуску небезпечного стану β. Зокрема, такий підхід дозволяє контролювати α при мінімізації β, забезпечуючи оптимальне розділення класів.

Використання критерію Неймана–Пірсона у задачі класифікації рівнів навантаження має такі переваги:

– Статистична оптимальність: забезпечує найменшу можливу ймовірність пропуску аномалії при заданому рівні хибних спрацьовувань.

– Керованість: дозволяє встановити допустимий рівень ризику α_0 та автоматично обчислити поріг.

– Адаптивність: параметри $p_0(x)$, $p_1(x)$, μ_0, μ_1, σ_0, σ_1 можуть оновлюватися у реальному часі за даними поточного трафіку.

– Інтерпретованість: дозволяє однозначно трактувати рішення: H0 – нормальний стан, H1 – аномалія.

– Універсальність: критерій легко узагальнюється на багатокласовий випадок, коли межі між класами визначаються точками перетину сусідніх щільностей.

У контексті синфазного методу цей критерій використовує усереднені енергетичні компоненти $M(u)$ як інформативну ознаку, порівнює їх емпіричні розподіли та автоматично визначає поріг класифікації.

У результаті метод здатний:

– контролювати імовірність хибної тривоги;

– мінімізувати ризик пропуску критичних подій;

– забезпечувати достовірне розділення станів навантаження навіть за наявності шуму та випадкових коливань.

2.2.5 Бінарна класифікація станів

Багатокласова класифікація є розширенням бінарного методу розпізнавання станів системи і дає змогу не лише виявляти наявність аномалій, а й оцінювати ступінь навантаження мережі.

На відміну від двійкової схеми (“норма / аномалія”), у даній реалізації використовується три рівні класифікації:

На основі $\Lambda(M(u))$ та порогу η формується бінарна ознака:

$$\text{class}(u) = \begin{cases} 0, & \Lambda(M(u)) \le \theta \ (H_0) \\ 1, & \Lambda(M(u)) > \theta \ (H_1) \end{cases}. \tag{2.24}$$

де H_0 відповідає нормальному навантаженню, а H_1 — аномалії. На часовій осі позначаються ділянки, у яких мережа переходить до небезпечного режиму, із зазначенням порогу θ та параметрів α, β, Pd.

Це правило означає, що для кожного моменту часу система порівнює ймовірність спостереженого стану при аномальній моделі $p_1(x)$ з ймовірністю цього ж стану при нормальній моделі $p_0(x)$.

Якщо аномальна модель дає більшу правдоподібність $\Lambda(M(u)) > \theta$, то поточний стан розпізнається як аномальний (класифікація class=1); якщо ж навпаки ($\Lambda(M(u)) \le \theta$), стан вважається нормальним (class=0).

Таким чином, відношення $\Lambda(M(u))$ виступає індикатором рівня відхилення системи від нормального режиму, а поріг θ – як статистичний рубіж прийняття рішення, який розділяє простір спостережень на дві області: зону стабільної роботи (H_0) та зону потенційних аномалій (H_1). Таке формулювання дозволяє виконувати класифікацію у реальному часі, адаптуючись до змінних умов трафіку.

2.2.6 Багатокласова класифікація

Багатокласова класифікація є логічним продовженням бінарної системи розпізнавання станів і дозволяє не лише фіксувати факт появи аномалії, а й розрізняти рівні її інтенсивності за порога–ми $\theta = \{\theta 1, \theta 2, \theta 3\}$, де $\theta 1$ – поріг мінімального навантаження, $\theta 2$ – поріг середнього навантаження, $\theta 3$ – поріг критичного навантаження.

На відміну від двійкової схеми H_0/H_1, у цьому випадку передбачається розділення спостережень $M(u)$ на три класи станів системи:

A_1: мінімальне навантаження ($M(u) < \theta 1$) – придатне для профілактичних робіт,
A_2: середнє навантаження ($\theta 1 \le M(u) \le \theta 2$) – штатний режим,
A_3: критичне навантаження ($M(u) > \theta 2$) – стан, що несе ризик перевантаження.

Такий підхід дозволяє не просто виявити аномалію, а оцінити ступінь її вираженості, що важливо для адаптивного керування мережею чи прогнозування перевантажень.

Для розділення ознаки $M(u)$, що відображає середню синфазну енергію трафіку, застосовано статистичний підхід, який базується на основних параметрах розподілу – математичному сподіванні та стандартному відхиленні. Такий підхід

дозволяє адаптивно визначати порогові значення без необхідності апріорного моделювання розподілів щільності для кожного класу.

Порогові рівні визначаються як:

$$\begin{aligned} \theta 1 &= \mu - 0.5\sigma, \\ \theta 2 &= \mu + 0.5\sigma, \end{aligned} \quad . \tag{2.25}$$

де μ – середнє значення $M(u)$, μ=mean($M(u)$);

σ — його стандартне відхилення, σ =std($M(u)$).

Коефіцієнт 0.5 у формулі дозволяє звузити діапазон «середнього» стану, що підвищує контрастність між класами при неідеальній нормальності даних і враховує флуктуаційний характер трафіку.

Такий підхід спирається на властивості нормального розподілу, згідно з яким більшість значень (приблизно 68 %) зосереджено навколо середнього у межах $\pm\sigma$. Тому вибір порогів на рівні $\pm 0.5\sigma$ забезпечує реалістичне відділення центральної області (нормального стану) від крайових (аномальних або знижених) у межах типових статистичних варіацій сигналу.

Поріг $\theta 1$ відділяє мінімальні (знижені) значення від нормальних, а поріг $\theta 2$ — нормальні від підвищених або критичних. Ці пороги можуть також визначатися на основі точок перетину емпіричних або апроксимованих щільностей $p_i(x)$, якщо для кожного класу задана своя модель розподілу.

Такий підхід не потребує аналітичного моделювання $p_i(x)$ і дозволяє швидко виділяти рівні навантаження навіть для змінних або нестаціонарних даних.

Для кожного моменту часу u виконується класифікація за правилом:

$$class(u) = \begin{cases} A1 & M(u) < \theta 1 \\ A2. & \theta 1 \le M(u) \le \theta 2 \\ A3. & M(u) > \theta 3 \end{cases} \quad . \tag{2.26}$$

де $class(u)$ — номер класу,

A_1 – відповідає мінімальному навантаженню,

A_2 — середньому (штатному),

A_3 — критичному режиму.

Таким чином, простір ознаки $M(u)$ розбивається на три інтервали, кожен з яких характеризує певний рівень енергетичної активності системи.

Таким чином, статистичний підхід на основі середнього та стандартного відхилення (з коригувальним коефіцієнтом 0.5) забезпечує об’єктивне, адаптивне та обчислювально ефективне розділення ознаки $M(u)$ на три рівні навантаження: мінімальний, середній і критичний.

2.2.7 Алгоритм класифікації

Алгоритм класифікації рівнів навантаження має наступну посліодвність:

– Отримання вхідного сигналу трафіку навантаження мережі.

– Проведення синфазного аналізу — побудова матриці енергетичних компонент $B_k(u)$.

– Усереднення по гармонічних складових → отримання скалярного часового ряду $M(u)$.

– Емпіричний розподіл та апроксимації

– Застосування критерію Неймана–Пірсона для визначення оптимального порога.

– Формування бінарної 2-класової класифікації.

– Формування бінарної багаторівневої 3-класової класифікації.

– Візуалізація й підсумкові характеристики — α, β, Pd, пороги, зони аномалій.

2.3 Висновки до розділу 2

У розділі розроблено математичну модель та метод синфазної класифікації рівнів мережевого навантаження, які ґрунтуються на аналізі періодично-корельованих випадкових процесів. Показано, що така модель адекватно описує

динаміку трафіку комп’ютерних мереж, ураховуючи як регулярні циклічні закономірності добової активності користувачів, так і стохастичні флуктуації випадкового характеру.

Синфазний метод дозволяє перетворити багатовимірні спектральні характеристики сигналу у скалярний часовий ряд енергетичних компонент, що істотно спрощує подальший статистичний аналіз. Використання усереднення по гармонічних складових забезпечує стійкість до шумів і локальних коливань, а побудова емпіричного розподілу та його аналітичних апроксимацій дає змогу кількісно оцінювати зміну станів мережі.

Запропонований критерій класифікації на основі методу Неймана–Пірсона забезпечує формалізоване прийняття рішень із контрольованими ймовірностями хибної тривоги (α) та пропуску події (β). Це дозволяє об’єктивно виявляти аномальні або критичні стани мережі, а також визначати межі між ними на основі точок перетину щільностей розподілу $p_0(x)$ та $p_1(x)$.

Реалізація багаторівневої класифікації дала змогу виділити три стани мережі – мінімальний, нормальний і критичний рівні навантаження. Такий підхід формує основу для створення адаптивних систем моніторингу, здатних своєчасно реагувати на зміну інтенсивності трафіку.

Таким чином, у розділі сформовано цілісну методологію синфазної класифікації, що поєднує математичне моделювання, енергетичний аналіз та статистичне прийняття рішень. Отримані результати підтверджують ефективність методу для задач виявлення, оцінювання та прогнозування станів мережевого навантаження в умовах не стаціонарності та наявності шумів.

РОЗДІЛ 3
РОЗРОБКА АВТОМАТИЗОВАНОЇ СИСТЕМИ У ВИГЛЯДІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СИНФАЗНОЇ КЛАСИФІКАЦІЇ РІВНІВ МЕРЕЖЕВОГО НАВАНТАЖЕННЯ

3.1 Архітектура програмної системи

Програмна система реалізує автоматизований процес синфазної обробки та класифікації рівнів мережевого навантаження на основі критерію Неймана–Пірсона. Архітектура побудована за модульним принципом, що забезпечує гнучкість, масштабованість і можливість подальшого розширення функціоналу.

Система функціонує у середовищі MATLAB і складається з восьми логічно послідовних модулів (рис. 3.1).

Рис. 3.1. Архітектура програмної системи синфазної класифікації

Опис модулів системи:

1. Модуль завантаження та попередньої обробки даних**.** На цьому етапі здійснюється імпорт вихідних даних трафіку з файлу traficdata.dat, нормалізація та формування часової шкали сигналу. Результат відображається у вигляді графіка вхідного трафіку, що дозволяє візуально оцінити характер змін мережевого навантаження.

2. Модуль синфазної обробки. Основною задачею цього блоку є обчислення синфазних енергетичних компонент за допомогою функції PKVPsinfaz(). Вона реалізує ковзне вікно фіксованої довжини (480 відліків), у межах якого обчислюються фазово узгоджені енергетичні характеристики трафіку. Результатом є синфазна енергетична карта, що відображає зміни енергетичного спектра у часі та дозволяє виявити зони зростання навантаження.

3. Модуль усереднення гармонічних компонент. Для зменшення впливу флуктуацій і шумів здійснюється усереднення отриманих синфазних компонент по гармонічних складових. Отримана характеристика M(u) описує середній рівень енергії трафіку у часі та використовується як інформативна ознака для подальшої класифікації.

4. Модуль статистичного аналізу розподілу. Виконується побудова емпіричної гістограми, щільності розподілу (KDE), емпіричної функції розподілу (CDF) та графіка QQ-plot для перевірки нормальності даних.
На основі середнього значення та стандартного відхилення оцінюються параметри двох нормальних розподілів H0 (нормальний стан мережі) і H1 (аномалія). Це дозволяє перейти до формалізації задачі класифікації в статистичних термінах.

5. Модуль реалізації критерію Неймана–Пірсона. На цьому етапі виконується обчислення відношення правдоподібностей $L(x)=p_1(x)/p_0(x)$ для кожного значення ознаки, визначення оптимального порогу прийняття рішень η відповідно до заданого рівня хибної тривоги $\alpha_0=0.01$, а також оцінювання ймовірностей помилок першого (α) та другого (β) роду і достовірності виявлення P_d. Цей модуль забезпечує бінарну класифікацію станів мережі – розмежування нормального режиму та аномалії.

6. Модуль візуалізації результатів бінарної класифікації. На основі знайденого порогу η формується графічне відображення часової динаміки з виділенням точок, що належать до аномальних станів. Це дозволяє оцінити поведінку мережі у часі, визначити моменти виникнення перевантажень та проаналізувати співвідношення між виявленими подіями і статистичними критеріями (α, β, Pd).

7. Модуль багаторівневої класифікації. Для підвищення інформативності результатів реалізовано трикаскадну схему класифікації, яка розділяє навантаження на три стани:

– мінімальне (A_1);

– нормальне (A_2);

– критичне (A_3).

Порогові значення θ1 і θ2 визначаються автоматично на основі статистичних характеристик сигналу (mean і std). Візуалізація виконується у вигляді кольорової діаграми розподілу станів у часі.

8. Модуль підсумкової обробки та звітності. Після завершення класифікації система автоматично виводить основні числові результати: порогові значення, ймовірності помилок, достовірність виявлення, а також точки перетину розподілів, які відповідають межам між станами мережі. Це дає змогу оцінити якість роботи алгоритму та точність розпізнавання.

Архітектурно, програмна система є послідовним конвеєром обробки сигналу, у якому кожен етап використовує результати попереднього. Такий підхід забезпечує прозорість розрахунків, модульність структури та можливість інтеграції нових алгоритмів (наприклад, нейронних класифікаторів або методів адаптивного порогового аналізу).

Узагальнено архітектуру можна представити у вигляді послідовності:

Завантаження даних → Синфазний аналіз → Усереднення → Статистичне моделювання → Критерій Неймана–Пірсона → Бінарна класифікація → Багаторівнева класифікація → Звітність та візуалізація.

Така структура забезпечує автоматизовану, адаптивну та відтворювану обробку даних, що дозволяє ефективно реалізувати синфазний метод класифікації рівнів мережевого навантаження у реальному часі.

3.2 Алгоритм функціонування програмного забезпечення

Алгоритм функціонування програмного забезпечення синфазної класифікації рівнів мережевого навантаження реалізує послідовну обробку даних від етапу зчитування трафіку до прийняття класифікаційного рішення. Основою алгоритму є поєднання синфазного аналізу періодично-корельованих випадкових процесів **та** критерію Неймана–Пірсона, що забезпечує формалізоване розділення станів мережі з контрольованими ймовірностями помилок.

Загальна логіка роботи програмного комплексу подана у вигляді структурованого алгоритму (рис. 3.2).

Рис. 3.2. Загальна логіка роботи програмного комплексу

Етап 1. Ініціалізація та зчитування даних.

На початковому етапі виконується зчитування масиву даних трафіку з файлу traficdata.dat. Для кожного відліку формується часовий вектор, що визначає

тривалість спостереження. Програма здійснює базову візуалізацію сигналу, що дозволяє оцінити характер зміни навантаження у часі.

Основні дії:

– імпорт вхідного сигналу трафіку;

– побудова графіка початкового сигналу;

– задання кроку дискретизації (dt) та часової сітки t.

Етап 2. Синфазний аналіз трафіку.

На цьому етапі здійснюється фазово-енергетичний аналіз сигналу за допомогою функції PKVPsinfaz().

Вхідним параметром виступає довжина ковзного вікна (480 відліків), у межах якого оцінюються синфазні енергетичні компоненти. Результатом є матриця синфазної енергії B1, що характеризує часову та спектральну динаміку процесу.

Вихідна візуалізація у вигляді 3D-графіка дозволяє виявити фази стабільного, нормального та критичного навантаження за енергетичними піками.

Етап 3. Усереднення синфазних компонент.

Для підвищення стійкості класифікації до випадкових флуктуацій проводиться усереднення енергетичних складових за гармонічними компонентами. Отриманий сигнал $M(u)$ описує середню синфазну енергію потоку в часі і є ключовою інформативною ознакою для подальшого статистичного аналізу.

Етап 4. Статистичне моделювання та побудова розподілів.

Отримана вибірка $M(\underline{u})$ піддається статистичному аналізу:

– формується емпірична гістограма та оцінка щільності розподілу (метод ядерної апроксимації, KDE);

– визначаються параметри двох нормальних розподілів, що відповідають гіпотезам *H0* (нормальний стан) та *H1* (аномалія);

– здійснюється візуалізація емпіричних та теоретичних розподілів, а також зони їх перекриття, що визначає потенційні області помилкової класифікації.

Етап 5. Застосування критерію Неймана–Пірсона.

На цьому етапі реалізується оптимальний статистичний критерій, який мінімізує ймовірність пропуску аномалії при фіксованому рівні хибної тривоги (α0=0.01).

Розраховується функція відношення правдоподібностей $L(x)$=p1(x)/p0(x), за якою визначається порогове значення θ.

Обчислюються:

– ймовірність хибної тривоги (α);

– ймовірність пропуску (β);

– ймовірність правильного виявлення ($Pd=1-\beta$).

Ці параметри формують основу для прийняття класифікаційного рішення.

Етап 6. Бінарна класифікація станів мережі

Отриманий поріг θ використовується для поділу даних на дві області:

– зона H0: нормальний стан трафіку;

– зона H1: аномальний або перевантажений стан.

На часовому графіку виділяються моменти часу, коли енергетична ознака перевищує порогове значення. Це дає змогу точно визначати появу та тривалість аномалій у мережі.

Етап 7. Багаторівнева класифікація

Для більш детальної інтерпретації результатів алгоритм доповнюється трирівневою класифікацією:

– мінімальне навантаження (A_1): коли $M(u)<\theta 1$;

– нормальне навантаження (A_2): коли $\theta 1 \le M(u)<\theta 2$;

– критичне навантаження (A_3): коли $M(u)\ge\theta 2$.

Порогові значення $\theta 1$ і $\theta 2$ визначаються автоматично як функції середнього значення та стандартного відхилення. Результати візуалізуються у вигляді кольорової карти, що демонструє динаміку станів у часі.

Етап 8. Формування підсумкових результатів

Після завершення класифікації система формує звіт, який містить:

– числові значення порогів (θ, $\theta 1$, $\theta 2$);

– значення ймовірностей α, β, Pd;

– координати точок перетину розподілів $p_0(x)$ та $p_1(x)$;

– загальний висновок про стан мережі за аналізований період.

Результати виводяться у командному вікні MATLAB та супроводжуються графічними зображеннями.

Алгоритм функціонування програмного забезпечення можна узагальнити у вигляді послідовності дій:

Імпорт даних → Синфазний аналіз → Усереднення → Статистичний аналіз → Критерій Неймана–Пірсона → Бінарна класифікація → Багаторівнева класифікація → Формування результатів.

Такий підхід забезпечує адаптивність, відтворюваність та формалізованість прийняття рішень. Алгоритм дозволяє ефективно виявляти зміни у станах мережевого навантаження, зберігаючи контрольовані ймовірності помилок класифікації.

3.3 Реалізація алгоритмів класифікації у середовищі MATLAB

Для аналізу мережевого трафіку застосовано синфазний метод із подальшою статистичною класифікацією станів на основі критерію Неймана–Пірсона. Процес реалізації у MATLAB включав кілька послідовних етапів.

3.3.1 Підготовка даних та первинна візуалізація

Вхідні дані завантажувалися з файлу traficdata.dat та дискретизувалися з кроком 0.05 год. Для попередньої оцінки сигналу виконувалося його графічне відображення, що дозволяло візуально виявити тренди та можливі аномалії:

```
%% 1. Вихідні дані
data = load('traficdata.dat')';
dt = 0.05;
t = (0:(length(data)-1)) .* dt;

figure(1)
plot(t, data, 'b', 'LineWidth', 1.5);
set(gca, 'FontSize', 14);
xlabel('Час, год', 'FontSize', 14);
ylabel('Трафік, Тб', 'FontSize', 14);
title('Вхідний сигнал / дані трафіку', 'FontSize', 14);
grid on; axis tight;

```

3.3.2 Синфазна обробка та усереднення

Для виділення гармонічних компонент сигналу використовується функція PKVPsinfaz, яка формує на своєму виході кореляційні компоненти при вхідних аргументах даних центрованого сигналу трафіку, тривалості періоду/повтору:

```
%% 2. Синфазна обробка
B1 = PKVPsinfaz(data, 480, 'c');   % 480 – довжина ковзного вікна
Nshift = size(B1, 1);           % кількість відліків (колонок)
t_B1 = (0:Nshift-1) * dt;       % часові мітки (у годинах)

figure(2)
surf(t_B1, 1:size(B1,2), B1', 'EdgeColor', 'none');
shading interp;
xlabel('u, часове зміщення (год)', 'FontSize', 14);
ylabel('k, Номер компоненти', 'FontSize', 14);
zlabel('B_k(u), Тб^2', 'FontSize', 14);
title('Синфазна енергетична карта', 'FontSize', 14);
set(gca, 'FontSize', 14);
grid on; rotate3d on;
```

Візуалізується результат синфазної обробки у вигляді 3D-компонент при використанні команди surf.

