КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ РАДІОФІЗИКИ, ЕЛЕКТРОНІКИ ТА КОМП'ЮТЕРНИХ

СИСТЕМ Кафедра радіотехніки та радіоелектронних систем

«На правах рукопису»

Робота допущена до захисту в ЕК
рішенням кафедри радіотехніки та радіоелектронних систем
від _____ 2025 року, протокол № _____.
Завідувач кафедри доктор фіз.-мат. наук, професор
_____Ігор АНІСІМОВ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему:

«АЛГОРИТМИ РОЗПІЗНАВАННЯ АНОМАЛІЙ У СИГНАЛАХ З ВИКОРИСТАННЯМ PYTHON ТА НЕЙРОМЕРЕЖ»

Виконав: студент 4-го курсу
денної форми навчання
спеціальності 172 - Електронні комунікації та радіотехніка
ОПП «Інформаційна безпека телекомунікаційних систем і мереж»
Солнцев Арсеній Богданович

Науковий керівник:
кандидат технічних наук, с.н.с.
Жиров Генадій Борисович _____

Рецензент:
Начальник науково-дослідного відділу
науково дослідного центру ВІКНУ
кандидат технічних наук, с.н.с.
Охрамович Михайло Миколайович _____

Засвідчую, що у цій бакалаврській роботі
немає запозичень з праць інших авторів без
відповідних посилань
Студент _____ Солнцев Арсеній

Київ – 2025

РЕФЕРАТ

Дипломна робота: 30 с., 17 рис., 11 джерел.

Об’єкт розроблення – модель нейромережі для виявлення аномалій у сигналі.

Мета роботи – розробка нескладної моделі нейромережі для виявлення аномалій у сигналі.

У роботі побудовано повний програмний цикл обробки сигналу, що включає попередню обробку FM-сигналу, обчислення миттєвої частоти, віконування сигналу, формування міток, навчання згорткового автоенкодера та реконструкцію очищеного сигналу. Також увагу приділено застосуванню швидкого перетворення Фур’є (FFT) для аналізу спектра сигналів до та після обробки.

Для забезпечення точного виявлення відхилень використано віконування по миттєвій частоті з фіксованим кроком та віконним розміром, що дало змогу формувати навчальні дані з високою часовою точністю. Мітки будувалися на основі положення аномалії в сигналі.

Навчена нейронна мережа дозволяє ефективно розпізнавати відхилення у частотному просторі та реконструювати "здоровий" сигнал, що підтверджено спектральним аналізом та візуалізацією результатів. Виявлено значне зниження амплітудності сплесків у спектрі після відновлення сигналу, що свідчить про успішне приглушення аномалій.

Розроблений підхід потенційно може застосовуватись у системах контролю якості радіосигналів, моніторингу технічного стану систем зв’язку або в задачах попередньої обробки даних перед класифікацією.

У подальшій роботі доцільно покращити модель, дослідити інші типи архітектур глибинного навчання для виявлення складніших типів аномалій, зокрема у багатоканальних сигналах чи реальному часі.

ЗМІСТ

Вступ……..…... 4
1. Теоретичні основи реконструкції та виявлення аномалій у радіосигналах…….5
1.1. Представлення сигналу через миттєву частоту…… 5
1.2. Формування вікон……..……..7
1.3. Нормалізація та подання даних……. 8
1.4. Архітектура згорткового автоенкодера……..... 8
1.5. Виявлення аномалій за помилкою реконструкції…… 9
1.6. Метрики оцінки……... 9
1.7. Відновлення “чистого” сигналу…….....10
2. Розробка моделі нейронної мережі……12
2.1. Побудова робочої схеми для генерування сигналу……..12
2.2. Завантаження та обробка сигналу…… 13
2.3. Підготовка даних для навчання нейромережі…… 14
2.4. Нормалізація даних……... 15
2.5. Побудова автоенкодера……. 16
2.6. Навчання моделі…….17
2.7. Оцінка моделі……..... 17
2.8. Аналіз отриманих результатів……. 19
2.9. Реконструкція сигналу…….. 24
2.10. Перетворення Фур’є……..……...25
Висновки……..…….…….....29
Перелік джерел посилання ……30

ВСТУП

У сучасному світі, де цифрові технології проникають у всі сфери життя — від побутових пристроїв до складних телекомунікаційних систем — забезпечення надійності та якості передачі сигналів набуває критичного значення. Однією з ключових загроз ефективному функціонуванню таких систем є аномалії в сигналі, які можуть бути спричинені як зовнішніми перешкодами, так і внутрішніми збоями апаратури. Їх вчасне виявлення є запорукою стабільної роботи систем зв’язку, радіолокації, медичної діагностики, а також об’єктів критичної інфраструктури.

Традиційні методи аналізу сигналів, засновані на класичних фільтраційних або спектральних підходах, мають обмеження щодо ефективності при роботі зі складними, динамічно змінюваними або слабко структурованими сигналами. У цьому контексті на перший план виходять методи машинного навчання, зокрема глибокі нейронні мережі, які здатні адаптивно виявляти приховані закономірності та відхилення в сигнальному потоці.

Метою цієї дипломної роботи є дослідження можливостей автоматичного розпізнавання аномалій у радіосигналах за допомогою нейронних мереж, зокрема автоенкодерів на основі згорткових структур. Особливу увагу приділено підходам до попередньої обробки даних, формуванню навчальних вікон сигналу, а також аналізу якості реконструкції сигналу після відфільтровування шумових компонент.