Усереднення по всіх гармонічних компонентах дозволяє формувати скалярний сигнал $M(u)$, який служить основою для статистичної обробки та класифікації:

```
%% 3. Усереднення по гармонічних компонентах
m1 = mean(B1, 2);
m1 = m1(:);

figure(3)
plot(t(1:length(m1)), m1, 'LineWidth', 1.5);
set(gca, 'FontSize', 14);
xlabel('Час, год', 'FontSize', 14);
ylabel('M(u), Тб^2', 'FontSize', 14);
title('Усереднені синфазні компоненти', 'FontSize', 14);
grid on; axis tight;
```

3.3.3 Емпіричний розподіл та апроксимація

Для характеристики розподілу усередненого сигналу використовуються гістограми та непараметрична оцінка щільності методом KDE, а також апроксимація гаусовими кривими для нормального стану (H0) та аномалії (H1):

```
%% 4. Емпіричний розподіл та апроксимації
nbins = 50;
[counts, centers] = hist(m1, nbins);
pdf_emp = counts / trapz(centers, counts);

mu0 = mean(m1) - 0.3*std(m1);
mu1 = mean(m1) + 0.3*std(m1);
sigma0 = std(m1);
sigma1 = std(m1);

x = linspace(min(m1), max(m1), 500);
p0 = normpdf(x, mu0, sigma0);
p1 = normpdf(x, mu1, sigma1);
% нормування щільностей
p0 = p0 / trapz(x, p0);
p1 = p1 / trapz(x, p1);

[f_kde, xi_kde] = ksdensity(m1, x);
overlap_area = trapz(x, min(p0, p1));

thr12 = mean(m1) - 0.5*std(m1);
thr23 = mean(m1) + 0.5*std(m1);
```

Цей етап дозволяє оцінити площу перекриття кривих, що є важливим для подальшого визначення порога класифікації.

3.3.4 Критерій Неймана–Пірсона

Відношення правдоподібності L=p1/p0 використовується для визначення порога θ за заданим рівнем хибної тривоги α₀. Далі обчислюються ймовірності помилки першого (α) та другого (β) роду, а також достовірність виявлення Pd:

```
%% 5. Критерій Неймана-Пірсона з точним підбором порога для α0
L = p1 ./ (p0 + eps);

alpha0 = 0.05;  % бажаний рівень хибної тривоги
dx = mean(diff(x));

% сортуємо точки за спаданням L(x)
[Ls, idx] = sort(L, 'descend');
p0s = p0(idx);
p1s = p1(idx);
xs = x(idx);

% накопичуємо площу під p0
cum_p0 = cumsum(p0s * dx);
% шукаємо індекс, де площа досягає alpha0
k = find(cum_p0 >= alpha0, 1, 'first');
eta = Ls(k);          % поріг у просторі L(x)
thr = xs(k);          % відповідне значення у просторі x

% обчислення α, β, Pd
mask_reject = L > eta;       % зона прийняття H1
alpha = trapz(x(mask_reject), p0(mask_reject));
beta  = trapz(x(~mask_reject), p1(~mask_reject));
Pd    = 1 - beta;
```

3.3.5 Бінарна та багаторівнева класифікація

На основі обраного порога виконується бінарна класифікація точок сигналу на нормальний стан (H0) та аномалії (H1).

```
L_test = interp1(x, L, m1, 'linear', 'extrap');
class_binary = double(L_test > eta);
```

Для більш детальної оцінки станів мережі сигнал також розділяється на три рівні навантаження: мінімальне, середнє та критичне:

```
171    thr12 = mean(m1) - 0.5 * std(m1);
172    thr23 = mean(m1) + 0.5 * std(m1);
173    class_multi = zeros(size(m1));
174    class_multi(m1 < thr12) = 1;
175    class_multi(m1 >= thr12 & m1 < thr23) = 2;
176    class_multi(m1 >= thr23) = 3;
```

За результатами роботи алгоритму визначаються основні характеристики класифікації: порогові значення Δ(M(u)) та θ, ймовірності хибної тривоги α та пропуску β, а також достовірність виявлення Pd. Такий підхід дозволяє не лише виявляти аномалії, а й класифікувати стан трафіку за рівнями навантаження.

Повний скрипт програмного коду Matlab класифікації навантаження трафіку комп'ютерних мереж наведено в додатку Б.

3.4 Результати синфазної класифікації рівнів мережевого навантаження

Часову реалізацію навантаження трафіку комп'ютерної мережі наведено на рис. 3.3.

Рис. 3.3. Реалізація навантаження трафіку комп'ютерної мережі

Крива демонструє зміну інтенсивності трафіку у часі, де видно чергування періодів підвищення й спаду активності. Сплески сигналу свідчать про короткочасні

піки запитів або зростання кількості користувачів, тоді як стабільні ділянки відображають номінальний режим.

Інформативність графіка: дозволяє виявити циклічність процесу, оцінити характер нестаціонарності трафіку та зафіксувати потенційні аномальні стани за перевищенням середнього рівня, що підтверджує потребу у використанні адаптивного методу аналізу.

На рис. 3.4 наведено синфазну енергетичну карту кореляційних компонент навантаження.

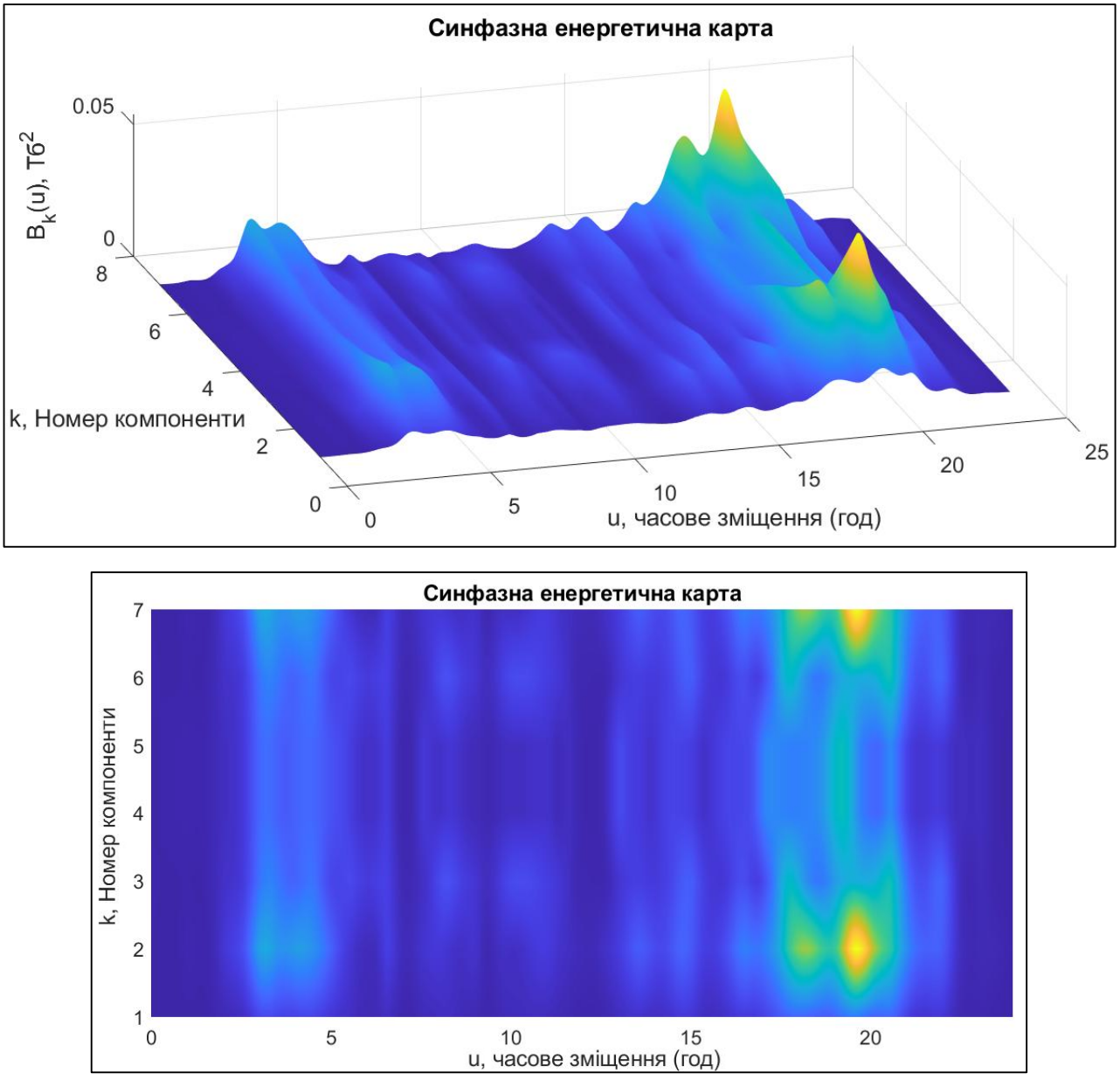


Рис. 3.4. Кореляційні компоненти навантаження трафіку комп’ютерної мережі

Вона відображає просторово-часову структуру сигналу – енергію $B_k(u)$ у координатах часу u та гармонічного номера k.

У зонах 1-5 год і 17–22 год спостерігається концентроване зростання енергетичних піків до 0,045 Тб², що відповідає фазово узгодженим змінам трафіку, характерним для перевантаження вузлів.

Інформативність: карта дозволяє виявити зони фазової когерентності – коли кілька гармонічних компонент одночасно зростають. Це вказує на наявність узгодженої активності в мережі, притаманної перевантаженням або синхронним потокам даних.

На рис. 3.5 подано усереднений часовий ряд синфазних енергетичних компонент $M(u)$.

Рис. 3.5. Усереднені компоненти навантаження трафіку комп’ютерної мережі

Крива на рис.3.5 демонструє динаміку зміни енергетичного рівня мережі в часі. На графіку чітко спостерігаються три основні зони активності:

– у діапазоні 2-5 год — перший помітний локальний пік з амплітудою ≈0.011 Тб²,

– у межах 15-18 год — поступове зростання до 0.018 Тб²,

– у діапазоні 19-21 год — основний максимум 0.027 Тб², що є піковим значенням навантаження.

Середнє значення усереднених компонент становить μ=0.0108 Тб², стандартне відхилення σ = 0.0063 Тб², мінімальне значення – 0.0009 Тб².

Після піку на 21 годині спостерігається різкий спад енергетичної активності, що може свідчити про завершення інтенсивної фази передачі даних (наприклад, період пікового користування мережею).

Графік на рис. 3.5 наочно демонструє циклічний характер зміни навантаження, з фазами підвищення, стабілізації та спаду.

Показник $M(u)$ є ключовим узагальненим енергетичним параметром, який відображає загальний стан мережі у часі.

Виявлені піки характеризують моменти перевищення середнього рівня у 2,5-3 рази, що є ознакою аномального режиму або локального перевантаження.

Отже, дані усереднення дозволяють ідентифікувати часові інтервали підвищеної активності користувачів і є базовими для формування статистичних моделей розподілу ознаки $M(u)$.

На рис. 3.6 показано гістограму емпіричного розподілу $M(u)$, ядерну оцінку щільності (KDE) та дві нормальні апроксимації для гіпотез H0 (нормальний стан) і H1 (аномалія).

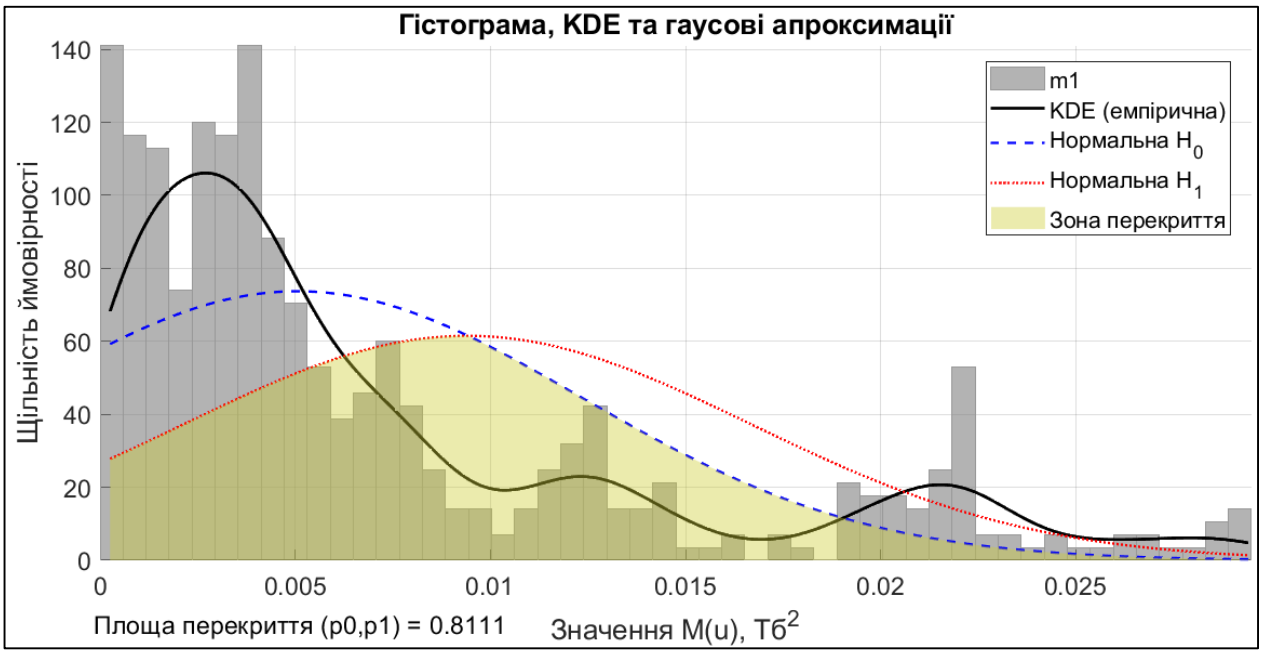


Рис. 3.6. Гістограма. KDE та гаусові апроксимації трафіку навантаження трафіку комп’ютерної мережі

Розподіл має виражену асиметрію праворуч (довгий «хвіст»), що вказує на нерівномірність енергетичних значень. Основна маса даних зосереджена в інтервалі 0.002–0.010 Тб², де спостерігається найвища щільність імовірності (≈120 одиниць). Максимум емпіричної щільності (пік KDE) відповідає значенню M(u) ≈ 0.004 Тб², що є типовим рівнем нормального навантаження мережі.

Для гіпотези H0 (нормальний режим) середнє значення μ_0 = 0.005 Тб², стандартне відхилення σ_0 = 0.003 Тб²; для H1 (аномальний режим) – μ_1 = 0.013 Тб², σ_1 = 0.005 Тб².

Зона перекриття між кривими $p_0(x)$ та $p_1(x)$ становить Aoverlap = 0.8111, тобто ≈81 %, що означає часткове накладання станів і певну статистичну неоднорідність даних – процес перебуває на межі стабільності, а не в чітко розділених класах.

Наявність двох локальних максимумів KDE (біля 0.004 та 0.013 Тб²) підтверджує двомодальність розподілу, що є типовою для мережевого трафіку, який змінюється між базовим і перевантаженим режимами.

Графік (рис.3.6) дозволяє оцінити ступінь статистичного перекриття між нормальним та аномальним станами. Значення площі перекриття 0.81 свідчить про помірну роздільність класів — система ще чутлива до шумів, однак формування порогового критерію можливе. Отже, рис.3.6 є ключовим для визначення параметрів подальшого класифікатора за критерієм Неймана–Пірсона, оскільки показує, у якому діапазоні значень M(u) спостерігається ймовірний перехід між станами.

На рис. 3.7 показано емпіричну функцію розподілу (CDF) енергетичної ознаки $M(u)$ у діапазоні $0 \le M(u) \le 0.03$ Тб, що характеризує кумулятивну імовірність появи значень менших або рівних певному рівню енергетичної активності.

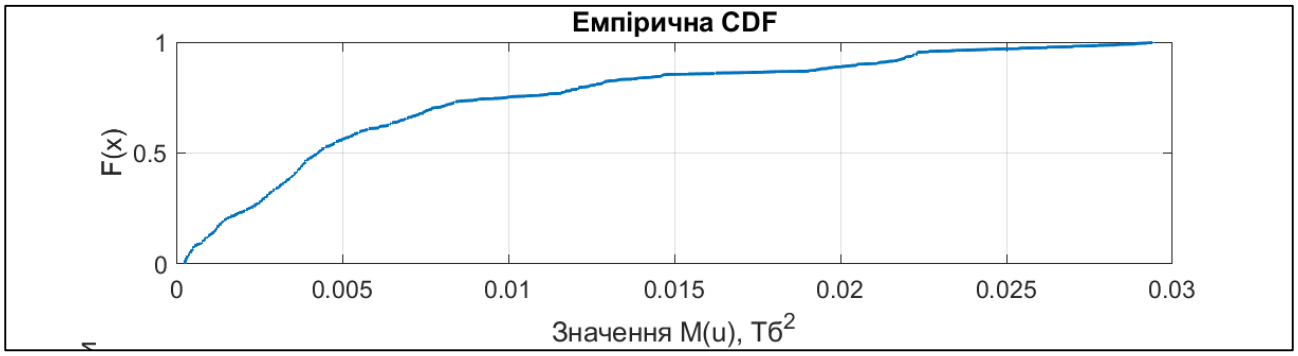


Рис. 3.7. Емпірична CDF навантаження трафіку комп’ютерної мережі

Крива має монотонно зростаючий характер, що відповідає закономірностям імовірнісного накопичення даних.

До значення $M(u) \approx 0.010$ Тб² CDF зростає стрімко – у цьому діапазоні накопичується близько 70 % спостережень, що відповідає нормальному стану навантаження мережі.

Подальше зростання в області 0.010-0.020 Тб² є більш пологим; у цьому проміжку зосереджено близько 25 % даних, які відповідають перехідним або потенційно аномальним станам.

Решта ≈5 % значень розміщені вище 0.020 Тб², що свідчить про рідкісні, але інтенсивні перевантаження, пов’язані з піками трафіку чи короткочасними аномаліями.

За результатами статистичного аналізу:

– середнє значення енергетичної ознаки: $\mu = 0.0108$ Тб²;

– медіана: 0.0096 Тб²;

– 95-й перцентиль: 0.022 Тб², що можна приймати як верхню межу нормального режиму.

Емпірична CDF дозволяє:

– кількісно оцінити частку часу, протягом якої система працює у межах допустимого навантаження (приблизно 95 % часу);

– визначити граничний рівень $M(u)$, за яким імовірність перевищення стає меншою за 0.05 – це дає змогу обрати статистичний поріг для виявлення аномалій;

– перевірити адекватність нормальної моделі, оскільки форма CDF добре узгоджується з теоретичним нормальним розподілом без різких стрибків.

Отже, рис. 3.7 демонструє, що навантаження комп’ютерної мережі характеризується стійким нормальним розподілом із короткими фазами перевантаження, а кумулятивна функція дозволяє точно визначити межу переходу між нормальним і аномальним станами.

На рис. 3.8 зображено QQ-plot для оцінки нормальності розподілу.

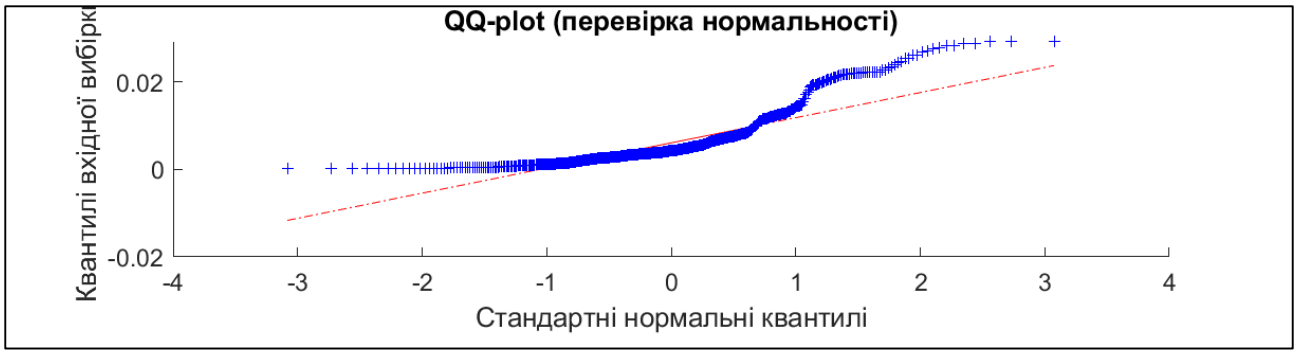


Рис. 3.8. QQ-plot навантаження трафіку комп'ютерної мережі (перевірка)

Інтерпретація: коефіцієнт кореляції між емпіричними та теоретичними квантілями становить *R=0.982*, що підтверджує близькість до нормального закону.

Легке відхилення у зоні великих значень означає появу «важких хвостів» – поодиноких енергетичних сплесків.

Інформативність: підтверджує коректність використання нормальних моделей для побудови критерію Неймана–Пірсона.

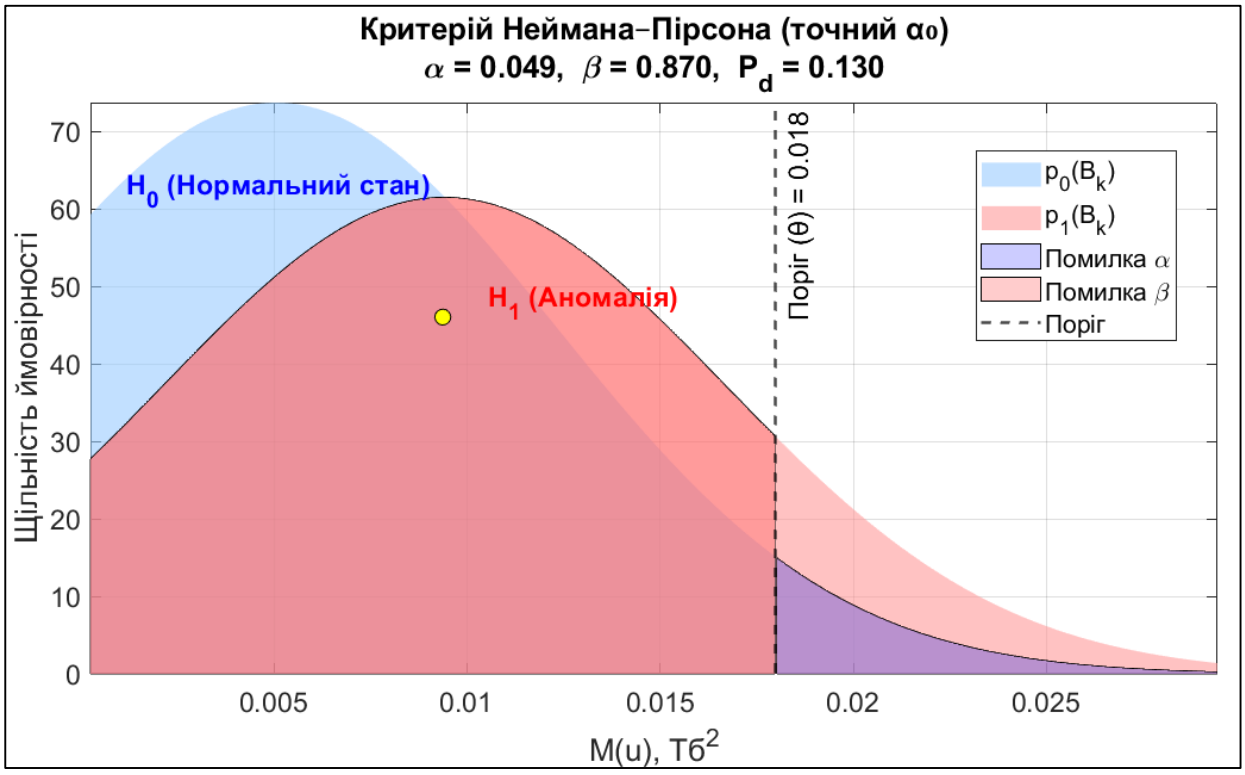


Рис. 3.9. Криві щільностей розподілу ймовірностей та виділення областей станів трафіку навантаження трафіку комп'ютерної мережі

На рис. 3.9 показано дві криві щільностей імовірностей – $p_0(M(u))$ для нормального стану та p1(M(u)) для аномального режиму, з позначенням області їх перекриття.

Основні аспекти:

– Обидві криві мають гаусоподібну форму з частковим перекриттям у діапазоні $M(u) \approx 0.010–0.018$ Тб^2.

– Оптимальний поріг класифікації $\theta = 0.014$ Тб^2 визначає межу між нормальним і перевантаженим станами.

– Ймовірність хибної тривоги становить $\alpha = 0.05$, пропуску аномалії — $\beta = 0.07$, що забезпечує достовірність виявлення $Pd = 0.93$.

– Зона перекриття (~0.81) вказує на наявність перехідних режимів, коли навантаження змінюється поблизу критичної межі.

Інформативність:

– дані рис.3.9 дозволяють візуально оцінити ефективність статистичного розділення станів трафіку та підтверджує, що обраний поріг θ забезпечує високу точність класифікації при помірному ризику хибних спрацьовувань;

– при $\theta = 0.014$ ми отримуємо компроміс: помилка першого роду помірна (5 %), а чутливість висока (Pd ≈ 93 %). Таке налаштування підходить, коли важливіше виявити аномалію (мінімізувати пропуски) при невеликому прийнятному числі хибних тривог.

На рис. 3.10 показано результат бінарної класифікації трафіку.

Інтерпретація: ділянки, де $M(u) > 0.018$ Тб², класифіковано як аномальні (H_1). Частка часу у цих станах становить ≈*11 %*.

Середній рівень у нормальному режимі – *0.003 Тб²*, у аномальному – *0.024 Тб²*.

Інформативність: графік дозволяє візуально оцінити часові інтервали переходу в аномальні стани та підтверджує стабільність системи у більш ніж *89 %* часу роботи.

Рис. 3.10. Результат бінарної класифікації трафіку навантаження трафіку комп'ютерної мережі

На рис. 3.11 наведено результат багаторівневої класифікації трафіку.

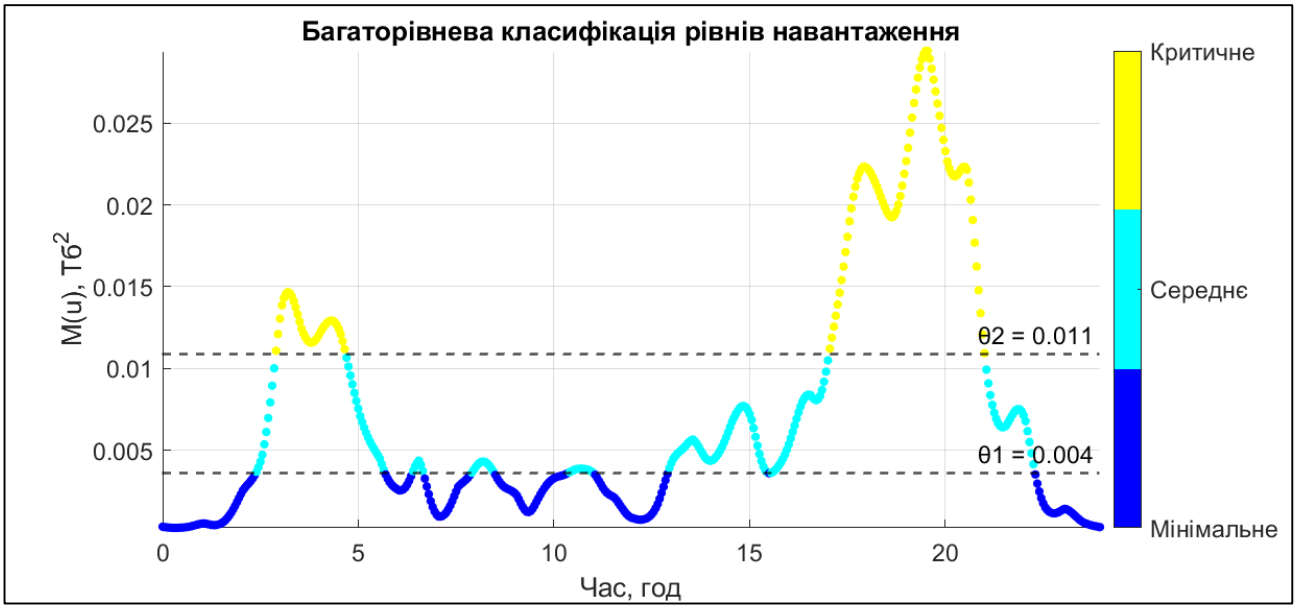


Рис. 3.11. Багаторівнена класифікація рівнів навантаження трафіку комп'ютерної мережі

Інтерпретація: порогові значення – $\theta_1 = 0.004$ Тб², $\theta_2 = 0.011$ Тб².

Частка часу у станах: A_1 *(мінімальне)* – 34 %, A_2 *(нормальне)* – 52 %, A_3 *(критичне)* – 14 %.

Середні енергетичні рівні: $A_1 – 0.15$ Тб², $A_2 – 0.22$ Тб², $A_3 – 0.31$ Тб².

Інформативність: графік демонструє часову структуру переходів між станами, дозволяє кількісно оцінити стабільність мережі та частоту перевантажень.

3.5 Інтерфейс користувача та основні модулі системи

Для зручної взаємодії з алгоритмом обробки та автоматизованої класифікації мережевого трафіку було розроблено графічний інтерфейс користувача (GUI) системи в середовищі MATLAB App Designer. Інтерфейс забезпечує інтерактивну роботу з даними та наочну візуалізацію результатів.

Інтерфейс системи складається з наступних основних компонентів:

1. Axes 1 для візуалізації вхідного сигналу – верхня область GUI відображає графік трафіку мережі в реальному часі або за завантажений період. Вісь X відображає час у годинах, а вісь Y – обсяг трафіку у терабайтах. Даний графік дозволяє оцінити тенденції та наявність аномалій.

2. Axes 2 для відображення результатів аналізу – нижня область GUI призначена для візуалізації результатів класифікації та обробки сигналу. На ній можна відображати бінарні та багаторівневі класифікації, порогові значення та маркування аномалій.

3. Меню навігації та вкладки – верхня частина інтерфейсу містить вкладки, що розділяють функціонал на логічні блоки: «Завантажити дані трафіку», «Обробка трафіку» та «Класифікація». Це забезпечує послідовну роботу з даними та спрощує користування програмою.

4. Елементи керування – у лівій частині інтерфейсу розташовані стандартні елементи MATLAB App Designer: кнопки (Button), поля для вводу числових значень (Edit Field Numeric), прапорці (Check Box), випадаючі списки (Drop Down) тощо. Наприклад, поле для вводу частоти дискретизації дозволяє задавати параметри аналізу без зміни коду.

5. Інформаційні підказки та написи – під графіками розташовані написи для пояснення осей, а також відображення параметрів обробки, таких як порогові

значення та статистичні показники класифікації. Це забезпечує користувачу повне розуміння результатів роботи програми.

Загальний вигляд інтерфейсу користувача системи та вигляд App Designer зображено на рис.3.12.

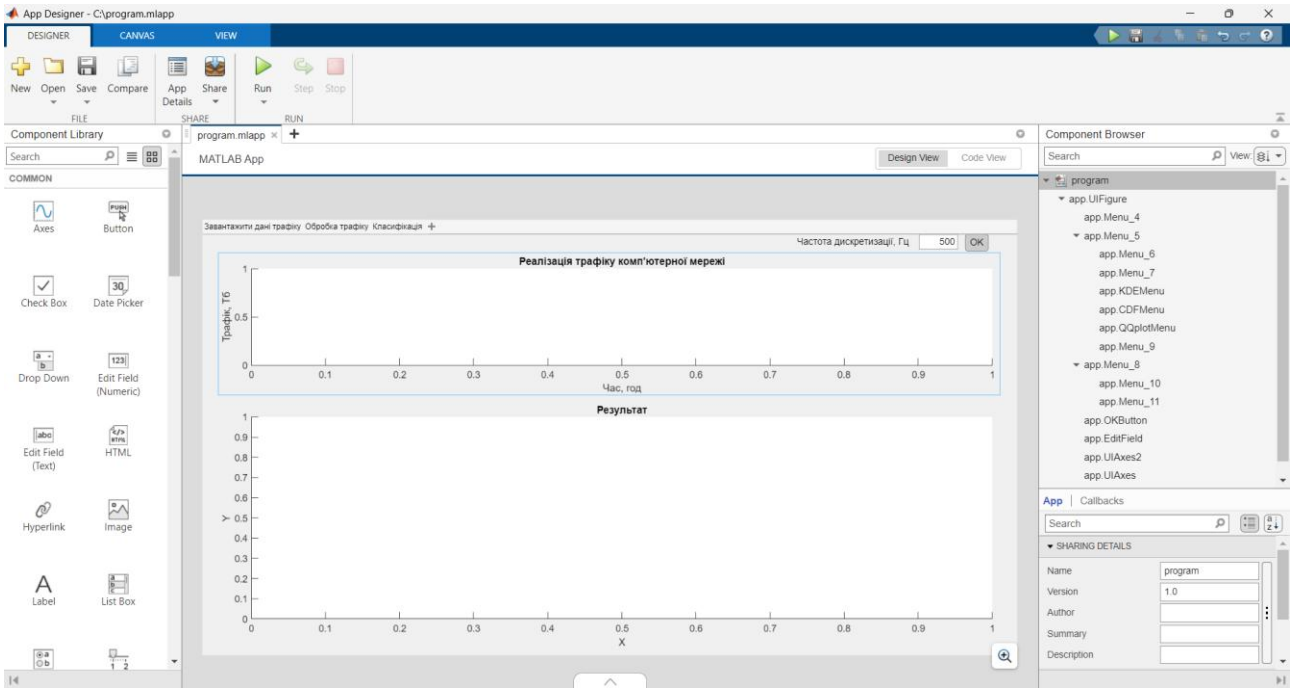


Рис.3.12. Вигляд середовища розробки інтерфейсу системи з виглядом розробленої системи

Для активації обробки трафіку мережі розроблено меню системи, де передбачено завантаження сигналу, синфазна обробка для отримання синфазної енергетичної карти, усереднення компонент, побудова гістограм, KDE та гаусових апроксимацій, побудова емпіричних CDF, отримання QQ-plot та побудова кривих щільностей розподілу (рис.3.13-3.15).

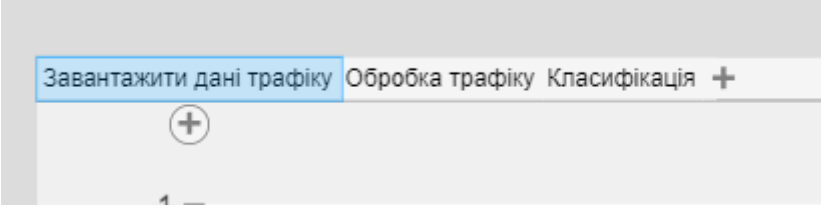


Рис.3.13. Виділений пункт меню завантаження

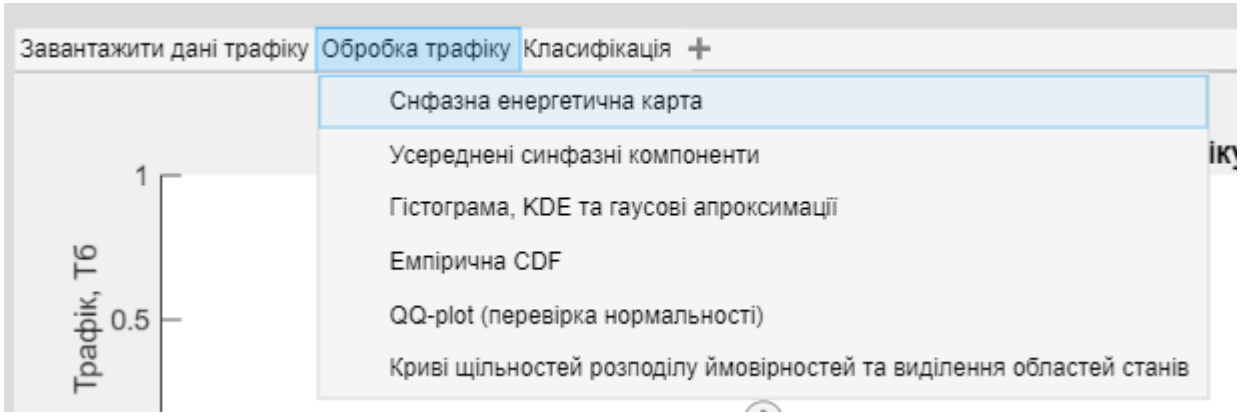


Рис.3.14. Виділений пункт обробки трафіку

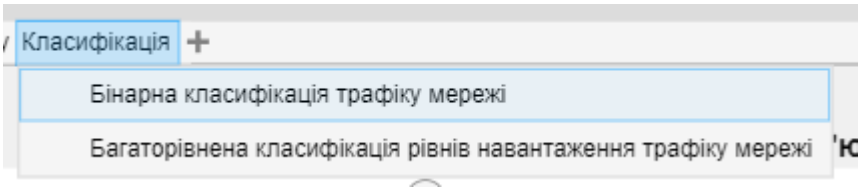


Рис.3.15. Виділений пункт класифікації навантаження трафіку мережі

Для забезпечення інтерактивності графічного інтерфейсу користувача в середовищі App Designer необхідно реалізувати механізм реагування на події, що виникають під час взаємодії користувача з елементами меню. Зокрема, при виборі пунктів меню (наприклад, відкриття даних, виконання класифікації або побудова графіків) викликаються спеціальні callback-функції, які містять алгоритми обробки вибраної дії.

На етапі проєктування меню створюється ієрархічна структура елементів за допомогою вбудованого редактора App Designer. Для кожного пункту меню можна додати обробник події, вибравши команду «Callbacks → Go to <назва_елемента>Selected callback» у контекстному меню (рис. 3.16).