1. ТЕОРЕТИЧНІ ОСНОВИ РЕКОНСТРУКЦІЇ ТА ВИЯВЛЕННЯ АНОМАЛІЙ У РАДІОСИГНАЛАХ

У сучасних телекомунікаційних системах обробка сигналів відіграє ключову роль для забезпечення стабільного, безпечного та ефективного передавання інформації. Однією з серйозних проблем є виникнення аномалій у сигналі, які можуть бути спричинені зовнішніми завадами, апаратними збоями або втручанням у канал зв’язку. Такі спотворення не лише ускладнюють подальшу інтерпретацію сигналу, а й можуть свідчити про потенційні загрози — як технічного, так і безпекового характеру.

З розвитком штучного інтелекту та зростанням обчислювальних потужностей виникла можливість застосування нейронних мереж для автоматизованого виявлення таких аномалій. На відміну від традиційних методів, які здебільшого базуються на пороговій обробці або статистичних підходах, нейромережеві рішення дозволяють гнучко адаптуватися до складної структури сигналу, виявляти приховані закономірності та проводити не лише класифікацію, але й реконструкцію сигналу без спотворень.

У межах цієї роботи здійснено розробку алгоритму розпізнавання аномалій у радіосигналі на основі аналізу миттєвої частоти з використанням згорткових автоенкодерів. Такий підхід дозволяє не лише виявляти відхилення, а й формувати очищену версію сигналу, що є критично важливим для систем, де достовірність прийнятої інформації визначає подальшу ефективність функціонування.

Таким чином, розробка інтелектуальної системи для виявлення та обробки аномалій у сигналі є надзвичайно актуальною задачею, яка має як наукову, так і практичну значущість для сучасної радіоелектроніки, систем зв’язку та інформаційної безпеки.

1.1 Представлення сигналу через миттєву частоту

У задачах цифрової обробки радіосигналів особливе значення має правильне представлення даних, що передають фізичну суть процесів у каналі зв’язку. Радіосигнали, зокрема ті, що використовують модуляцію (наприклад,

частотну), не є простою послідовністю дійсних чисел — вони мають як амплітудну, так і фазову компоненти, що змінюються у часі. Для повного представлення такої інформації використовується комплексна форма сигналу, де реальна частина зазвичай відображає одну з ортогональних складових (наприклад, I – in-phase), а уявна – іншу (Q – quadrature).

З технічного погляду, це дозволяє працювати з аналітичним представленням сигналу, уникнувши втрат фазової інформації, яка є критично важливою для подальшого аналізу частотного спектра, обчислення миттєвої частоти або демодуляції. Саме комплексна форма сигналу забезпечує можливість точного застосування таких методів, як перетворення Гільберта або швидке перетворення Фур'є.

Вибір формату complex64 (тобто 64-бітне подання комплексного числа: по 32 біти на дійсну та уявну частини) обумовлений балансом між точністю представлення даних і ефективністю обчислень. Такий формат широко підтримується сучасними бібліотеками для чисельного аналізу (зокрема NumPy та PyTorch) і дозволяє працювати з великими масивами сигналів у реальному часі без зайвого навантаження на пам'ять чи процесор. Таким чином, збереження сигналу у вигляді complex64 є не лише зручним, а й технічно виправданим рішенням, що забезпечує високу точність подальшої обробки та розпізнавання аномалій за допомогою нейронної мережі.

Початкові дані являють собою складний сигнал у форматі complex64, записаний у бінарному вигляді. Щоб отримати фізично осмислену ознаку, проводиться перехід від сигналу до його миттєвої фази, за допомогою функції np.unwrap():

$$\varphi[n] = unwrap(arg(x[n])) \tag{1.1}$$

$x[n]$ — комплексні відліки сигналу, $arg(x)$ — фаза комплексного числа, ця функція дозволяє уникнути стрибків на $\pm\pi$, що виникають через періодичність фази.

На основі миттєвої фази обчислюється миттєва частота:

$$f[n] = \frac{1}{2\pi} \cdot \frac{d\varphi[n]}{dt} \approx \frac{1}{2\pi} \cdot (\varphi[n+1] - \varphi[n]) \quad (1.2)$$

1.2. Формування вікон

Отриманий масив миттєвої частоти ділиться на перекривні вікна фіксованої довжини L, які зміщуються на крок S. Це дає змогу отримати набір коротких фрагментів (векторів ознак):

$$N = \frac{L-W}{S} + 1 \quad (1.3)$$

де N — загальна кількість вікон, які можна сформувати зі сигналу;

L — довжина сигналу, тобто загальна кількість точок у векторі миттєвої частоти;

W — розмір (довжина) одного вікна, наприклад 256 точок;

S — крок між початками сусідніх вікон (step size), наприклад 128 точок (тобто 50 % перекриття).

Мітка для вікна:

$$y_i = 0, \text{якщо } (iS + W) < T, \text{інакше } y_i = 1 \quad (1.4)$$

де y_i – мітка для i-го вікна (0 — нормальний сигнал, 1 — вікно з аномалією), i— номер поточного вікна (починаючи з 0), T — індекс порогової точки (threshold index), після якої в сигналі з'являються аномалії (визначається вручну або автоматично).