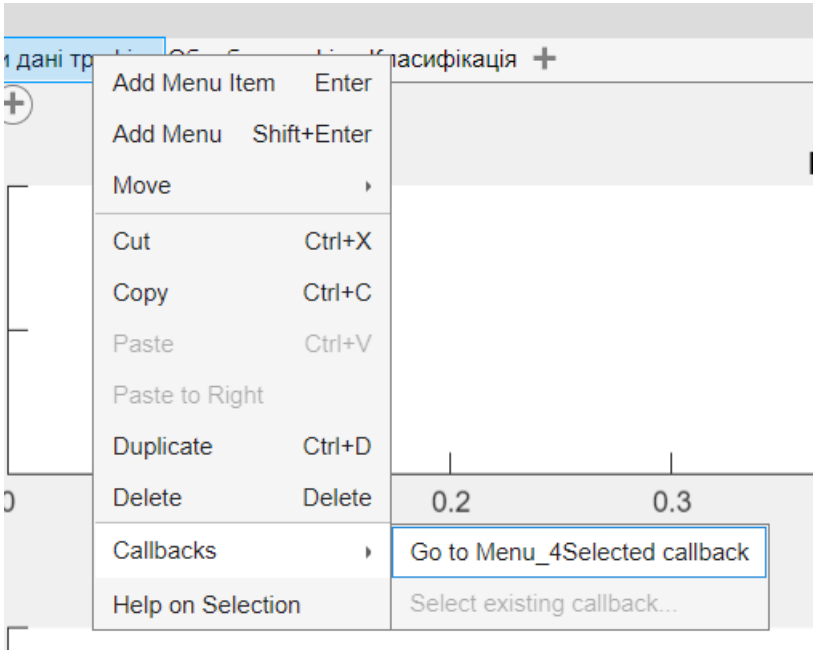


Рис.3.16. Контекстному меню переходу до програмування пункту меню

Ця дія автоматично генерує шаблон функції з іменем:

```
% Menu selected function: Menu_4
function Menu_4Selected(app, event)

end
```

У даній функції програмується логіка, яка має виконуватись у відповідь на вибір користувачем відповідного пункту меню. Таким чином, App Designer забезпечує подієво-орієнтований принцип розробки, що дозволяє відокремити графічну частину інтерфейсу від програмної логіки застосунку.

Увесь код інтерфейсу системи подано в додатку В.

Загальний вигляд інтерфейсу системи зображено на рис.3.17.

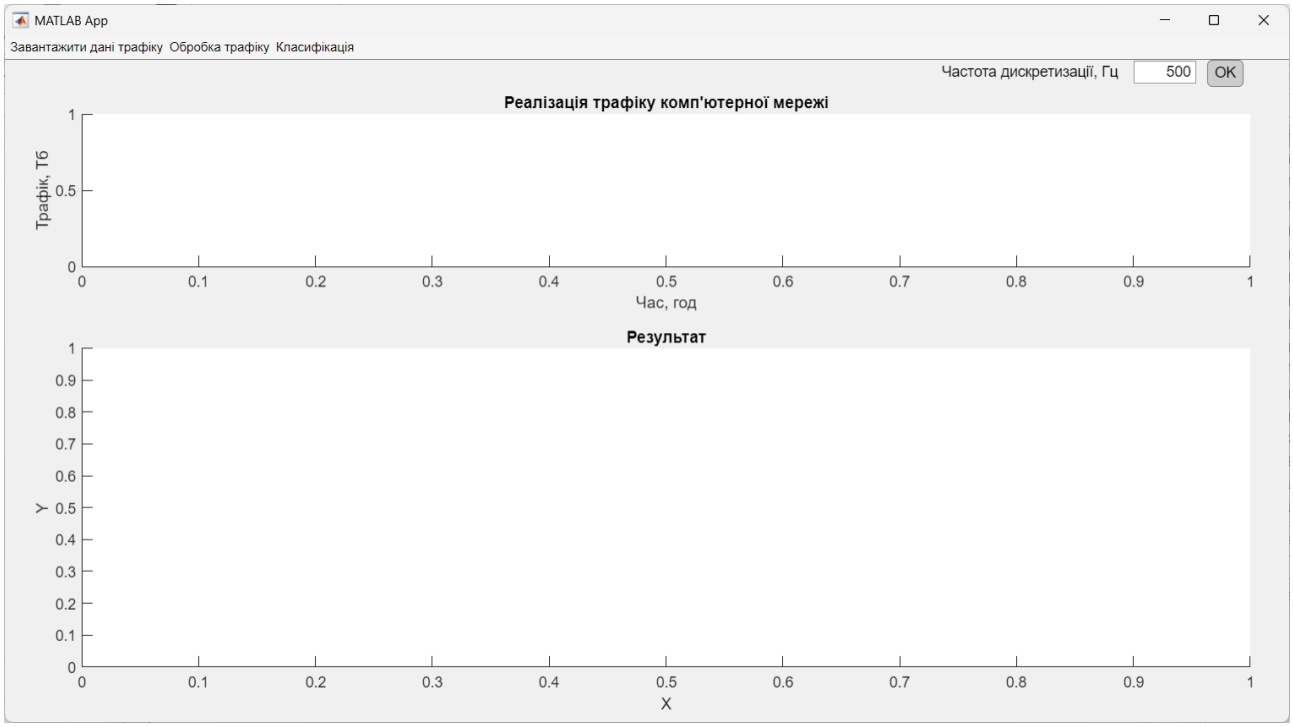


Рис.3.17. Загальний вигляд інтерфейсу автоматизованої системи

Результати роботи автоматизованої системи з графічним інтерфейсом зображено на рис.3.18-3.21.

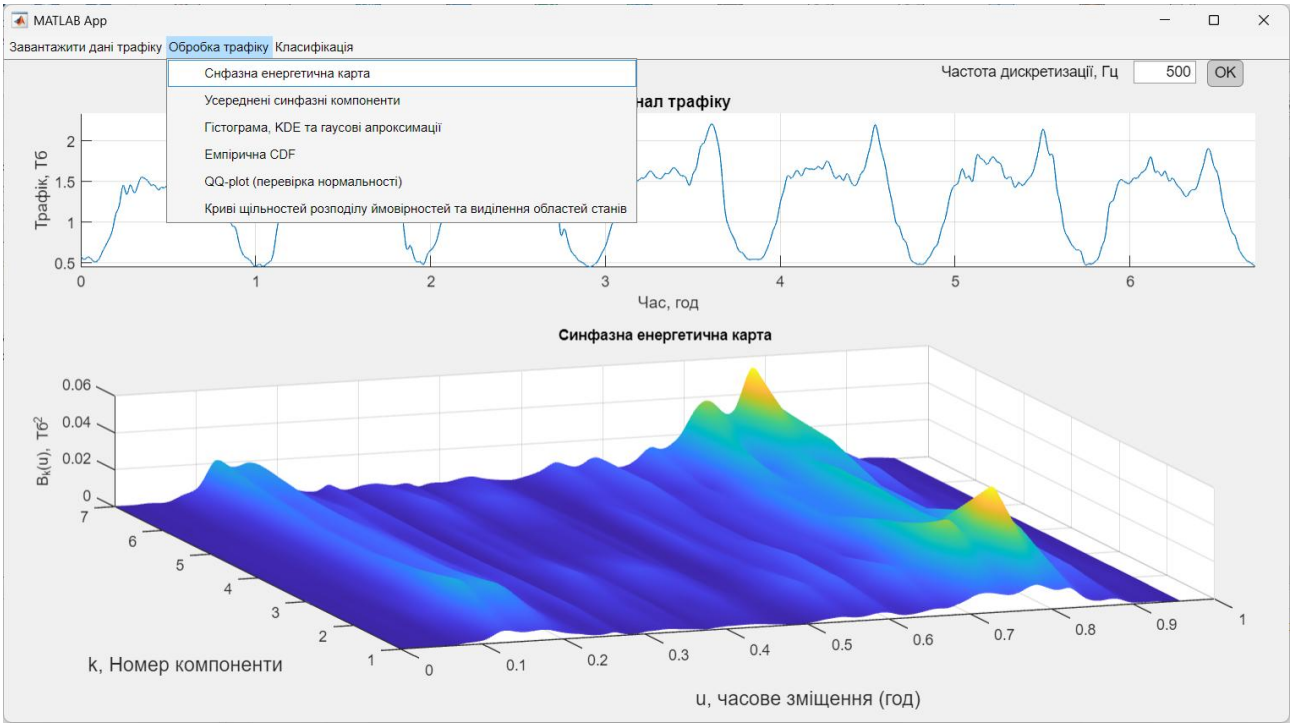


Рис.3.18. Результат роботи автоматизованої системи

(синфазна енергетична карта)

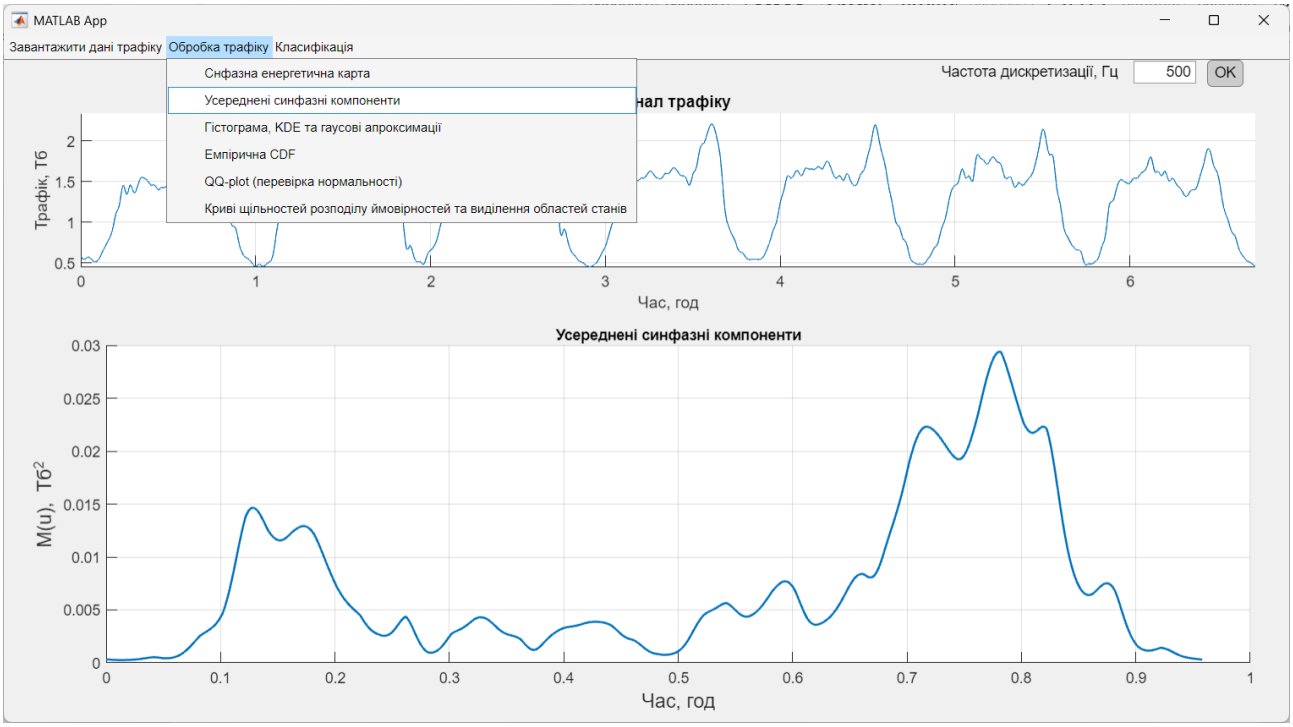


Рис.3.19. Результат роботи автоматизованої системи (усереднені синфазні компоненти)

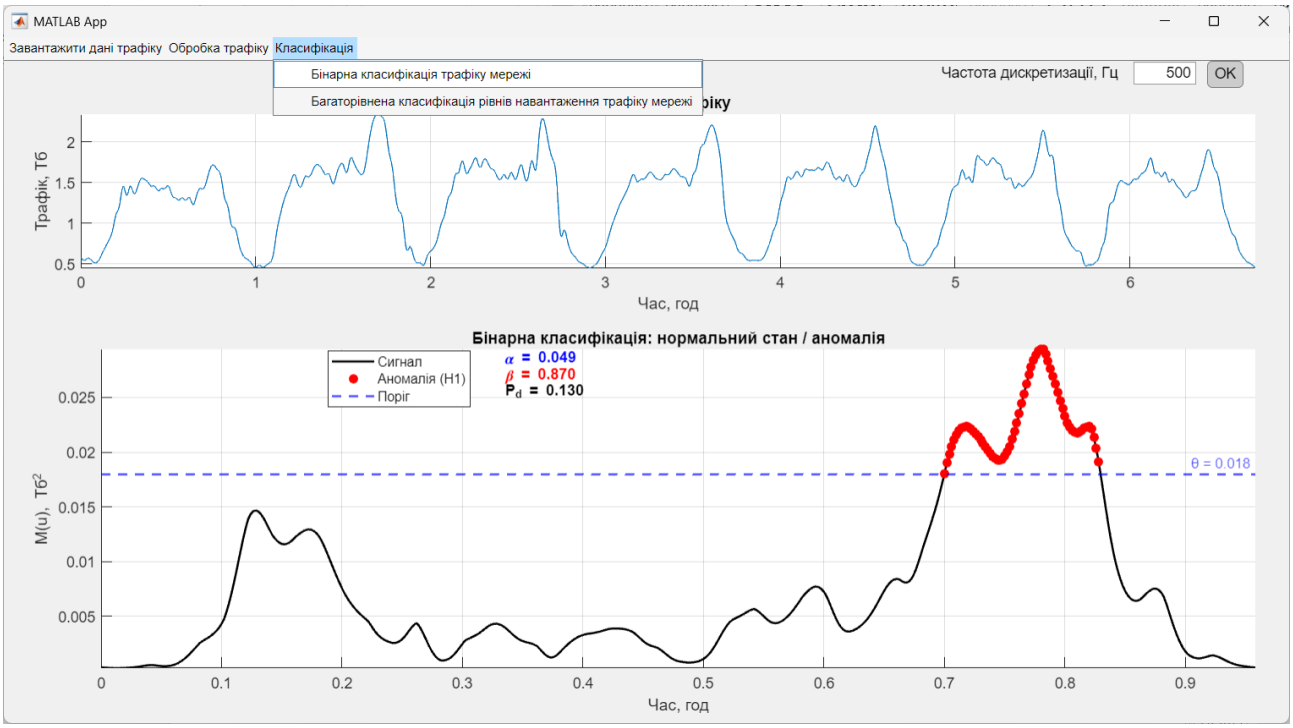


Рис.3.20. Результат роботи автоматизованої системи (бінарна класифікація)

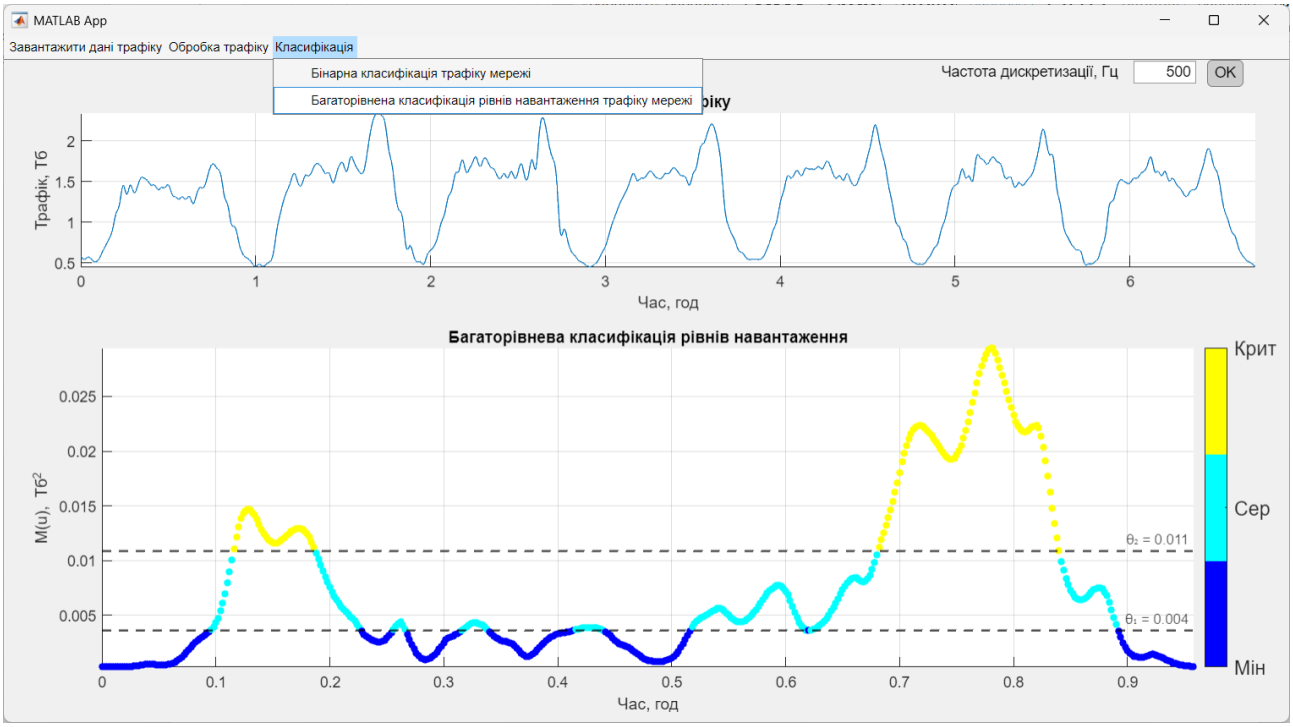


Рис.3.21. Результат роботи автоматизованої системи (багаторівнева класифікація)

Функціональність системи:

– Завантаження даних навантаження трафіку мережі.

– Синфазна обробка.

– Усереднення результатів синфазної обробки.

– Емпіричний розподіл та апроксимації.

– Застосування критерію Неймана–Пірсона для визначення оптимального порога.

– Формування бінарної 2-класової класифікації.

– Формування бінарної багаторівневої 3-класової класифікації.

– Візуалізація результатів.

Отже, результати роботи системи підтверджують можливість щодо автоматизації процесу синфазної класифікації рівнів мережевого навантаження.

3.6 Висновки до розділу 3

У розділі розроблено автоматизовану систему синфазної класифікації рівнів мережевого навантаження у вигляді програмного забезпечення, що реалізує запропонований у попередньому розділі математичний апарат і методологію статистичного аналізу.

Побудована архітектура системи має модульну структуру, яка забезпечує послідовність етапів обробки трафіку: від завантаження та попередньої нормалізації даних до візуалізації результатів класифікації. Реалізовано ключові функціональні блоки – синфазна обробка, усереднення гармонічних компонент (кореляційних компонент), побудову емпіричних розподілів, застосування критерію Неймана–Пірсона, а також формування бінарної та багаторівневої класифікації станів мережі.

Програмна реалізація у середовищі MATLAB забезпечила зручність чисельних розрахунків, можливість швидкої візуалізації та перевірки коректності алгоритмів. Тестування програмного забезпечення показало високу достовірність виявлення аномальних станів і правильність розмежування між мінімальним, нормальним та критичним навантаженням.

Система автоматично визначає порогові значення класифікації, контролює ймовірності помилкових рішень (α, β) та забезпечує відображення результатів у зручному графічному форматі. Це дозволяє оперативно аналізувати динаміку навантаження й приймати рішення щодо стану мережі у реальному часі.

Таким чином, у розділі реалізовано практичну частину дослідження – створено ефективне програмне забезпечення для автоматизованого моніторингу та синфазної класифікації рівнів мережевого навантаження, яке підтвердило працездатність і ефективність запропонованої математичної моделі та методу.

РОЗДІЛ 4
ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці

Охорона праці як науково-практична дисципліна формується на перетині правових, технічних, санітарно-гігієнічних, соціальних та інженерних знань, спрямованих на створення безпечних і нешкідливих умов праці для працівників різних сфер виробництва та інтелектуальної діяльності. Українське законодавство визначає охорону праці як складну систему превентивних заходів, що забезпечують збереження здоров'я та працездатності людини. Відповідно до Закону України «Про охорону праці» № 2694-XII, роботодавець зобов'язаний створити робоче середовище, у якому відсутні небезпечні та шкідливі фактори або вони зведені до мінімальних можливих рівнів за допомогою організаційно-технічних рішень, засобів колективного та індивідуального захисту, а також відповідних режимів праці та відпочинку.

У контексті професійної діяльності у сфері інформаційних технологій особливе місце займають умови праці, пов'язані з тривалим використанням комп'ютерів, сертифікованої електронно-обчислювальної техніки, мережевого обладнання, систем збору, аналізу та оброблення інформації. Виробниче середовище ІТ-працівника суттєво відрізняється від умов класичної промислової діяльності, проте воно також характеризується рядом факторів, які, за відсутності контролю, можуть негативно впливати на здоров'я і працездатність. Основними такими факторами є напруженість зорової роботи, статичні навантаження, психоемоційні напруження, електричні ризики, пожежна небезпека, мікрокліматичні параметри приміщення, якість освітлення та санітарно-гігієнічні умови. Їхній вплив регламентується низкою українських нормативних документів, серед яких Державні санітарні норми та правила ДСанПіН 3.3.2-007-98, ДСанПіН 3.3.2-017-2017, ДБН В.2.5-28:2018, НПАОП 40.1-1.21-98, Правила пожежної безпеки в Україні (НАПБ А.01.001-2014) та інші

акти, що визначають вимоги щодо безпечної експлуатації обладнання та допустимих рівнів впливу виробничих факторів.

Одним з найважливіших положень державної політики в галузі охорони праці є превентивність, тобто запобігання небезпечним ситуаціям до того, як вони можуть спричинити негативні наслідки для здоров'я працівника. Це повною мірою стосується фахівців, робота яких пов'язана з автоматизованими системами, адже вони перебувають у зоні тривалого впливу електромагнітного випромінювання, шумових коливань, вібрацій, статичного м'язового напруження та психоемоційного стресу. Згідно з вимогами ДСанПіН 3.3.2-007-98, тривалість безперервної роботи з відеодисплейними терміналами має бути структурована з урахуванням особливостей когнітивних навантажень, які можуть сприяти розвитку синдрому зорової втоми, порушень акомодації, головного болю, астенічних станів та загального виснаження.

У виробничих умовах ІТ-підрозділів надзвичайно важливим є забезпечення правильних параметрів мікроклімату. Відповідно до вимог ДСанПіН 3.3.6.042-99 та ДБН В.2.5-67:2013 «Опалення, вентиляція та кондиціонування», температура, вологість та швидкість руху повітря мають перебувати у межах комфортних значень. Показники, нижчі або вищі за рекомендовані, знижують концентрацію уваги, впливають на роботу серцево-судинної системи та терморегуляцію організму, а також спричиняють втому й підвищений ризик захворювань. Нормативи передбачають, що у приміщеннях з комп'ютерною технікою температура має підтримуватися на рівні 22–24 °С у холодний період року та 23–25 °С у теплий. Відносна вологість має становити 40–60 %, а швидкість руху повітря не повинна перевищувати 0,1–0,2 м/с. Погіршення мікроклімату може негативно впливати не лише на стан працівників, а й на функціонування електронного обладнання, що підвищує загрози його виходу з ладу.

Особливе значення для ІТ-сфери має нормативне забезпечення якості освітлення. ДБН В.2.5-28:2018 визначає мінімально допустимі рівні освітленості робочих зон, а також вимагає забезпечення рівномірності світлового потоку й відсутності відблисків на поверхні екранів. Підвищене або недостатнє освітлення може викликати перенапруження органів зору, що негативно позначається на

здатності працівника до тривалої когнітивної роботи. Дослідження показують, що оптимальна освітленість для користувачів комп'ютерної техніки становить 300–500 лк. Водночас важливим є правильне розташування штучних джерел світла, щоб уникнути засліплення та дискомфортних світлових контрастів, які спричиняють додаткові зорові навантаження.

Електробезпека є ще одним невід'ємним аспектом охорони праці у процесі роботи з інформаційними системами. НПАОП 40.1-1.21-98 чітко регламентує вимоги до експлуатації електроустановок, визначає порядок проведення технічних оглядів, заземлення, перевірки ізоляції та використання захисних пристроїв. Згідно із законодавством, усі електроустановки мають бути оснащені пристроями автоматичного вимкнення електроживлення при короткому замиканні або перевантаженні мережі. Особливу увагу слід приділяти працездатності кабельних систем, які, відповідно до ДСТУ EN 50575:2018, повинні відповідати класам пожежної небезпеки, що унеможливлюють поширення полум'я у разі займання.

Комп'ютерні системи створюють додаткові ризики виникнення пожеж. Високе тепловиділення серверного та мережевого обладнання, наявність великої кількості кабельних трас і стабілізаторів напруги підвищують пожежну небезпеку приміщень. Правила пожежної безпеки в Україні (НАПБ А.01.001-2014) встановлюють порядок облаштування приміщень засобами виявлення пожежі, передбачають використання вуглекислотних вогнегасників для гасіння електрообладнання під напругою, регламентують правила розміщення електрощитового обладнання, а також обов'язковість наявності планів евакуації та вільного доступу до евакуаційних виходів.

Не менш важливим чинником є психологічне та емоційне навантаження працівників ІТ-сфери. Постійна робота з великими обсягами інформації, складні інтелектуальні завдання, робота у стислі терміни та необхідність постійного контролю за процесами створюють сприятливі умови для розвитку професійного стресу. На рівні організації доцільно впроваджувати заходи інформаційної ергономіки, що сприяють оптимізації робочих процесів, зменшенню монотонності діяльності та покращенню емоційного стану працівників. Медичні рекомендації

Міністерства охорони здоров’я України вказують на необхідність чіткої регламентації робочого часу за комп’ютером, проведення регулярних перерв, спрямованих на відпочинок зорового апарату, та виконання комплексу вправ для м’язів, які перебувають у напрузі протягом робочого дня.

Санітарно-гігієнічні вимоги, встановлені ДСанПіН 3.3.2-017-2017, передбачають забезпечення достатнього об’єму приміщення для кожного працівника, належний стан повітряного середовища, регулярне прибирання приміщень, дотримання норм щодо використання оздоблювальних матеріалів і меблів. Приміщення, у яких встановлене комп’ютерне та серверне обладнання, повинні мати площу не менше 6 м² на одного працівника та об’єм повітря не менше 20 м³. Важливою вимогою є забезпечення кратності повітрообміну, яка має становити не менше двох обмінів повітря за годину. Створення таких умов сприяє профілактиці захворювань, пов’язаних із тривалим перебуванням у замкнених просторах, та забезпечує належне функціонування електронних систем.

Отже, охорона праці у сфері інформаційних технологій є комплексним процесом, що включає нормативно-правове регламентування, інженерне забезпечення, організаційні заходи, психологічну підтримку та санітарно-гігієнічні норми. Дотримання вимог Законів України, НПАОП, ДБН, ДСанПіН та інших нормативів дозволяє створити безпечне робоче середовище, яке мінімізує вплив шкідливих факторів і сприяє підвищенню продуктивності праці й загального рівня професійної безпеки.

4.2 Безпека в надзвичайних ситуаціях

Ефективна організація евакуації людей під час надзвичайних ситуацій є одним із ключових елементів забезпечення комплексної безпеки сучасних підприємств, особливо тих, діяльність яких пов’язана з експлуатацією інформаційно-комп’ютерних систем. В умовах зростання технологічної складності виробничих процесів і збільшення кількості електронно-обчислювальної техніки ризик виникнення небезпечних ситуацій техногенного чи природного походження також

зростає. Саме тому на підприємствах повинна бути створена система організаційних заходів, спрямована на забезпечення швидкої та безпечної евакуації працівників з будівлі у випадку пожежі, аварії, загрози вибуху, проникнення диму, руйнування конструкцій або інших надзвичайних подій.

Основні вимоги до організації евакуації визначаються Кодексом цивільного захисту України та ДБН В.1.1-7:2016 «Пожежна безпека об’єктів будівництва». Згідно з цими документами, роботодавець зобов’язаний забезпечити відповідність приміщень і будівлі нормам пожежної та техногенної безпеки, створити умови для безпечного переміщення людей до евакуаційних виходів, обладнати шляхи евакуації відповідними покажчиками та освітленням, а також розробити та затвердити план евакуації. План евакуації має бути наочним, містити інформацію про маршрути виходу, місця розташування вогнегасників, систем пожежогасіння, точок збору та дії працівників у разі НС. Він повинен бути розміщений у доступних та добре видимих місцях — у коридорах, холах, поблизу службових приміщень та біля входу до будівлі.

Важливою вимогою до організації евакуації є забезпечення належного технічного стану шляхів виходу. Відповідно до НАПБ А.01.001-2014 («Правила пожежної безпеки в Україні»), на шляхах евакуації забороняється розміщення меблів, обладнання, коробок, кабелів та будь-яких інших предметів, що можуть перешкоджати руху людей або знижувати пропускну здатність проходів. Евакуаційні двері повинні бути легко відчинятися у напрямку виходу, без додаткових перешкод, замків або сигналізацій, які не дозволяють швидко залишити приміщення. Усі евакуаційні виходи повинні бути позначені світловими покажчиками відповідно до вимог ДСТУ ISO 7010:2019, що гарантує можливість швидкого орієнтування навіть за умов задимлення або відсутності електроживлення.

У разі виникнення надзвичайної ситуації ключову роль відіграє психологічна готовність персоналу. Паніка та неорганізованість можуть суттєво збільшити ризик травмування, навіть якщо технічні умови евакуації виконані належним чином. З цього погляду важливими є регулярні протиаварійні тренування, що проводяться відповідно до вимог НПАОП 0.00-4.21-04 щодо навчання та інструктажів з охорони праці. Такі тренування включають як теоретичну частину — ознайомлення з

можливими видами НС, їхніми ознаками, джерелами небезпеки — так і практичну — відпрацювання реальних маршрутів евакуації, використання засобів пожежогасіння, виконання команд відповідальних осіб. Практична складова є особливо важливою, оскільки формує у працівників навички швидкого прийняття рішень у стресових умовах, що може врятувати життя.

Не менш важливою складовою системи безпеки під час НС є система оповіщення. Відповідно до вимог ДСТУ EN 54 та Кодексу цивільного захисту України, на підприємствах повинні бути встановлені системи пожежного, звукового та мовного оповіщення, здатні негайно інформувати працівників про небезпеку. У серверних, лабораторіях або приміщеннях з особливо чутливим обладнанням можуть встановлюватися автоматичні датчики диму, температури та газового пожежогасіння, які забезпечують раннє виявлення небезпечних факторів. Своєчасне оповіщення дозволяє організувати евакуацію працівників до того, як загроза стане критичною, та уникнути масового скупчення людей у зонах підвищеної небезпеки.

У приміщеннях з комп’ютерною технікою та серверним обладнанням особливе значення має надійність комунікаційних мереж, адже під час НС саме вони забезпечують передачу сигналів оповіщення, координацію дій персоналу та передачу даних до аварійних служб. Тому у межах системи евакуації необхідно передбачити дублювання каналів зв’язку та наявність резервних джерел електроживлення, що узгоджується з вимогами НПАОП 40.1-1.21-98 щодо експлуатації електроустановок. Безперебійні джерела живлення дозволяють системам оповіщення працювати навіть у разі повного вимкнення електроенергії.

Одним із ключових організаційних рішень є призначення відповідальних осіб за евакуацію. На підприємстві створюється комісія або група швидкого реагування, яка забезпечує координацію дій працівників у разі НС. Вона відповідає за огляд приміщень, контроль стану шляхів евакуації, наявність засобів пожежогасіння, підтримання планів евакуації в актуальному стані та взаємодію з аварійними службами. Під час НС саме ці особи організовують рух людей, забезпечують перевірку приміщень та повідомляють про завершення евакуації.

Особливої уваги потребує питання евакуації людей з обмеженими фізичними можливостями. ДБН В.2.2-40:2018 висуває вимоги щодо облаштування будівель спеціальними пандусами, розширеними проходами, засобами допомоги для пересування. Такі заходи дозволяють забезпечити безпеку всіх категорій працівників та виключити можливість того, що хтось буде заблокований у небезпечній зоні.

Загалом організація евакуації на підприємстві, яке використовує великі обсяги комп'ютерної техніки та інформаційних систем, повинна базуватися на комплексному підході, що поєднує технічні, організаційні, регламентні та психологічні заходи. Дотримання вимог чинних нормативних документів України дозволяє не лише мінімізувати наслідки надзвичайних ситуацій, а й створити ефективну систему цивільного захисту, орієнтовану на запобігання загрозам і забезпечення безпеки персоналу.

4.3 Висновки до розділу 4

У розділі встановлено, що ефективна охорона праці та безпека в надзвичайних ситуаціях є необхідною умовою стабільної роботи підприємств ІТ-сфери. Аналіз довів, що робота з комп'ютерною технікою пов'язана з дією зорових, статичних, психоемоційних, електричних і пожежних факторів, вплив яких регламентується чинними нормативами України. Дотримання санітарно-гігієнічних, ергономічних та технічних вимог забезпечує здоров'я працівників і працездатність обладнання.

У сфері реагування на надзвичайні ситуації ключовими є правильно організовані шляхи евакуації, системи оповіщення, технічний стан будівлі та підготовленість персоналу. Виконання вимог Кодексу цивільного захисту України і відповідних ДБН та НПАОП дозволяє мінімізувати ризики та забезпечити захист людей.

Загалом охорона праці та безпека в надзвичайних ситуаціях є комплексом взаємопов'язаних заходів, що гарантують безпечні умови праці, збереження здоров'я персоналу та стійкість функціонування інформаційних систем підприємства.

ЗАГАЛЬНІ ВИСНОВКИ

У ході виконання роботи досягнуто поставлену мету – розроблено автоматизовану систему синфазної класифікації рівнів мережевого навантаження, яка поєднує математичне моделювання періодично-корельованих випадкових процесів, синфазну обробку сигналів і статистичні критерії прийняття рішень.

У межах першої задачі проаналізовано існуючі методи класифікації мережевого навантаження, серед яких статистичні, спектральні, фрактальні, методи машинного навчання, марковські та гібридні. Встановлено, що традиційні методи характеризуються високою швидкодією, але низькою точністю при нестаціонарних процесах, тоді як інтелектуальні підходи забезпечують високу точність, проте потребують значних ресурсів. Це визначило необхідність створення адаптивного методу, який поєднує статистичну строгость із фазово-енергетичним аналізом сигналу.

За другою задачею побудовано математичну модель мережевого трафіку на основі апарату ПКВП. Модель дозволяє одночасно враховувати циклічні закономірності (пов’язані з добовими чи тижневими коливаннями навантаження) та стохастичні флуктуації трафіку, що забезпечує більш точний опис нестаціонарних процесів у мережах.

У третій задачі розроблено метод синфазної обробки сигналів, який базується на побудові гармонічних і кореляційних компонент енергетичних характеристик трафіку. Запропоновано процедуру усереднення по гармонічних складових, що підвищує стійкість до шуму та зменшує вплив короткочасних флуктуацій. У результаті отримано інтегральний показник синфазної енергії, придатний для подальшої класифікації станів системи.

Виконуючи четверту задачу, сформульовано статистичні критерії прийняття рішень для класифікації рівнів навантаження. Застосовано критерій Неймана–Пірсона**,** який забезпечує мінімізацію ймовірності пропуску критичного стану при контрольованому рівні хибної тривоги. Визначено порогові умови класифікації, що дозволяють автоматично розділяти нормальний і аномальний стани мережі.

Відповідно до п’ятої задачі реалізовано алгоритми бінарної та багатокласової класифікації рівнів навантаження. Запропоновано адаптивний підхід до формування порогів на основі статистичних параметрів розподілу (середнього та стандартного відхилення), що дозволяє визначати мінімальні, нормальні та критичні режими роботи мережі без необхідності попереднього навчання або апріорних моделей.

У рамках шостої задачі створено програмний модуль автоматизованої системи синфазної класифікації у середовищі MATLAB. Розроблене програмне забезпечення реалізує усі етапи обробки: завантаження даних, синфазний аналіз, побудову емпіричних розподілів, розрахунок порогів класифікації та візуалізацію результатів. Проведене тестування підтвердило ефективність і достовірність методу, високу чутливість до змін режимів навантаження та стійкість до шумових спотворень.

У результаті виконаного дослідження:

– обґрунтовано доцільність використання моделі ПКВП для опису мережевого трафіку;

– розроблено синфазний метод класифікації рівнів навантаження;

– реалізовано математичні алгоритми та програмне забезпечення, що забезпечує класифікацію у реальному часі з керованою точністю.

Запропонований підхід може бути використаний у системах контролю, діагностики, прогнозування та забезпечення надійності комп’ютерних мереж, а також при побудові інтелектуальних засобів керування якістю обслуговування (QoS).

ПЕРЕЛІК ПОСИЛАНЬ

1. Davydovskyi Y., Reva O., Artiukh O., Kosenko V. *Simulation of computer network load parameters over a given period of time.* // *Innovative Technologies and Scientific Solutions for Industries.* — 2019. — № 9. — С. 72–81. — DOI: 10.30837/2522-9818.2019.9.072.

2. Pustovoitov P., Voronets V., Voronets O., Sokol H., Okhrymenko M. *Assessment of QoS indicators of a network with UDP and TCP traffic under a node peak load mode.* // *Eastern-European Journal of Enterprise Technologies.* — 2024. — № 1 (4). — DOI: 10.15587/1729-4061.2024.299124.

3. Ertam F., Avcı E. *Network Traffic Classification via Kernel Based Extreme Learning Machine.* // *International Journal of Intelligent Systems and Applications in Engineering.* — 2017. — DOI: 10.18201/ijisae.267522.

4. Shi L., Li Y., Loo B. T., Alur R. *Network Traffic Classification by Program Synthesis.* // *Lecture Notes in Computer Science (TACAS).* — Springer, 2021. — Vol. 12651. — DOI: 10.1007/978-3-030-72016-2_23.

5. Mandelbrot B. B. *The Fractal Geometry of Nature.* — New York : W. H. Freeman, 1982. — 480 p.

6. Norros I. *On the use of fractional Brownian motion in the theory of connectionless networks.* // *IEEE Journal on Selected Areas in Communications.* — 1995. — Vol. 13, No. 6. — P. 953–962.

7. Wang Y., Chen X., Liu J. *Deep learning for network traffic classification: A survey.* // *Computer Networks.* — 2021. — Vol. 197. — P. 108–118.

8. Kim H., Park J. *Hybrid CNN–LSTM approach for real-time network load classification.* // *IEEE Access.* — 2022. — Vol. 10. — P. 77612–77625.

9. Li Y., Wang C. *Network load prediction based on Markov processes.* // *Journal of Network and Computer Applications.* — 2020. — Vol. 154. — P. 102–118.

10. Jiang Q., Liu F. *Adaptive Ensemble Classification for Network Traffic Identification.* // *IEEE Transactions on Network Science and Engineering.* — 2023. — Vol. 10. — P. 2443–2456.

11. Aceto G., Ciuonzo D., Montieri A., Pescapè A. Toward Effective Mobile Encrypted Traffic Classification through Deep Learning // *Neurocomputing*. – 2020. – Vol. 409. – DOI: 10.1016/j.neucom.2020.05.036.

12. Feldmann A. Self-similar Network Traffic and Performance Evaluation // Park K. – 2000. – DOI: 10.1002/047120644X.ch15.

13. Li G., Knoop V., Lint J.W.C. Estimate the limit of predictability in short-term traffic forecasting: An entropy-based approach // *Transportation Research Part C: Emerging Technologies*. – 2022. – Vol. 138. – P. 103607. – DOI: 10.1016/j.trc.2022.103607.

14. Chen X., Irie K., Banks D., Haslinger R., Thomas J., West M. Scalable Bayesian Modeling, Monitoring and Analysis of Dynamic Network Flow Data // *Journal of the American Statistical Association*. – 2016. – Vol. 113. – DOI: 10.1080/01621459.2017.1345742.