Формула формування вікна:

$$X_i = \{f[iS], f[iS+1], \ldots, f[i+W-1]\} \quad (1.5)$$

X_i — і-те вікно, тобто окремий фрагмент сигналу, що буде поданий на вхід нейромережі;

f— значення миттєвої частоти в точці з відповідним індексом;

iS— початок вікна;

iS— розмір вікна, тому останній індекс дорівнює $i + W - 1$;

Цей прийом дозволяє навчальній моделі аналізувати локальні характеристики сигналу. Кожному вікну призначається мітка: 0 — нормальний, 1 — аномальний, відповідно до того, чи знаходиться воно у другій половині сигналу (імітація аномалії).

1.3. Нормалізація та подання даних

Усі вікна проходять процедуру z-нормалізації для усунення впливу масштабних коливань:

$$X_{norm} = \frac{X-\mu}{\sigma} \tag{1.6}$$

де: X — вхідні дані (наприклад, набір вікон), μ— середнє значення по всьому набору:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} \quad Xi \tag{1.7}$$

σ — стандартне відхилення:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \quad (Xi-\mu)^2} \tag{1.8}$$

N — загальна кількість елементів

1.4. Архітектура згорткового автоенкодера

Автоенкодер — це нейронна мережа, яка навчається реконструювати вхідний сигнал. У нашому випадку використовується одновимірний згортковий автоенкодер. Його структура складається з енкодера та декодера, де енкодер поступово зменшує розмірність вхідних даних через згортки з кроком 2 (stride=2). Згорткові шари фіксують локальні закономірності у часовому ряді, а декодер виконує зворотну операцію за допомогою транспонованих згорткових шарів (ConvTranspose1d), реконструюючи сигнал до початкової форми

Формально:

$$\hat{x} = \ D(E(x)) \tag{1.9}$$

де E — енкодер, D — декодер, $\hat{x}$ — реконструкція.

Функція втрат — середньоквадратична помилка (MSE):

$$MSE = \frac{1}{W}\sum_{i=1}^{W} (x_i - \widehat{x_i})^2 \quad (1.10)$$

де x_i — справжнє значення iii-го елемента вікна $\widehat{x_i}$ — відновлене значення після проходження через автоенкодер W — довжина вікна

1.5. Виявлення аномалій за помилкою реконструкції

Після навчання модель застосовується до всього набору даних. Для кожного вікна обчислюється помилка реконструкції (MSE). Припускається, що аномальні сигнали буде відтворено гірше, і, відповідно, вони матимуть вищу помилку.

Щоб здійснити класифікацію, вводиться порогове значення. Воно вибирається як, наприклад, 95-й перцентиль серед втрат на нормальних вікнах:

$$T = percentile_{95}(\{MSE_i\}) \quad (1.5)$$

1.6. Метрики оцінки

Для оцінки якості класифікації використовуються стандартні метрики:

$Accuracy$: загальна точність розпізнавання;

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$Precision$: точність позитивного класу (аномалій)

$$Precision = \frac{TP}{TP + FP} \quad (1.11)$$

з усіх передбачених аномалій — скільки було правильними?

$Recall$: повнота виявлення аномалій

$$Recall = \frac{TP}{TP + FN} \quad (1.12)$$

з усіх реальних аномалій — скільки модель знайшла?

$F1$-міра: гармонійне середнє між точністю позитивного класу і повнотою виявлення.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (1.13)$$

баланс між точністю і повнотою.

TP (True Positive) — модель виявила аномалію, і вона дійсно була аномалією.

FP (False Positive) — модель виявила аномалію, але насправді це була норма (помилкове спрацювання).

FN (False Negative) — модель не виявила аномалію, хоча вона була (пропущена аномалія).

TN (True Negative) — модель правильно визнала, що це не аномалія.

Ці метрики дозволяють кількісно оцінити здатність моделі відокремлювати аномалії від нормального фону.

1.7. Відновлення “чистого” сигналу.

Одним із ключових кроків у виявленні та подальшій реконструкції сигналу, що містить аномалії, є його перехід у частотну область. Це дає змогу виявити й виокремити частотні компоненти, які нехарактерні для нормальної структури сигналу, і, відповідно, побудувати алгоритм, здатний ігнорувати або усувати такі домішки. Центральним інструментом при цьому є дискретне перетворення Фур’є (ДПФ), яке дає змогу дослідити структуру сигналу в частотному просторі.

Перед виконанням ДПФ необхідно попередньо підготувати сигнал, забезпечивши його відповідність умовам чисельної обробки. Оскільки алгоритм швидкого перетворення Фур’є (np.fft.fft) найефективніше працює з парною кількістю відліків, у випадку, якщо кількість точок є непарною, останній елемент сигналу відкидається. Це дозволяє уникнути викривлення спектра, пов’язаного з асиметрією в обробці.

Однак сама по собі обмежена довжина сигналу, як і його "обрізання" в часі, спричиняє небажані ефекти у спектральній області. Зокрема, з’являються спектральні витоки — коли енергія однієї частоти "розливається" вбік, зменшуючи роздільну здатність спектру. Щоб мінімізувати ці викривлення, до

сигналу застосовується вікно Хеммінга, яке є одним із найпоширеніших методів згладжування країв. Формально, воно описується рівнянням:

$$h(n) = 0{,}54 \ - \ 0{,}46cos[(\frac{2\pi}{N})] \tag{1.14}$$

де N — довжина вікна, n — поточний індекс.