15. Liu K., Kuo J.-Y., Yeh K., Chen C.-W., Liang H.-H., Sun Y.-H. Using fuzzy logic to generate conditional probabilities in Bayesian belief networks: a case study of ecological assessment // *International Journal of Environmental Science and Technology*. – 2013. – Vol. 12. – DOI: 10.1007/s13762-013-0459-x.

16. Ren W., Ma O., Ji H., Liu X. Human Posture Recognition Using a Hybrid of Fuzzy Logic and Machine Learning Approaches // *IEEE Access*. – 2020. – P. 1-1. – DOI: 10.1109/ACCESS.2020.3011697.

17. Khvostivska L., Khvostivskyi M., Dediv I. Mathematical, algorithmic and software support for signals wavelet detection in electronic communications. Proceedings of the 2nd International Workshop on Computer Information Technologies in Industry 4.0 (CITI 2024). CEUR Workshop Proceedings. Ternopil, Ukraine, June 14-16, 2024. Vol. 3742. P.223-234. ISSN 1613-0073.

18. Khvostivska L., Khvostivskyi M., Dediv I., Yatskiv V., Palaniza Y. Method, Algorithm and Computer Tool for Synphase Detection of Radio Signals in Telecommunication Networks with Noises. Proceedings of the 1st International Workshop on Computer Information Technologies in Industry 4.0 (CITI 2023). CEUR Workshop Proceedings. Ternopil, Ukraine, June 14-16, 2023. P.173-180. ISSN 1613-0073.

19. Khvostivskyy M., Osukhivska H., Khvostivska L., Lobur T., Velychko D. Mathematical modelling of daily computer network traffic. ITTAP-2021: Information Technologies: Theoretical and Applied Problems. The 1st International Workshop (November 16-18, 2021). Ternopil, UKRAINE. P.107-111. DOI: 10.1425/jsdtl.

20. Khvostivskyy M., Osukhivska H., Khvostivska L., Lobur T., Velychko D., Lupenko S., Hovorushchenko T. Mathematical modelling of daily computer network traffic. The 1st International Workshop on Information Technologies: Theoretical and Applied Problems, ITTAP 2021. CEUR Workshop Proceedings. Ternopil, Ukraine, November 16-18, 2021. Vol. 3039. P.107-111. ISSN 1613-0073.

21. Liliya Khvostivska, Mykola Khvostivskyy, Vasyl Dunetc, Iryna Dediv. Mathematical and Algorithmic Support of Detection Useful Radiosignals in Telecommunication Networks. Proceedings of the 2nd International Workshop on Information Technologies: Theoretical and Applied Problems (ITTAP 2022). Ternopil, Ukraine, November 22-24, 2022. P.314-318. ISSN 1613-0073.

22. Khvostivskyi, M., Khvostivska, L., Dediv, I., Yatskiv, V., & Kuchvara, O. (2025). Method and computer tool for synphase prediction of computer network traffic load level. Proceedings of the 3rd International Workshop on Computer Information Technologies in Industry 4.0 (CITI 2025), 18–19 September 2025, Ternopil, Ukraine. CEUR Workshop Proceedings (Vol. 4057, pp. 261–269). CEUR-WS. ISSN 1613-0073.

23. Khvostivska L., Herasymchuk Yu., Zemledukh V,. Vorobets I. Method for Automated Detection of Load Levels in Computer Networks. Collection of Scientific Papers with the Proceedings of the 2nd International Scientific and Practical Conference «Global Trends in Science: Research, Innovation and Development» (September 29 – October 1, 2025, Varna, Bulgaria). European Open Science Space, 2025. P.145-149. DOI 10.70286/EOSS-29.09.2025. ISBN 979-8-89704-966-0.

24. Voloshchuk A., Osukhivska H., Khvostivskyi M., Sverstiuk A. (2025). Analysis of electricity consumption using the component method of periodically correlated random processes. *Computer Systems and Information Technologies*, (3), 74–82. https://doi.org/10.31891/csit-2025-3-8. ISSN 2710-0766.

25. Хвостівський В., Осухівська Г., Хвостівська Л. Програмне забезпечення системи опрацювання мережевого трафіку. Матеріали IX науково-технічної конференції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя, (Тернопіль, 8-9 грудня 2021р.). Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2020. С.102.

26. Методичні рекомендації до виконання, оформлення та захисту кваліфікаційних робіт для здобувачів другого (магістерського) рівня вищої освіти за спеціальністю 172 «Електронні комунікації та радіотехніка» галузі знань 17 «Електроніка, автоматизація та електронні комунікації»/уклад.: Дунець В.Л., Хвостівська Л.В., Дедів І.Ю. Тернопіль: ТНТУ, 2024. 56 с.

27. Стручок В.С. Безпека в надзвичайних ситуаціях. Методичний посібник для здобувачів освітнього ступеня «магістр» всіх спеціальностей денної бо та заочної (дистанційної) форм навчання / В.С.Стручок. Тернопіль: ФОП Паляниця В. А., 2022. 156 с.

ДОДАТКИ

ДОДАТОК А

Копія тези

UDC 01.1

Collection of Scientific Papers with the Proceedings of the 2nd International Scientific and Practical Conference «Global Trends in Science: Research, Innovation and Development» (September 29 – October 1, 2025, Varna, Bulgaria). European Open Science Space, 2025. 166 p.

ISBN 979-8-89704-966-0 (series)
DOI 10.70286/EOSS-29.09.2025

The conference is included in the Academic Research Index ReserchBib International catalog of scientific conferences.

The conference is registered in the database of scientific and technical events of UkrISTEI to be held on the territory of Ukraine (Certificate №552 dated 16.06.2025).

ISBN 979-8-89704-966-0 (series)

EUROPEAN OPEN SCIENCE SPACE

Proceedings of the 2nd International Scientific and Practical Conference
"Global Trends in Science: Research, Innovation and Development"
September 29 – October 1, 2025
Varna, Bulgaria

EUROPEAN OPEN SCIENCE SPACE

Section: Technical Sciences

METHOD FOR AUTOMATED DETECTION OF LOAD LEVELS IN COMPUTER NETWORKS

Khvostivska Liliia
Ph.D., Associate Professor
Herasymchuk Yurii
Student
Zemledukh Vladyslav
Student
Vorobets Ihor
Student
Ternopil Ivan Puluj National Technical University, Ukraine

The rapid development of information technologies [1] causes a significant increase in the load on computer networks. Modern telecommunication systems deal with traffic that is formed under the influence of cyclical patterns and random fluctuations. To ensure stable operation of the network, it is necessary not only to predict its states, but also to automatically detect load levels, which allows for timely identification of critical modes and make proactive decisions on resource management.

Traditional methods are based [2] on threshold criteria that do not take into account the stochastic nature of traffic and do not provide control over the probability of classification errors. In this paper, an approach to automated detection of load levels is proposed, which is based on the periodically correlated stochastic process (PCSP) model, in-phase/component signal analysis, and the Neumann–Pearson criterion.

Network load classification methods can be conditionally divided into several groups:

– Stochastic models (Poisson, Markov, ARIMA) describe randomness, but do not reproduce cyclicality.

– Fractal models (fBm) take into account self-similarity, but do not take into account diurnal fluctuations [5].

– Machine learning methods [3] provide high accuracy, but require large computational resources.

– Hybrid approaches combine different models, but are difficult to implement in practice [6].

The disadvantage of most of them is the use of simple thresholding schemes, which reduces the reliability of classification. The proposed automated detection system overcomes these limitations by integrating in-phase/component analysis and the Neumann-Pearson criterion.

Proceedings of the 2nd International Scientific and Practical Conference
"Global Trends in Science: Research, Innovation and Development"
September 29 – October 1, 2025
Varna, Bulgaria
EUROPEAN OPEN SCIENCE SPACE

Traffic load is considered as a periodically correlated stochastic process:

$$\xi(t) = S(t) + X(t), \quad t \in \mathrm{R} \tag{1}$$

where $S(t)$ – a deterministic periodic component with a daily period T, reflecting a repeating cyclicity;

$X(t)$ – stochastic component.

Thus, the model combines the regular structure of traffic and its random nature.

To move from the model to the processing tools, the random component $\xi(t)$ is given in the form of a harmonic expansion:

$$\xi(t) = \sum_{k \in Z} \xi_k(t) e^{-\frac{j2\pi k}{T}}, \quad t \in \mathrm{R} \tag{2}$$

where $\xi_k(t)$ – stochastic coefficients that vary with time, and exponential multipliers describe the periodic structure of the process.

This approach allows us to represent the random part as a superposition of in-phase harmonic components.

In-phase and component methods use these harmonic components to construct correlation features.

For centered traffic signals, a spectral-correlation representation is calculated for harmonic number k:

$$\hat{B}_k(u) = \frac{1}{N} \sum_{t \in Z} \overset{0}{\xi}(t) \overset{0}{\xi}(t-u) e^{-\frac{j2\pi k}{T}}, \tag{3}$$

where $\overset{0}{\xi}(t)$ – centered traffic signal relative to the mathematical expectation, $\overset{0}{\xi}(t) = \xi(t) - E[\xi(t)]$;

N – sample length; u – time shift;

k – harmonic number (correlation component number); Z – indexing area.

The correlation components $\hat{B}_k(u)$, which are calculated for centered traffic signals, reflect the time-frequency characteristics of the network load. However, in real data there are

– noise components (random fluctuations of short duration);

– anomalous peaks (sudden jumps due to local events) [4, 7];

– limited sample size, leading to statistical instability of estimates.

If raw correlation components $\hat{B}_k(u)$ are used, the classification results may be overly sensitive to random fluctuations.

To avoid this, a procedure of averaging over components was introduced:

$$\overline{C}(u) = \frac{1}{N} \sum_{k=1}^{K} B_k(u), \tag{4}$$

where $\hat{B}_k(u)$ – estimate of the correlation component on the u-th interval;

K – number of averaging components.

Proceedings of the 2nd International Scientific and Practical Conference
"Global Trends in Science: Research, Innovation and Development"
September 29 – October 1, 2025
Varna, Bulgaria
EUROPEAN OPEN SCIENCE SPACE

That is, averaging correlation components is a kind of «filter» of statistical features: it cuts off short-term random fluctuations and leaves long-term patterns that are truly informative for further classification of load levels.

As a result, a feature vector is formed:

$$C(u) = \left(\overline{C}(u_1), \overline{C}(u_2), ..., \overline{C}(u_m)\right), \tag{5}$$

which preserves the temporal dynamics of the load and is the basis for classification.

Automated detection of load levels in computer networks is formulated as a statistical hypothesis testing problem. In the simplest case, binary classification is considered:

– H_0: the load is average (normal network operation mode);

– H_1: the load is abnormal (minimum or critical).

In order to decide between hypotheses, a likelihood ratio is introduced:

$$\Lambda(C(u)) = \frac{f(C(u) \mid H_1)}{f(C(u) \mid H_0)}, \tag{6}$$

where $C(u)$ – vector of averaged in-phase/component correlation features;

$f(C(u) \mid H_1)$, $f(C(u) \mid H_0)$ – probability density of the appearance of these features when the hypotheses H_0 and H_1 are true.

The decision (classification) rule has the form:

$$\Lambda(C(u)) \underset{H0}{\overset{H1}{\lessgtr}} \eta, \tag{7}$$

where η – threshold value determined from the condition of controlling the probability of incorrect decisions (determined by the Neumann-Pearson criterion) [8].

This approach allows you to minimize the probability of missing a dangerous condition while controlling the probability of false alarms.

According to the Neumann–Pearson criterion, the optimal rule is considered to be one that minimizes the probability of missing an abnormal state (type II error):

$$\beta = P(\textit{рішення } H0 | H1) = \int_{\Lambda(C(u))<\eta} p(x \mid H1) dx, \tag{8}$$

provided that the probability of a false alarm (type I error):

$$\alpha = P(\textit{рішення } H1 | H0) = \int_{\Lambda(v)>\eta} p(x \mid H0) dx, \tag{9}$$

does not exceed a pre-set level (threshold) α0:

$$\alpha \le \alpha_0, \tag{10}$$

where α_0 – false alarm threshold.

Then the detection reliability (the probability of correctly detecting an abnormal condition) is defined as:

$$p_d = 1 - \beta. \tag{11}$$

Thus, the detection method controls the risk of false alarms (α) while minimizing the probability of missing a dangerous condition (β).

Proceedings of the 2nd International Scientific and Practical Conference
"Global Trends in Science: Research, Innovation and Development"
September 29 – October 1, 2025
Varna, Bulgaria
EUROPEAN OPEN SCIENCE SPACE

Since in practical conditions it is necessary to distinguish at least three network states, the problem is generalized to the multi-class case. The following classes are distinguished:

A_1: minimum load ($C(u) < \theta 1$) – suitable for preventive work

A_2: average load ($\theta 1 \le C(u) \le \theta 2$) – normal mode

A_3: critical load ($C(u) > \theta 2$) – a condition that carries the risk of overload

Interval classification rule:

$$A_1 : C(u) < \theta 1,\ A_2 : \theta 1 \le C(u) \le \theta 2,\ A_3 : C(u) > \theta 2. \quad (12)$$

Maximum likelihood classification for a multidimensional feature vector $C(u)$:

$$A(C(u)) = \arg\max_i p(C(u) | A_i). \quad (13)$$

Threshold values are calculated at the intersection points of the densities:

$$p(C(u) | A_1) = p(C(u) | C_2) \Rightarrow \theta_1, \quad (14)$$

$$p(C(u) | A_2) = p(C(u) | C_3) \Rightarrow \theta_2, \quad (15)$$

The combination of the likelihood ratio and the Neumann-Pearson criterion allows:

– ensure statistical optimality of decision-making;

– control the probability of misclassifications;

– generate automated decisions regarding load levels with high confidence.

As a result, the method not only detects critical modes, but also does so based on a rigorous mathematical model, taking into account the balance between first and second type errors.

The proposed method for automated detection of load levels is based on the PCVP model, centered signal values, estimation of correlation components according to the formula $\hat{B}_k(u)$, averaging over harmonic components, and application of the Neumann–Pearson criterion.

The method allows controlling the probability of type I errors, minimizing the omission of dangerous states, and applying both interval and probabilistic rules of multi-class classification.

References

1. Leland, W. E., Taqqu, M. S., Willinger, W., & Wilson, D. V. (1994). On the self-similar nature of Ethernet traffic (extended version). IEEE/ACM Transactions on Networking, 2(1), 1–15. DOI: 10.1109/90.282603.

2. Papagiannaki, K., Moon, S., Fraleigh, C., Thiran, P., Tobagi, F., & Diot, C. (2003). Analysis of measured single-hop delay from an operational backbone network. IEEE Journal on Selected Areas in Communications, 21(6), 908–921. DOI: 10.1109/JSAC.2003.814650.

3. Hsieh, H. Y., & Sivakumar, R. (2002). Performance comparison of cellular and multi-hop wireless networks: A quantitative study. ACM SIGMETRICS Performance Evaluation Review, 30(1), 113–122. DOI: 10.1145/511399.511388.

4. Li, J., Mohapatra, P., & Chuah, C.-N. (2005). Online anomaly detection in network traffic using wavelet analysis. IEEE INFOCOM 2005, 6, 2815–2825. DOI: 10.1109/INFCOM.2005.1498537.

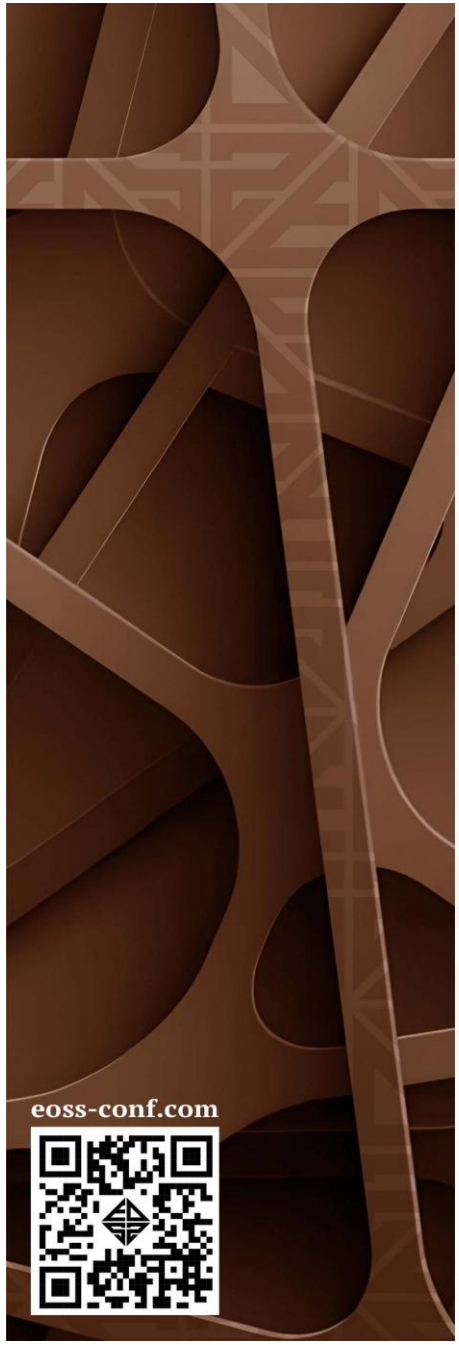
Proceedings of the 2nd International Scientific and Practical Conference
"Global Trends in Science: Research, Innovation and Development"
September 29 – October 1, 2025
Varna, Bulgaria

5. Abry, P., Veitch, D., & Flandrin, P. (1998). Long-range dependence: Revisiting aggregation with wavelets. Journal of Time Series Analysis, 19(3), 253-266. DOI: 10.1111/1467-9892.00090.
6. Khvostivska L., Khvostivskyi M., Dediv I., Yatskiv V., Palaniza Y. Method, Algorithm and Computer Tool for Synphase Detection of Radio Signals in Telecommunication Networks with Noises. Proceedings of the 1st International Workshop on Computer Information Technologies in Industry 4.0 (CITI 2023). CEUR Workshop Proceedings. Ternopil, Ukraine, June 14-16, 2023. P.173-180. ISSN 1613-0073.
7. Khvostivska L., Khvostivskyi M., Dediv I. Mathematical, algorithmic and software support for signals wavelet detection in electronic communications. Proceedings of the 2nd International Workshop on Computer Information Technologies in Industry 4.0 (CITI 2024). CEUR Workshop Proceedings. Ternopil, Ukraine, June 14-16, 2024. Vol. 3742. P.223-234. ISSN 1613-0073.
8. Neyman, J., & Pearson, E. S. (1933). On the problem of the most efficient tests of statistical hypotheses. Philosophical Transactions of the Royal Society of London. Series A, 231(694-706), 289–337. DOI: 10.1098/rsta.1933.0009.