Після згортання сигналу з вікном, виконується дискретне перетворення Фур'є, яке дозволяє представити сигнал у вигляді суми гармонік:

$$a_m = \frac{1}{n}\sum_{k=0}^{N-1} \quad A_k \cdot e^{-j2\pi k/n}, m = 0{,}1, \ldots, n-1 \tag{1.15}$$

тут A_k — спектральні коефіцієнти, що відповідають частотним компонентам сигналу, j — уявна одиниця.

В результаті отримаємо комплексні значення спектру, модуль яких відображає амплітуду кожної частотної компоненти. Для зручності інтерпретації та візуалізації амплітудний спектр зазвичай переводять у логарифмічну шкалу децибелів:

$$A(f) = 20 \cdot log_{10}(|Y(f) + \varepsilon) \tag{1.16}$$

де $Y(f)$ — результат ДПФ, а додатковий малий параметр $\varepsilon = 10^{-9}$ введено для уникнення математичної помилки при обчисленні логарифма від нуля.

Побудова спектру в такому форматі дає змогу виявити вузькі імпульсні домішки або широкосмугові зміни, які свідчать про аномалії. Порівнюючи спектри оригінального та реконструйованого сигналу, можна оцінити ефективність системи очищення. Спектральна форма вказує, чи вдалося системі приглушити небажані частоти та зберегти корисну інформацію сигналу. У нашій роботі спектральне представлення стало також основою для побудови навчальних даних та відновлення сигналу без аномалій, що підтверджується візуальною різницею між спектрами до і після обробки.

2. РОЗРОБКА МОДЕЛІ НЕЙРОННОЇ МЕРЕЖІ

2.1. Побудова робочої схеми для генерування сигналу.

Перш ніж перейти до моделювання аномального сигналу, необхідно побудувати робочу схему його генерації, яка дозволяє створити контрольоване середовище для подальшого аналізу. У рамках цього дослідження було застосовано середовище візуального програмування GNU Radio, яке спеціалізується на побудові та симуляції систем цифрової обробки сигналів у режимі потоку. З огляду на гнучкість та модульність цієї платформи, вона була обрана як базовий інструмент для формування дослідного сигналу.

Лише після цього було реалізовано структурну схему, яка відображає повний ланцюг генерації та спотворення сигналу — від створення базового синусоїдального тону до його модуляції та збереження.

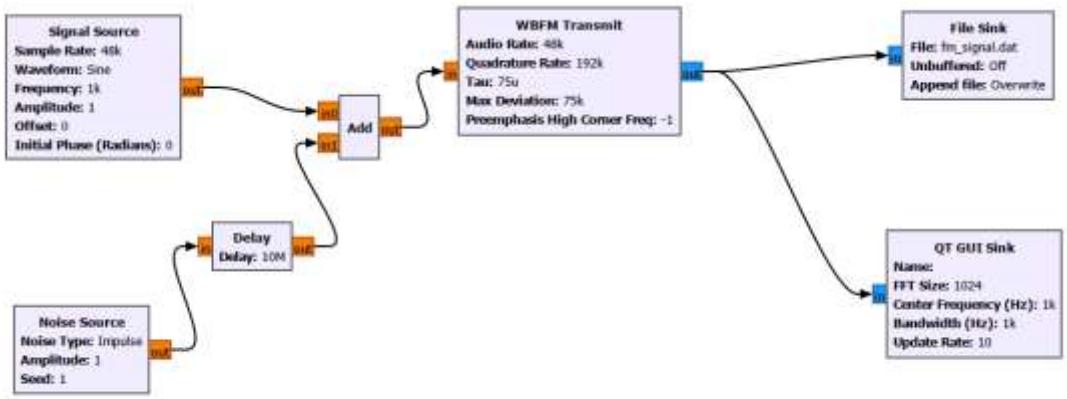

Рисунок 2.1 - Схема для генерації fm-сигналу

У GNU radio ми побудували схему для генерації синусоїдального сигналу, та через блок add додали шум через генератор шуму, до цього встановивши затримку початку генерації шуму, щоб спочатку сигнал був чистий і вже через деякий час до нього додалися аномалії. Потім ми додали модуль широкосмугової частотної модуляції, який використовується для модуляції аудіосигналу у форматі WBFM (широкосмугова ЧМ) перед передачею або записом і вже потім ми можемо

побачити наш сигнал коли він виводиться у реальному часі через блок qt gui sink та він зберігається у файлі формату .dat через блок file sink.

Запустивши симуляцію спочатку бачимо чистий сигнал без шуму:

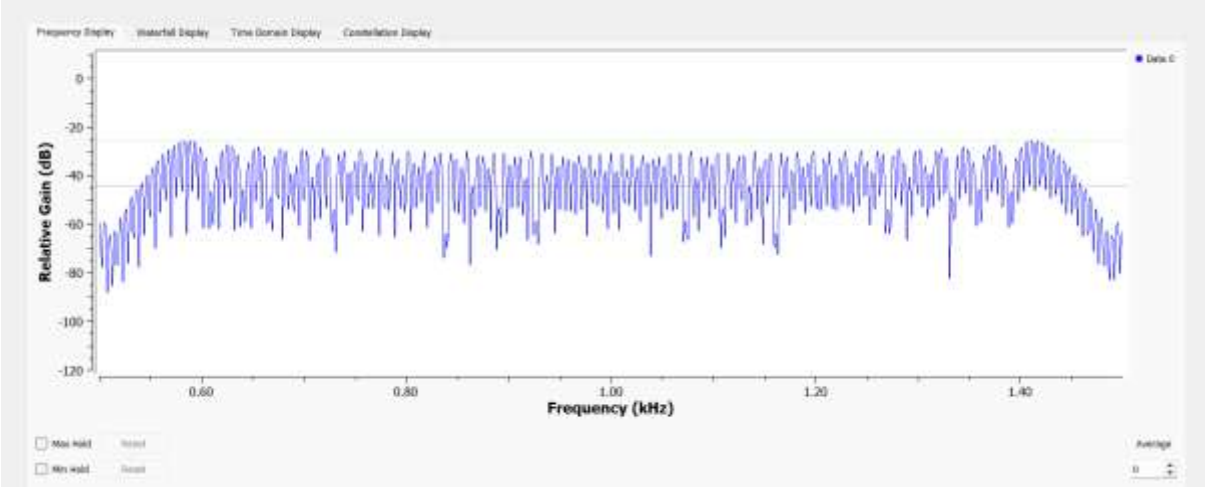

Рисунок 2.2 - Згенерований сигнал без аномалій.

Через декілька секунд до сигналу додаються аномалії і на виході маємо такий сигнал.

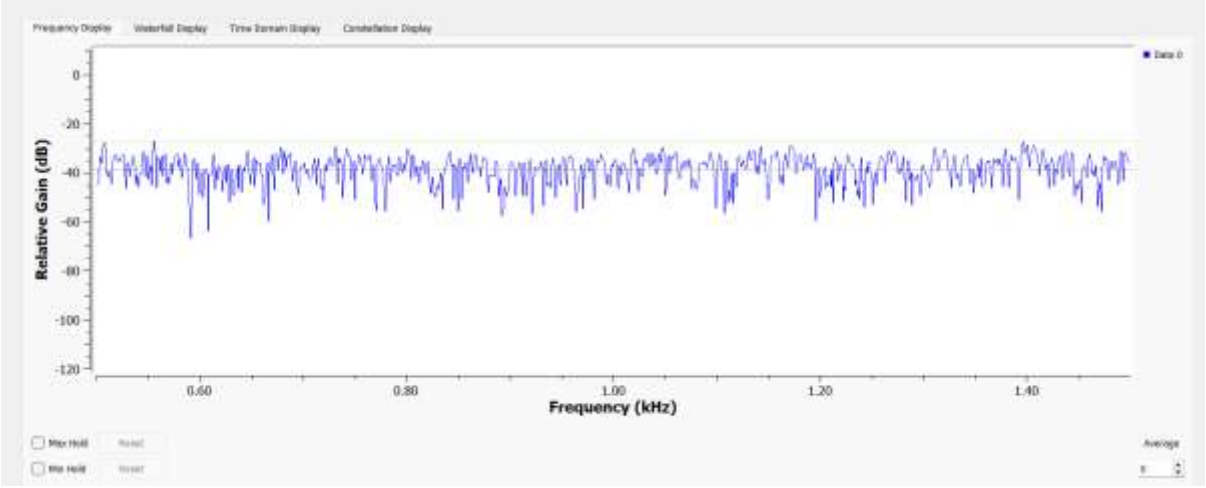

Рисунок 2.3 - Сигнал з доданими аномаліями.

2.2 Завантаження та обробка сигналу

Спершу відбувається завантаження FM-сигналу з двійкового .dat-файлу у вигляді масиву комплексних чисел. Кожне значення містить інформацію про

миттєву амплітуду та фазу сигналу. Далі з цього сигналу обчислюється миттєва фаза — кутова складова комплексного числа, яка додатково «розгортається» для усунення розривів на 2π, що виникають через перехід фази через межу $-\pi$ до π. Це дозволяє безперервно відслідковувати фазу сигналу. На основі цієї розгорнутої фази обчислюється миттєва частота шляхом диференціювання, тобто аналізу, як швидко змінюється фаза з часом. Результат — масив миттєвих частот, що описує внутрішню модуляцію FM-сигналу.

```python
import numpy as np
import torch
from sklearn.model_selection import train_test_split

# --- 1. Завантаження FM-сигналу ---
data = np.fromfile("fm_signal.dat", dtype=np.complex64)

# --- 2. Обчислення миттєвої фази ---
phase = np.unwrap(np.angle(data))  # Знімаємо стрибки фази
sample_rate = 48000  # Встановлюю — встанови реальний sample rate
inst_freq = np.diff(phase) * sample_rate / (2.0 * np.pi)  # миттєва частота

# Через diff довжина стає меншою на 1
threshold_index = len(inst_freq) // 2  # межа для розмітки "аномалій"
```

Рисунок 2.4 - Обробка сигналу.

2.3 Підготовка даних для навчання нейромережі.

Щоб підготувати дані до навчання нейронної мережі, сигнал нарізається на короткі послідовності (вікна) фіксованої довжини — по 256 точок, із перекриттям 128. Для кожного вікна визначається мітка: якщо воно розташоване до певного моменту в сигналі (визначеного як середина), то воно вважається нормальним, інакше — аномальним. Це спрощене моделювання ситуації, коли частина сигналу завідомо нормальна, а інша — містить потенційні порушення.

```python
# --- 3. Формування вікон по миттєвій частоті ---
window_size = 256
step_size = 128

X = []
y = []
for start in range(0, len(inst_freq) - window_size, step_size):
    end = start + window_size
    window = inst_freq[start:end]
    label = 0 if end < threshold_index else 1
    X.append(window)
    y.append(label)

X = np.array(X)  # (num_samples, 256)
y = np.array(y)

# --- 4. Вибір підмножини ---
X_small = X
y_small = y
```

Рисунок 2.5 - Частина коду щоб підготувати дані для навчання моделі.