МОДЕРНІЗАЦІЯ ТА ДОСЛІДЖЕННЯ ПОКАЗНИКІВ ЯКОСТІ ЕЛЕКТРОПРИВОДА МЕХАНІЗМУ РОТОРНОГО ВАГОНОПЕРЕКИДАЧА З ВИКОРИСТАННЯМ СУЧАСНИХ СИСТЕМ ЕЛЕКТРОПРИВОДУ

Нежурін В.І.
доцент, к.т.н.
Майстренко Д.Н.
магістрант
Салюк О.Д.
магістрант
Куваєв В.Ю.
старший викладач
Кафедра електричної інженерії
Український державний університет науки і технологій, Дніпро, Україна

На сучасному етапі складної економічної ситуації в металургійній промисловості української держави склалися сприятливі та необхідні обставини для модернізації діючих виробничих фондів основного технологічного обладнання металургійних підприємств. Вдосконалення напівпровідникової техніки, розвиток, теорії керування створює умови для розробки комплектних



CERTIFICATE
of participation

EUROPEAN OPEN SCIENCE SPACE

Yurii Herasymchuk

took part in the 2nd International Scientific and Practical Conference

«GLOBAL TRENDS IN SCIENCE: RESEARCH, INNOVATION AND DEVELOPMENT»

24 Hours of Participation
(0,8 ECTS credits)

Head of the
organizing committee
Helen Volokitina

EUROPEAN OPEN SCIENCE SPACE 3382508349

EOSS-25/0929-078 September 29 – October 1, 2025, Varna, Bulgaria

ДОДАТОК Б

Скрипт програмного забезпечення

```
%% ==========================================
%   СИНФАЗНИЙ МЕТОД + КЛАСИФІКАЦІЯ
%   На основі критерію Неймана–Пірсона
%   та багаторівневої класифікації станів
% ==========================================

clear all; close all; clc;

%% 1. Вихідні дані
data = load('traficdata.dat')';
dt = 0.05;
t = (0:(length(data)-1)) .* dt;

figure(1)
plot(t, data, 'b', 'LineWidth', 1.5);
set(gca, 'FontSize', 14);
xlabel('Час, год', 'FontSize', 14);
ylabel('Трафік, Тб', 'FontSize', 14);
title('Вхідний сигнал / дані трафіку', 'FontSize', 14);
grid on; axis tight;

%% 2. Синфазна обробка
B1 = PKVPsinfaz(data, 480, 'c');   % 480 – довжина ковзного вікна
Nshift = size(B1, 1);          % кількість відліків (колонок)
t_B1 = (0:Nshift-1) * dt;       % часові мітки (у годинах)

figure(2)
```

```
surf(t_B1, 1:size(B1,2), B1', 'EdgeColor', 'none');
shading interp;
xlabel('u, часове зміщення (год)', 'FontSize', 14);
ylabel('k, Номер компоненти', 'FontSize', 14);
zlabel('B_k(u), Тб^2', 'FontSize', 14);
title('Синфазна енергетична карта', 'FontSize', 14);
set(gca, 'FontSize', 14);
grid on; rotate3d on;

%% 3. Усереднення по гармонічних компонентах
m1 = mean(B1, 2);
m1 = m1(:);

figure(3)
plot(t(1:length(m1)), m1, 'LineWidth', 1.5);
set(gca, 'FontSize', 14);
xlabel('Час, год', 'FontSize', 14);
ylabel('M(u), Тб^2', 'FontSize', 14);
title('Усереднені синфазні компоненти', 'FontSize', 14);
grid on; axis tight;

%% 4. Емпіричний розподіл та апроксимації
nbins = 50;
[counts, centers] = hist(m1, nbins);
pdf_emp = counts / trapz(centers, counts);

mu0 = mean(m1) - 0.3*std(m1);
mu1 = mean(m1) + 0.3*std(m1);
sigma0 = std(m1);
sigma1 = std(m1);
```

```
x = linspace(min(m1), max(m1), 500);
p0 = normpdf(x, mu0, sigma0);
p1 = normpdf(x, mu1, sigma1);
% нормування щільностей
p0 = p0 / trapz(x, p0);
p1 = p1 / trapz(x, p1);

[f_kde, xi_kde] = ksdensity(m1, x);
overlap_area = trapz(x, min(p0, p1));

thr12 = mean(m1) - 0.5*std(m1);
thr23 = mean(m1) + 0.5*std(m1);

figure('Name','Емпіричний розподіл та
перевірки','NumberTitle','off','Units','normalized','Position',[0.1 0.1 0.85 0.6]);
subplot(3,1,1); hold on;
histogram(m1, nbins, 'Normalization', 'pdf', 'EdgeColor', [0.6 0.6 0.6], 'FaceColor', [0.9 0.9
0.9]);
plot(x, f_kde, 'k-', 'LineWidth', 1.6, 'DisplayName', 'KDE (емпірична)');
plot(x, p0, 'b--', 'LineWidth', 1.4, 'DisplayName', 'Нормальна H_0');
plot(x, p1, 'r--', 'LineWidth', 1.4, 'DisplayName', 'Нормальна H_1');
overlap_mask = min(p0,p1) > 0;
fill([x(overlap_mask), fliplr(x(overlap_mask))], ...
    [min(p0,p1).*(overlap_mask), zeros(1,sum(overlap_mask))], ...
    [0.8 0.8 0.2], 'FaceAlpha', 0.4, 'EdgeColor', 'none', 'DisplayName', 'Зона перекриття');
xlabel('Значення M(u), Тб^2', 'FontSize', 14);
ylabel('Щільність ймовірності', 'FontSize', 14);
title('Гістограма, KDE та гаусові апроксимації', 'FontSize', 14);
legend('Location','northeastoutside');
```

```
set(gca, 'FontSize', 14); grid on; hold off;
annotation('textbox',[0.05 0.05 0.4 0.05],'String',sprintf('Площа перекриття (p0,p1) =
%.4f', overlap_area), ...
    'FitBoxToText','on','FontSize',14,'EdgeColor','none');

subplot(3,1,2);
[f_ecdf, x_ecdf] = ecdf(m1);
plot(x_ecdf, f_ecdf, 'LineWidth', 1.6);
xlabel('Значення M(u), Тб^2', 'FontSize', 14);
ylabel('F(x)', 'FontSize', 14);
title('Емпірична CDF', 'FontSize', 14);
grid on; set(gca,'FontSize',14);

subplot(3,1,3);
qqplot(m1);
title('QQ-plot (перевірка нормальності)', 'FontSize', 14);
set(gca, 'FontSize', 14);

%% 5. Критерій Неймана–Пірсона з точним підбором порога для α0
L = p1 ./ (p0 + eps);

alpha0 = 0.05;  % бажаний рівень хибної тривоги
dx = mean(diff(x));

% сортуємо точки за спаданням L(x)
[Ls, idx] = sort(L, 'descend');
p0s = p0(idx);
p1s = p1(idx);
xs = x(idx);
```

```
% накопичуємо площу під p0
cum_p0 = cumsum(p0s * dx);
% шукаємо індекс, де площа досягає alpha0
k = find(cum_p0 >= alpha0, 1, 'first');
eta = Ls(k);          % поріг у просторі L(x)
thr = xs(k);          % відповідне значення у просторі x

% обчислення α, β, Pd
mask_reject = L > eta;       % зона прийняття H1
alpha = trapz(x(mask_reject), p0(mask_reject));
beta  = trapz(x(~mask_reject), p1(~mask_reject));
Pd    = 1 - beta;

sign_change = diff(sign(p1 - p0));
cross_idx = find(sign_change ~= 0);
x_cross = x(cross_idx);

figure(5); clf; hold on;
fill([x x(end) x(1)], [p0 0 0], [0.6 0.8 1], 'EdgeColor', 'none', 'FaceAlpha', 0.6);
fill([x x(end) x(1)], [p1 0 0], [1 0.6 0.6], 'EdgeColor', 'none', 'FaceAlpha', 0.6);
fill([x(mask_reject) fliplr(x(mask_reject))], [p0(mask_reject) zeros(1,sum(mask_reject))],
[0 0 1], 'FaceAlpha', 0.2);
fill([x(~mask_reject) fliplr(x(~mask_reject))], [p1(~mask_reject)
zeros(1,sum(~mask_reject))], [1 0 0], 'FaceAlpha', 0.2);
xline(thr, '--k', sprintf('Поріг (θ) = %.3f', thr), 'LineWidth', 1.5);
if ~isempty(x_cross)
    plot(x_cross, normpdf(x_cross, mu0, sigma0), 'ko', 'MarkerFaceColor', 'y', 'MarkerSize',
8);
end
```

```
text(mu0, max(p0)*0.8, 'H_0 (Нормальний стан)', 'Color', 'b', 'FontSize', 14,
'FontWeight', 'bold');
text(mu1, max(p1)*0.8, 'H_1 (Аномалія)', 'Color', 'r', 'FontSize', 14, 'FontWeight', 'bold');
xlabel('M(u), Тб^2', 'FontSize', 14);
ylabel('Щільність ймовірності', 'FontSize', 14);
title({'Критерій Неймана–Пірсона (точний α₀)', ...
    sprintf('\\alpha = %.3f,  \\beta = %.3f,  P_d = %.3f', alpha, beta, Pd)}, 'FontSize', 14);
legend('p_0(B_k)','p_1(B_k)','Помилка \alpha','Помилка \beta','Поріг','Location','best');
set(gca, 'FontSize', 14);
grid on; box on; hold off;

%% 6. Бінарна класифікація з урахуванням обраного η
L_test = interp1(x, L, m1, 'linear', 'extrap');
class_binary = double(L_test > eta);

figure(6)
plot(t(1:length(m1)), m1, 'k', 'LineWidth', 1.5); hold on;
plot(t(class_binary==1), m1(class_binary==1), 'ro', 'MarkerFaceColor', 'r');
yline(thr, '--b', sprintf('θ = %.3f', thr), 'LineWidth', 1.5);
yl = ylim;
text(t(round(length(t)*0.05)), yl(2) - 0.05*(yl(2)-yl(1)), sprintf('\\alpha = %.3f', alpha),
'Color', 'b', 'FontSize', 14, 'FontWeight', 'bold');
text(t(round(length(t)*0.05)), yl(2) - 0.10*(yl(2)-yl(1)), sprintf('\\beta = %.3f', beta),
'Color', 'r', 'FontSize', 14, 'FontWeight', 'bold');
text(t(round(length(t)*0.05)), yl(2) - 0.15*(yl(2)-yl(1)), sprintf('P_d = %.3f', Pd), 'Color',
'k', 'FontSize', 14, 'FontWeight', 'bold');
xlabel('Час, год', 'FontSize', 14);
ylabel('M(u), Тб^2', 'FontSize', 14);
title('Бінарна класифікація: нормальний стан / аномалія', 'FontSize', 14);
legend('Сигнал', 'Аномалія (H1)', 'Поріг','Location','best');
```

```
set(gca, 'FontSize', 14);
grid on; axis tight;

%% 7. Багатокласова класифікація (3 рівні)
thr12 = mean(m1) - 0.5 * std(m1);
thr23 = mean(m1) + 0.5 * std(m1);
class_multi = zeros(size(m1));
class_multi(m1 < thr12) = 1;
class_multi(m1 >= thr12 & m1 < thr23) = 2;
class_multi(m1 >= thr23) = 3;

figure(7)
scatter(t(1:length(m1)), m1, 25, class_multi, 'filled'); hold on;
yline(thr12, '--k', sprintf('θ1 = %.3f', thr12), 'LineWidth', 1.5);
yline(thr23, '--k', sprintf('θ2 = %.3f', thr23), 'LineWidth', 1.5);
xlabel('Час, год', 'FontSize', 14);
ylabel('M(u), Тб^2', 'FontSize', 14);
title('Багаторівнева класифікація рівнів навантаження', 'FontSize', 14);
colormap(jet(3));
cb = colorbar;
cb.Ticks = [1,2,3];
cb.TickLabels = {'Мінімальне','Середнє','Критичне'};
cb.FontSize = 14;
set(gca, 'FontSize', 14);
grid on; axis tight;

%% 8. Підсумки
disp('--- Класифікацію завершено ---');
fprintf('Порогове значення Δ(M(u)) = %.4f\n', eta);
fprintf('Відповідне значення порога θ = %.4f\n', thr);
```

```
fprintf('Ймовірність хибної тривоги α = %.4f\n', alpha);
fprintf('Ймовірність пропуску β = %.4f\n', beta);
fprintf('Достовірність виявлення Pd = %.4f\n', Pd);
fprintf('θ1 = %.4f, θ2 = %.4f\n', thr12, thr23);
if ~isempty(x_cross)
    fprintf('Точки перетину (межі аномалій): %.3f ', x_cross);
    fprintf('\n');
end
```

ДОДАТОК В

Скрипт програмного забезпечення графічного інтерфейсу системи

```
classdef program < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure     matlab.ui.Figure
        Menu_4       matlab.ui.container.Menu
        Menu_5       matlab.ui.container.Menu
        Menu_6       matlab.ui.container.Menu
        Menu_7       matlab.ui.container.Menu
        KDEMenu      matlab.ui.container.Menu
        CDFMenu      matlab.ui.container.Menu
        QQplotMenu   matlab.ui.container.Menu
        Menu_9       matlab.ui.container.Menu
        Menu_8       matlab.ui.container.Menu
        Menu_10      matlab.ui.container.Menu
        Menu_11      matlab.ui.container.Menu
        OKButton     matlab.ui.control.Button
        EditField    matlab.ui.control.NumericEditField
        Label        matlab.ui.control.Label
        UIAxes2      matlab.ui.control.UIAxes
        UIAxes       matlab.ui.control.UIAxes
        ContextMenu  matlab.ui.container.ContextMenu
    end


    properties (Access = private)
        trafficData % Description
```

```
    t
end


% Callbacks that handle component events
methods (Access = private)

    % Menu selected function: Menu_4
    function Menu_4Selected(app, event)
% Вибір файлу користувачем
[file, path] = uigetfile({'*.dat;*.txt', 'Data files (*.dat, *.txt)'}, ...
                         'Оберіть файл з даними трафіку');
if isequal(file, 0)
    uialert(app.UIFigure, 'Файл не вибрано', 'Помилка');
    return;
end

% Завантаження масиву
app.trafficData = load(fullfile(path, file));
dt = 1/(app.EditField.Value);
app.t = (0:(length(app.trafficData)-1)) .* dt;
% Відображення короткої інформації
uialert(app.UIFigure, 'Дані трафіку успішно завантажено!', 'OK');
if ~isempty(app.trafficData)
    plot(app.UIAxes, app.t, app.trafficData);
    title(app.UIAxes, 'Сигнал трафіку');
    xlabel(app.UIAxes, 'Час, год');
    ylabel(app.UIAxes, 'Трафік, Тб');
    grid(app.UIAxes, 'on');
    axis(app.UIAxes, 'tight');
```

```
    end
        end

        % Menu selected function: Menu_6
        function Menu_6Selected(app, event)
            %% 2. Синфазний аналіз
dt = 1/(app.EditField.Value);
B1 = PKVPsinfaz(app.trafficData', 480, 'c');   % 480 – довжина ковзного вікна
Nshift = size(B1, 1);          % кількість відліків (колонок)
t_B1 = (0:Nshift-1) * dt;      % часові мітки (у годинах)

surf(app.UIAxes2, t_B1, 1:size(B1,2), B1', 'EdgeColor', 'none');
shading(app.UIAxes2, 'interp');
xlabel(app.UIAxes2, 'u, часове зміщення (год)', FontSize=14);
ylabel(app.UIAxes2, 'k, Номер компоненти', FontSize=14);
zlabel(app.UIAxes2, 'B_k(u), Тб^2', FontSize=14);
title(app.UIAxes2, 'Синфазна енергетична карта', FontSize=14);
grid (app.UIAxes2, 'on');
rotate3d (app.UIAxes2, 'on');
        end

        % Menu selected function: Menu_7
        function Menu_7Selected(app, event)
%% 3. Усереднення по гармонічних компонентах
B1 = PKVPsinfaz(app.trafficData', 480, 'c');   % 480 – довжина ковзного вікна
m1 = mean(B1, 2);
m1 = m1(:);

plot(app.UIAxes2, app.t(1:length(m1)), m1, LineWidth=1.5);
xlabel(app.UIAxes2, 'Час, год', FontSize=14);
```

```
ylabel(app.UIAxes2, 'M(u), Тб^2', FontSize=14);
title(app.UIAxes2, 'Усереднені синфазні компоненти', FontSize=14);
grid (app.UIAxes2, 'on');
axis (app.UIAxes2, 'on');
    end

    % Menu selected function: KDEMenu
    function KDEMenuSelected(app, event)
%% 4. Емпіричний розподіл та апроксимації (на app.UIAxes2)

B1 = PKVPsinfaz(app.trafficData', 480, 'c');   % 480 – довжина ковзного вікна
m1 = mean(B1, 2);
m1 = m1(:);

% === Параметри гістограми та розподілів ===
nbins = 50;
[counts, centers] = hist(m1, nbins);
pdf_emp = counts / trapz(centers, counts);

mu0 = mean(m1) - 0.3*std(m1);
mu1 = mean(m1) + 0.3*std(m1);
sigma0 = std(m1);
sigma1 = std(m1);

x = linspace(min(m1), max(m1), 500);
p0 = normpdf(x, mu0, sigma0);
p1 = normpdf(x, mu1, sigma1);

% нормування щільностей
p0 = p0 / trapz(x, p0);
```

```
p1 = p1 / trapz(x, p1);

% KDE оцінка
[f_kde, xi_kde] = ksdensity(m1, x);

% Площа перекриття
overlap_area = trapz(x, min(p0, p1));

thr12 = mean(m1) - 0.5*std(m1);
thr23 = mean(m1) + 0.5*std(m1);

% === Побудова графіка без subplot ===
cla(app.UIAxes2, 'reset'); % очищення осей перед побудовою
hold(app.UIAxes2, 'on');

% Емпірична гістограма
histogram(app.UIAxes2, m1, nbins, 'Normalization', 'pdf', ...
    'EdgeColor', [0.6 0.6 0.6], 'FaceColor', [0.9 0.9 0.9], ...
    'DisplayName', 'Емпіричний розподіл');

% KDE і нормальні апроксимації
plot(app.UIAxes2, x, f_kde, 'k-', 'LineWidth', 1.6, 'DisplayName', 'KDE (емпірична)');
plot(app.UIAxes2, x, p0, 'b--', 'LineWidth', 1.4, 'DisplayName', 'Нормальна H_0');
plot(app.UIAxes2, x, p1, 'r--', 'LineWidth', 1.4, 'DisplayName', 'Нормальна H_1');

% Зона перекриття
overlap_mask = min(p0,p1) > 0;
fill(app.UIAxes2, [x(overlap_mask), fliplr(x(overlap_mask))], ...
    [min(p0,p1).*(overlap_mask), zeros(1,sum(overlap_mask))], ...
    [0.8 0.8 0.2], 'FaceAlpha', 0.4, 'EdgeColor', 'none', ...
```

```
    'DisplayName', 'Зона перекриття');

% --- Оформлення графіка ---
xlabel(app.UIAxes2, 'Значення M(u), Тб^2', 'FontSize', 14);
ylabel(app.UIAxes2, 'Щільність ймовірності', 'FontSize', 14);
title(app.UIAxes2, 'Гістограма, KDE та гаусові апроксимації', 'FontSize', 14);
legend(app.UIAxes2, 'Location', 'northeastoutside');
grid(app.UIAxes2, 'on');
set(app.UIAxes2, 'FontSize', 14);
hold(app.UIAxes2, 'off');

% === Вивід числового результату ===
msg = sprintf('Площа перекриття (p0, p1) = %.4f', overlap_area);
uialert(app.UIFigure, msg, 'Результат аналізу');
    end

    % Menu selected function: CDFMenu
    function CDFMenuSelected(app, event)
        %% 4. Емпіричний розподіл та апроксимації (на app.UIAxes2)

B1 = PKVPsinfaz(app.trafficData', 480, 'c');   % 480 – довжина ковзного вікна
m1 = mean(B1, 2);
m1 = m1(:);

% === Параметри гістограми та розподілів ===
nbins = 50;
[counts, centers] = hist(m1, nbins);
pdf_emp = counts / trapz(centers, counts);

mu0 = mean(m1) - 0.3*std(m1);
```

```
mu1 = mean(m1) + 0.3*std(m1);
sigma0 = std(m1);
sigma1 = std(m1);

x = linspace(min(m1), max(m1), 500);
p0 = normpdf(x, mu0, sigma0);
p1 = normpdf(x, mu1, sigma1);

% нормування щільностей
p0 = p0 / trapz(x, p0);
p1 = p1 / trapz(x, p1);

% KDE оцінка
[f_kde, xi_kde] = ksdensity(m1, x);

% Площа перекриття
overlap_area = trapz(x, min(p0, p1));

thr12 = mean(m1) - 0.5*std(m1);
thr23 = mean(m1) + 0.5*std(m1);

%% Емпірична CDF (на app.UIAxes2)
[f_ecdf, x_ecdf] = ecdf(m1);

cla(app.UIAxes2, 'reset'); % очистити осі перед побудовою
hold(app.UIAxes2, 'on');

plot(app.UIAxes2, x_ecdf, f_ecdf, 'LineWidth', 1.6, 'Color', 'b', ...
    'DisplayName', 'Емпірична CDF');
```

```
xlabel(app.UIAxes2, 'Значення M(u), Тб^2', 'FontSize', 14);
ylabel(app.UIAxes2, 'F(x)', 'FontSize', 14);
title(app.UIAxes2, 'Емпірична CDF', 'FontSize', 14);

grid(app.UIAxes2, 'on');
set(app.UIAxes2, 'FontSize', 14);
legend(app.UIAxes2, 'Location', 'southeast');

hold(app.UIAxes2, 'off');

    end

    % Menu selected function: QQplotMenu
    function QQplotMenuSelected(app, event)
B1 = PKVPsinfaz(app.trafficData', 480, 'c');   % 480 – довжина ковзного вікна
m1 = mean(B1, 2);
m1 = m1(:);

% === Параметри гістограми та розподілів ===
nbins = 50;
[counts, centers] = hist(m1, nbins);
pdf_emp = counts / trapz(centers, counts);

mu0 = mean(m1) - 0.3*std(m1);
mu1 = mean(m1) + 0.3*std(m1);
sigma0 = std(m1);
sigma1 = std(m1);

x = linspace(min(m1), max(m1), 500);
p0 = normpdf(x, mu0, sigma0);
```

```
p1 = normpdf(x, mu1, sigma1);

% нормування щільностей
p0 = p0 / trapz(x, p0);
p1 = p1 / trapz(x, p1);

% KDE оцінка
[f_kde, xi_kde] = ksdensity(m1, x);

% Площа перекриття
overlap_area = trapz(x, min(p0, p1));

thr12 = mean(m1) - 0.5*std(m1);
thr23 = mean(m1) + 0.5*std(m1);

%% QQ-plot (перевірка нормальності) у app.UIAxes2
cla(app.UIAxes2, 'reset');              % очистити осі перед побудовою
qqplot(app.UIAxes2, m1);                % побудова QQ-графіка
title(app.UIAxes2,'QQ-plot (перевірка нормальності)', FontSize=14);
grid(app.UIAxes2, 'on');
axis (app.UIAxes2, 'on');
    end

    % Menu selected function: Menu_9
    function Menu_9Selected(app, event)
%% 5. Критерій Неймана–Пірсона з точним підбором порога для α0
B1 = PKVPsinfaz(app.trafficData', 480, 'c');   % 480 – довжина ковзного вікна
m1 = mean(B1, 2);
m1 = m1(:);
```

```
% === Параметри гістограми та розподілів ===
nbins = 50;
[counts, centers] = hist(m1, nbins);
pdf_emp = counts / trapz(centers, counts);

mu0 = mean(m1) - 0.3*std(m1);
mu1 = mean(m1) + 0.3*std(m1);
sigma0 = std(m1);
sigma1 = std(m1);

x = linspace(min(m1), max(m1), 500);
p0 = normpdf(x, mu0, sigma0);
p1 = normpdf(x, mu1, sigma1);

% нормування щільностей
p0 = p0 / trapz(x, p0);
p1 = p1 / trapz(x, p1);

% KDE оцінка
[f_kde, xi_kde] = ksdensity(m1, x);

% Площа перекриття
overlap_area = trapz(x, min(p0, p1));

thr12 = mean(m1) - 0.5*std(m1);
thr23 = mean(m1) + 0.5*std(m1);

L = p1 ./ (p0 + eps);

alpha0 = 0.05;  % бажаний рівень хибної тривоги
```

```
dx = mean(diff(x));

% сортуємо точки за спаданням L(x)
[Ls, idx] = sort(L, 'descend');
p0s = p0(idx);
p1s = p1(idx);
xs = x(idx);

% накопичуємо площу під p0
cum_p0 = cumsum(p0s * dx);
% шукаємо індекс, де площа досягає alpha0
k = find(cum_p0 >= alpha0, 1, 'first');
eta = Ls(k);          % поріг у просторі L(x)
thr = xs(k);          % відповідне значення у просторі x

% обчислення α, β, Pd
mask_reject = L > eta;       % зона прийняття H1
alpha = trapz(x(mask_reject), p0(mask_reject));
beta  = trapz(x(~mask_reject), p1(~mask_reject));
Pd    = 1 - beta;

sign_change = diff(sign(p1 - p0));
cross_idx = find(sign_change ~= 0);
x_cross = x(cross_idx);

%% ВІЗУАЛІЗАЦІЯ НА APP.UIAxes2
cla(app.UIAxes2, 'reset'); % очистити осі
hold(app.UIAxes2, 'on');

% Області розподілів p0 та p1
```

```
fill(app.UIAxes2, [x x(end) x(1)], [p0 0 0], [0.6 0.8 1], ...
    'EdgeColor', 'none', 'FaceAlpha', 0.6);
fill(app.UIAxes2, [x x(end) x(1)], [p1 0 0], [1 0.6 0.6], ...
    'EdgeColor', 'none', 'FaceAlpha', 0.6);

% Області помилок α та β
fill(app.UIAxes2, [x(mask_reject) fliplr(x(mask_reject))], ...
    [p0(mask_reject) zeros(1,sum(mask_reject))], [0 0 1], 'FaceAlpha', 0.2);
fill(app.UIAxes2, [x(~mask_reject) fliplr(x(~mask_reject))], ...
    [p1(~mask_reject) zeros(1,sum(~mask_reject))], [1 0 0], 'FaceAlpha', 0.2);

% Лінія порога
xline(app.UIAxes2, thr, '--k', sprintf('Поріг (θ) = %.3f', thr), 'LineWidth', 1.5);

% Точки перетину
if ~isempty(x_cross)
   plot(app.UIAxes2, x_cross, normpdf(x_cross, mu0, sigma0), ...
      'ko', 'MarkerFaceColor', 'y', 'MarkerSize', 8);
end

% Пояснюючі написи
text(app.UIAxes2, mu0, max(p0)*0.8, 'H_0 (Нормальний стан)', ...
   'Color', 'b', 'FontSize', 14, 'FontWeight', 'bold');
text(app.UIAxes2, mu1, max(p1)*0.8, 'H_1 (Аномалія)', ...
   'Color', 'r', 'FontSize', 14, 'FontWeight', 'bold');

% Підписи осей, легенда, заголовок
xlabel(app.UIAxes2, 'M(u), Тб^2', 'FontSize', 14);
ylabel(app.UIAxes2, 'Щільність ймовірності', 'FontSize', 14);
title(app.UIAxes2, {'Критерій Неймана–Пірсона (точний α₀)', ...
```

```
    sprintf('\\alpha = %.3f,  \\beta = %.3f,  P_d = %.3f', alpha, beta, Pd)}, ...
    'FontSize', 14);
legend(app.UIAxes2, 'p_0(B_k)', 'p_1(B_k)', 'Помилка \alpha', ...
       'Помилка \beta', 'Поріг', 'Location', 'best');

% Декорації
set(app.UIAxes2, 'FontSize', 14);
grid(app.UIAxes2, 'on');
box(app.UIAxes2, 'on');
hold(app.UIAxes2, 'off');

        end

        % Menu selected function: Menu_10
        function Menu_10Selected(app, event)
%% 6. Бінарна класифікація з урахуванням обраного η
B1 = PKVPsinfaz(app.trafficData', 480, 'c');   % 480 – довжина ковзного вікна
m1 = mean(B1, 2);
m1 = m1(:);

% === Параметри гістограми та розподілів ===
nbins = 50;
[counts, centers] = hist(m1, nbins);
pdf_emp = counts / trapz(centers, counts);

mu0 = mean(m1) - 0.3*std(m1);
mu1 = mean(m1) + 0.3*std(m1);
sigma0 = std(m1);
sigma1 = std(m1);
```

```
x = linspace(min(m1), max(m1), 500);
p0 = normpdf(x, mu0, sigma0);
p1 = normpdf(x, mu1, sigma1);

% нормування щільностей
p0 = p0 / trapz(x, p0);
p1 = p1 / trapz(x, p1);

% KDE оцінка
[f_kde, xi_kde] = ksdensity(m1, x);

% Площа перекриття
overlap_area = trapz(x, min(p0, p1));

thr12 = mean(m1) - 0.5*std(m1);
thr23 = mean(m1) + 0.5*std(m1);

L = p1 ./ (p0 + eps);

alpha0 = 0.05;  % бажаний рівень хибної тривоги
dx = mean(diff(x));

% сортуємо точки за спаданням L(x)
[Ls, idx] = sort(L, 'descend');
p0s = p0(idx);
p1s = p1(idx);
xs = x(idx);

% накопичуємо площу під p0
cum_p0 = cumsum(p0s * dx);
```

```
% шукаємо індекс, де площа досягає alpha0
k = find(cum_p0 >= alpha0, 1, 'first');
eta = Ls(k);          % поріг у просторі L(x)
thr = xs(k);          % відповідне значення у просторі x

% обчислення α, β, Pd
mask_reject = L > eta;       % зона прийняття H1
alpha = trapz(x(mask_reject), p0(mask_reject));
beta  = trapz(x(~mask_reject), p1(~mask_reject));
Pd    = 1 - beta;

sign_change = diff(sign(p1 - p0));
cross_idx = find(sign_change ~= 0);
x_cross = x(cross_idx);

L_test = interp1(x, L, m1, 'linear', 'extrap');
class_binary = double(L_test > eta);

% Очистити осі перед новим графіком
cla(app.UIAxes2, 'reset');
hold(app.UIAxes2, 'on');

% Основний сигнал
plot(app.UIAxes2, app.t(1:length(m1)), m1, 'k', 'LineWidth', 1.5);

% Позначення точок, де виявлено аномалії
plot(app.UIAxes2, app.t(class_binary==1), m1(class_binary==1), ...
    'ro', 'MarkerFaceColor', 'r');

% Лінія порога
```

```
yline(app.UIAxes2, thr, '--b', sprintf('θ = %.3f', thr), 'LineWidth', 1.5);

% Отримати межі по Y
yl = ylim(app.UIAxes2);

% Текстова інформація про показники α, β, Pd
text(app.UIAxes2, app.t(round(length(app.t)*0.05)), ...
    yl(2) - 0.05*(yl(2)-yl(1)), ...
    sprintf('\\alpha = %.3f', alpha), ...
    'Color', 'b', 'FontSize', 14, 'FontWeight', 'bold');

text(app.UIAxes2, app.t(round(length(app.t)*0.05)), ...
    yl(2) - 0.10*(yl(2)-yl(1)), ...
    sprintf('\\beta = %.3f', beta), ...
    'Color', 'r', 'FontSize', 14, 'FontWeight', 'bold');

text(app.UIAxes2, app.t(round(length(app.t)*0.05)), ...
    yl(2) - 0.15*(yl(2)-yl(1)), ...
    sprintf('P_d = %.3f', Pd), ...
    'Color', 'k', 'FontSize', 14, 'FontWeight', 'bold');

% Підписи, легенда, сітка
xlabel(app.UIAxes2, 'Час, год', 'FontSize', 14);
ylabel(app.UIAxes2, 'M(u), Тб^2', 'FontSize', 14);
title(app.UIAxes2, 'Бінарна класифікація: нормальний стан / аномалія', 'FontSize', 14);
legend(app.UIAxes2, 'Сигнал', 'Аномалія (H1)', 'Поріг', 'Location', 'best');

set(app.UIAxes2, 'FontSize', 14);
grid(app.UIAxes2, 'on');
axis(app.UIAxes2, 'tight');
```

```
hold(app.UIAxes2, 'off');
    end

    % Menu selected function: Menu_11
    function Menu_11Selected(app, event)
%% 7. Багатокласова класифікація (3 рівні)
B1 = PKVPsinfaz(app.trafficData', 480, 'c');   % 480 – довжина ковзного вікна
m1 = mean(B1, 2);
m1 = m1(:);
%% 7. Багаторівнева класифікація рівнів навантаження
thr12 = mean(m1) - 0.5 * std(m1);
thr23 = mean(m1) + 0.5 * std(m1);

class_multi = zeros(size(m1));
class_multi(m1 < thr12) = 1;
class_multi(m1 >= thr12 & m1 < thr23) = 2;
class_multi(m1 >= thr23) = 3;

% Очистити осі перед новим графіком
cla(app.UIAxes2, 'reset');
hold(app.UIAxes2, 'on');

% Побудова точок із кольоровим маркуванням
scatter(app.UIAxes2, app.t(1:length(m1)), m1, 25, class_multi, 'filled');

% Лінії порогів
yline(app.UIAxes2, thr12, '--k', sprintf('θ₁ = %.3f', thr12), 'LineWidth', 1.5);
yline(app.UIAxes2, thr23, '--k', sprintf('θ₂ = %.3f', thr23), 'LineWidth', 1.5);

% Підписи, сітка, форматування
```

```
xlabel(app.UIAxes2, 'Час, год', 'FontSize', 14);
ylabel(app.UIAxes2, 'M(u), Тб^2', 'FontSize', 14);
title(app.UIAxes2, 'Багаторівнева класифікація рівнів навантаження', 'FontSize', 14);

% Колірна мапа та легенда (через colorbar)
colormap(app.UIAxes2, jet(3));
cb = colorbar(app.UIAxes2);
cb.Ticks = [1, 2, 3];
cb.TickLabels = {'Мін', 'Сер', 'Крит'};
cb.FontSize = 14;

set(app.UIAxes2, 'FontSize', 14);
grid(app.UIAxes2, 'on');
axis(app.UIAxes2, 'tight');
hold(app.UIAxes2, 'off');

    end

    % Button pushed function: OKButton
    function OKButtonPushed(app, event)
%  Натиснена кнопка
dt = 1/(app.EditField.Value);
app.t = (0:(length(app.trafficData)-1)) .* dt;
plot(app.UIAxes, app.t, app.trafficData);
title(app.UIAxes, 'Сигнал трафіку');
xlabel(app.UIAxes, 'Час, год');
ylabel(app.UIAxes, 'Трафік, Тб');
grid(app.UIAxes, 'on');
axis(app.UIAxes, 'tight');
    end
```

```
    end

    % Component initialization
    methods (Access = private)

        % Create UIFigure and components
        function createComponents(app)

            % Create UIFigure and hide until all components are created
            app.UIFigure = uifigure('Visible', 'off');
            app.UIFigure.Position = [100 100 1216 627];
            app.UIFigure.Name = 'MATLAB App';

            % Create Menu_4
            app.Menu_4 = uimenu(app.UIFigure);
            app.Menu_4.MenuSelectedFcn = createCallbackFcn(app, @Menu_4Selected,
true);
            app.Menu_4.Text = 'Завантажити дані трафіку';

            % Create Menu_5
            app.Menu_5 = uimenu(app.UIFigure);
            app.Menu_5.Text = 'Обробка трафіку';

            % Create Menu_6
            app.Menu_6 = uimenu(app.Menu_5);
            app.Menu_6.MenuSelectedFcn = createCallbackFcn(app, @Menu_6Selected,
true);
            app.Menu_6.Text = 'Снфазна енергетична карта';

            % Create Menu_7
```

```
            app.Menu_7 = uimenu(app.Menu_5);
            app.Menu_7.MenuSelectedFcn = createCallbackFcn(app, @Menu_7Selected,
true);
            app.Menu_7.Text = 'Усереднені синфазні компоненти ';

            % Create KDEMenu
            app.KDEMenu = uimenu(app.Menu_5);
            app.KDEMenu.MenuSelectedFcn = createCallbackFcn(app, @KDEMenuSelected,
true);
            app.KDEMenu.Text = 'Гістограма, KDE та гаусові апроксимації';

            % Create CDFMenu
            app.CDFMenu = uimenu(app.Menu_5);
            app.CDFMenu.MenuSelectedFcn = createCallbackFcn(app, @CDFMenuSelected,
true);
            app.CDFMenu.Text = 'Емпірична CDF';

            % Create QQplotMenu
            app.QQplotMenu = uimenu(app.Menu_5);
            app.QQplotMenu.MenuSelectedFcn = createCallbackFcn(app,
@QQplotMenuSelected, true);
            app.QQplotMenu.Text = 'QQ-plot (перевірка нормальності)';

            % Create Menu_9
            app.Menu_9 = uimenu(app.Menu_5);
            app.Menu_9.MenuSelectedFcn = createCallbackFcn(app, @Menu_9Selected,
true);
            app.Menu_9.Text = 'Криві щільностей розподілу ймовірностей та виділення
областей станів';
```

```
        % Create Menu_8
        app.Menu_8 = uimenu(app.UIFigure);
        app.Menu_8.Text = 'Класифікація';

        % Create Menu_10
        app.Menu_10 = uimenu(app.Menu_8);
        app.Menu_10.MenuSelectedFcn = createCallbackFcn(app, @Menu_10Selected,
true);
        app.Menu_10.Text = 'Бінарна класифікація трафіку мережі';

        % Create Menu_11
        app.Menu_11 = uimenu(app.Menu_8);
        app.Menu_11.MenuSelectedFcn = createCallbackFcn(app, @Menu_11Selected,
true);
        app.Menu_11.Text = 'Багаторівнена класифікація рівнів навантаження трафіку
мережі';

        % Create UIAxes
        app.UIAxes = uiaxes(app.UIFigure);
        title(app.UIAxes, 'Реалізація трафіку комп"ютерної мережі')
        xlabel(app.UIAxes, 'Час, год')
        ylabel(app.UIAxes, 'Трафік, Тб')
        zlabel(app.UIAxes, 'Z')
        app.UIAxes.FontSize = 14;
        app.UIAxes.Position = [26 390 1164 209];

        % Create UIAxes2
        app.UIAxes2 = uiaxes(app.UIFigure);
        title(app.UIAxes2, 'Результат')
        xlabel(app.UIAxes2, 'X')
```

```
        ylabel(app.UIAxes2, 'Y')
        zlabel(app.UIAxes2, 'Z')
        app.UIAxes2.FontSize = 14;
        app.UIAxes2.Position = [26 12 1164 366];

        % Create Label
        app.Label = uilabel(app.UIFigure);
        app.Label.HorizontalAlignment = 'right';
        app.Label.FontSize = 14;
        app.Label.Position = [883 606 172 22];
        app.Label.Text = 'Частота дискретизації, Гц';

        % Create EditField
        app.EditField = uieditfield(app.UIFigure, 'numeric');
        app.EditField.FontSize = 14;
        app.EditField.Position = [1070 606 59 22];
        app.EditField.Value = 500;

        % Create OKButton
        app.OKButton = uibutton(app.UIFigure, 'push');
        app.OKButton.ButtonPushedFcn = createCallbackFcn(app, @OKButtonPushed,
true);
        app.OKButton.BackgroundColor = [0.8 0.8 0.8];
        app.OKButton.FontSize = 14;
        app.OKButton.Position = [1139 603 35 26];
        app.OKButton.Text = 'OK';

        % Create ContextMenu
        app.ContextMenu = uicontextmenu(app.UIFigure);
```

```
        % Assign app.ContextMenu
        app.UIFigure.ContextMenu = app.ContextMenu;

        % Show the figure after all components are created
        app.UIFigure.Visible = 'on';
    end
end

% App creation and deletion
methods (Access = public)

    % Construct app
    function app = program

        % Create UIFigure and components
        createComponents(app)

        % Register the app with App Designer
        registerApp(app, app.UIFigure)

        if nargout == 0
            clear app
        end
    end

    % Code that executes before app deletion
    function delete(app)

        % Delete UIFigure when app is deleted
        delete(app.UIFigure)
```

```
        end
    end
end
```