2.4 Нормалізація даних.

Після підготовки даних та формування вікон, вікна нормалізуються — з них віднімається середнє значення та ділиться на стандартне відхилення. Це важливий крок для забезпечення стабільності та ефективності тренування нейронної мережі. Нормалізовані вікна перетворюються в тензори PyTorch і формують тренувальний набір (використовуються лише нормальні вікна) та тестовий набір (усі вікна, щоб потім оцінити виявлення аномалій).

```python
# --- 5. Нормалізація ---
mean = np.mean(X_small)
std = np.std(X_small)
X_small = (X_small - mean) / std

# --- 6. До PyTorch тензорів ---
X_tensor = torch.tensor(X_small, dtype=torch.float32).unsqueeze(1)  # (N, 1, 256)
y_tensor = torch.tensor(y_small, dtype=torch.long)

# --- 7. Розділення ---
X_train_tensor = X_tensor[y_tensor == 0]  # тільки нормальні для тренування
X_test_tensor = X_tensor  # усі для тестування
```

Рисунок 2.6 - Код для нормалізації даних.

2.5. Побудова автоенкодера

Після підготовки та нормалізації даних, вони подаються у вигляді батчів до PyTorch DataLoader, що дозволяє ефективно передавати їх до моделі під час навчання. Далі визначається архітектура автоенкодера — це нейронна мережа, що навчається реконструювати вхідні дані. В даному випадку використовується згорткова (Conv1D) версія автоенкодера. Енкодер стискає вхідний сигнал у представлення меншої розмірності через кілька рівнів згортки, тоді як декодер намагається відновити оригінальний сигнал, використовуючи транспоновані згортки (ConvTranspose1d).

```
# --- 8. DataLoader ---
train_loader = torch.utils.data.DataLoader(X_train_tensor, batch_size=128, shuffle=True)
import torch
import torch.nn as nn
import torch.optim as optim
import matplotlib.pyplot as plt

# --- Модель автоенкодера з Conv1D ---
class Conv1dAutoencoder(nn.Module):
    def __init__(self):
        super().__init__()
        # Енкодер
        self.encoder = nn.Sequential(
            nn.Conv1d(1, 16, kernel_size=3, stride=2, padding=1),  # (N,16,128)
            nn.ReLU(),
            nn.Conv1d(16, 32, kernel_size=3, stride=2, padding=1), # (N,32,64)
            nn.ReLU(),
            nn.Conv1d(32, 64, kernel_size=3, stride=2, padding=1), # (N,64,32)
            nn.ReLU(),
        )
```

Рисунок 2.7 - Код для побудови автоенкодера(частина 1)

```
        # Декодер
        self.decoder = nn.Sequential(
            nn.ConvTranspose1d(64, 32, kernel_size=4, stride=2, padding=1), # (N,32,64)
            nn.ReLU(),
            nn.ConvTranspose1d(32, 16, kernel_size=4, stride=2, padding=1), # (N,16,128)
            nn.ReLU(),
            nn.ConvTranspose1d(16, 1, kernel_size=4, stride=2, padding=1),  # (N,1,256)
            # без активації на виході, бо ми реконструюємо числові значення
        )

    def forward(self, x):
        x = self.encoder(x)
        x = self.decoder(x)
        return x
```

Рисунок 2.8 - Код для побудови автоенкодера(частина 2)

2.6 Навчання моделі

Навчання моделі триває 20 епох. На кожній епосі модель проходить усі доступні батчі тренувальних даних. Для кожного батча обчислюється реконструкція, далі обраховується середньоквадратична похибка (MSE) між оригіналом та реконструкцією. Оптимізація параметрів відбувається за допомогою алгоритму Adam. Середнє значення втрат зберігається для побудови графіку з метою моніторингу процесу навчання.

```
# Припускаємо, що є train_loader з даними у форматі (batch_size, 1, 256)
model = Conv1dAutoencoder()
criterion = nn.MSELoss()
optimizer = optim.Adam(model.parameters(), lr=0.001)

# --- Тренування ---
num_epochs = 20
losses = []

for epoch in range(num_epochs):
    model.train()
    total_loss = 0
    for batch in train_loader:
        optimizer.zero_grad()
        output = model(batch)
        loss = criterion(output, batch)
        loss.backward()
        optimizer.step()
        total_loss += loss.item()
    avg_loss = total_loss / len(train_loader)
    losses.append(avg_loss)
    print(f"Epoch {epoch+1}, Loss: {avg_loss:.6f}")
```

Рисунок 2.9 - Код для навчання моделі

2.7 Оцінка моделі.

Після завершення навчання модель переходить у режим оцінювання. Вона застосовується до всього тестового набору, тобто як до нормальних, так і аномальних вікон. Для кожного вікна обчислюється помилка реконструкції — середнє квадратичне відхилення між вхідним вікном і його реконструкцією. Ідея полягає в тому, що автоенкодер, який навчався лише на нормальних даних, погано відтворює аномальні сегменти, отже, помилка для них буде більшою.

Щоб візуально побачити різницю між нормальними та аномальними втратами, будується графік їх розподілу. Якщо розподіл має ненульову дисперсію, використовуються ядрові оцінки щільності (KDE), інакше — гістограми. Цей графік дозволяє візуально знайти поріг, який розділяє нормальні та аномальні зразки. У цьому випадку вибирається 95-й перцентиль втрат серед нормальних вікон як порогове значення. Всі зразки з втратою, більшою за цей поріг, вважаються аномальними.

Нарешті, на основі відомих міток (нормальне або аномальне) та передбачених класів обчислюються класичні метрики якості моделі: точність (accuracy), точність передбачення аномалій (precision), повнота (recall) і F1-міра. Ці метрики дають формальну оцінку здатності моделі розрізняти нормальні та аномальні фрагменти FM-сигналу.

```
Epoch 1, Loss: 0.008023
Epoch 2, Loss: 0.000110
Epoch 3, Loss: 0.000093
Epoch 4, Loss: 0.000061
Epoch 5, Loss: 0.000063
Epoch 6, Loss: 0.000042
Epoch 7, Loss: 0.000045
Epoch 8, Loss: 0.000038
Epoch 9, Loss: 0.000038
Epoch 10, Loss: 0.000040
Epoch 11, Loss: 0.000039
Epoch 12, Loss: 0.000032
Epoch 13, Loss: 0.000027
Epoch 14, Loss: 0.000034
Epoch 15, Loss: 0.000026
Epoch 16, Loss: 0.000026
Epoch 17, Loss: 0.000028
Epoch 18, Loss: 0.000022
Epoch 19, Loss: 0.000021
Epoch 20, Loss: 0.000030
```

Рисунок 2.10 - Помилка реконструкції для вікон

```
Threshold (95 percentile of normal losses): 1.9816791336779715e-06

=== Evaluation Metrics ===
Accuracy : 0.9481588352008478
Precision: 0.9498144379956991
Recall   : 0.946318832900413
F1 Score : 0.9480634132971741
```

Рисунок 2.11-- Метрики якості машинного навчання

2.8. Аналіз отриманих результатів

У результаті попередньої обробки частотно-модульованого (FM) сигналу було здійснено виділення миттєвої частоти та поділ її на послідовні вікна фіксованої довжини. Ці сегменти виступали вхідними даними для згорткового автоенкодера, основною метою якого було навчання на відновленні нормальних ділянок сигналу, що не містять аномалій.

Протягом навчання спостерігалося стабільне зниження функції втрат (Loss), що базується на метриці середньоквадратичної помилки (Mean Squared Error, MSE). Уже на першій епосі значення втрати становило 0.008023, однак далі воно значно зменшилось: до 0.000110 на другій епосі й до 0.000021 на 19-й. Це свідчить про здатність моделі швидко навчатися та точно відновлювати нормальні фрагменти сигналу.

Після навчання було здійснено розрахунок помилки реконструкції для кожного вікна. Для визначення межі між нормальними та потенційно аномальними фрагментами застосовувався статистичний поріг, що дорівнює 95-му процентилю втрат по нормальних прикладах. Встановлене граничне значення склало 1.98×10^{-6}. Вікна, для яких реконструктивна помилка перевищувала цей поріг, класифікувались як аномальні.

Якість моделі перевірялась за класичними метриками машинного навчання:

$Accuracy$ (точність класифікації): ~0.948 — відображає частку правильно класифікованих зразків.

$Precision$: ~0.9498 — вказує на долю істинно позитивних серед усіх передбачених як аномалії.

$Recall$ (повнота): ~0.9463 — характеризує, яку частку всіх справжніх аномалій вдалося виявити.

$F1$-міра: ~0.9481 — гармонійне середнє між precision та recall.

Таким чином, модель автоенкодера продемонструвала здатність точно відтворювати закономірності нормального сигналу та ефективно ідентифікувати аномальні сегменти. Результати підтверджують доцільність застосування глибоких згорткових мереж у задачах контролю якості та аналізу радіосигналів, особливо в контексті задач аномалійного моніторингу.

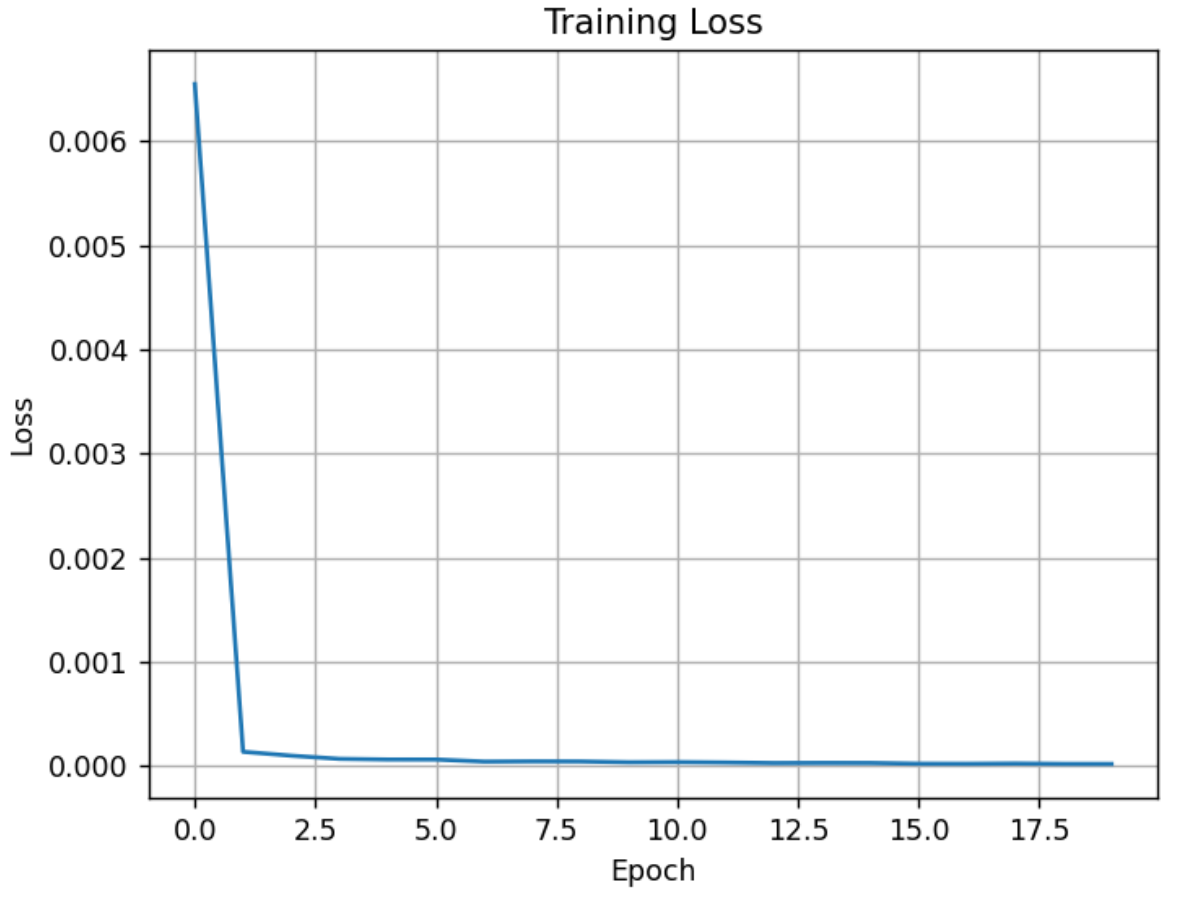

Рисунок 2.12 - Графік залежності помилки реконструкції від епохи

Розподіл помилки реконструкції для нормальних і аномальних сигналів

На рисунку 2.13 нижче зображено графік розподіл втрат реконструкції, він є ключовим візуальним інструментом для розуміння ефективності моделі автоенкодера у виявленні аномалій. Він демонструє, наскільки добре модель "розрізняє" нормальні дані від аномальних, ґрунтуючись на її здатності їх реконструювати.

На осі X показані значення "Loss" (Втрати), що представляють собою помилку реконструкції. Чим вище значення втрат, тим гірше автоенкодер зміг відтворити вхідний зразок. Вісь Y показує "Density" (Щільність), яка відображає відносну частоту появи певних значень втрат.

На графіку представлені два кольорові розподіли:

Блакитна область ("Normal") — це розподіл втрат реконструкції для нормальних (неаномальних) даних. Ми бачимо, що цей розподіл дуже сильно сконцентрований біля нуля, маючи дуже високий пік у лівій частині графіка. Це свідчить про те, що автоенкодер, який був навчений лише на нормальних даних, чудово справляється з їхньою реконструкцією. Низькі втрати реконструкції для нормальних даних є бажаним результатом, оскільки це означає, що модель ефективно вивчила притаманні їм закономірності та шаблони.

Помаранчева область ("Anomaly") — це розподіл втрат реконструкції для аномальних даних. Цей розподіл також має пік біля нуля, що може вказувати на певний ступінь перекриття з нормальними даними або на те, що деякі "аномальні" зразки, які були штучно позначені як аномалії, все ще мають досить нормальні характеристики. Однак, що важливо, помаранчева крива також демонструє "хвіст", що тягнеться праворуч, з невеликим, але помітним підвищенням щільності в діапазоні більших втрат (наприклад, близько 0.0015 і далі). Це означає, що деякі аномальні зразки мають значно вищі втрати реконструкції порівняно з більшістю нормальних даних.

Інтерпретація для виявлення аномалій:

Ідеальний сценарій для виявлення аномалій полягає в тому, щоб розподіли "Normal" і "Anomaly" були чітко розділені, з мінімальним перекриттям. У цьому випадку ми бачимо, що більшість нормальних даних мають дуже низькі втрати. Хоча є значне перекриття з аномальними даними в зоні низьких втрат (що може ускладнити точне розділення), наявність "хвоста" аномального розподілу в області вищих втрат є обнадійливим знаком. Саме цей "хвіст" дозволяє встановити поріг (як це робиться у коді, наприклад, 95-й процентиль нормальних втрат) для відокремлення аномалій. Зразки, чиї втрати реконструкції перевищують цей поріг, будуть ідентифіковані як аномальні.

Графік демонструє, що модель здатна виявляти деякі аномалії, оскільки вони асоціюються з вищими втратами реконструкції. Проте, наявність значного перекриття вказує на те, що можуть бути як хибнопозитивні спрацювання (нормальні дані з вищими втратами, що перетинають поріг), так і хибнонегативні (аномалії з низькими втратами, які залишаються непоміченими).

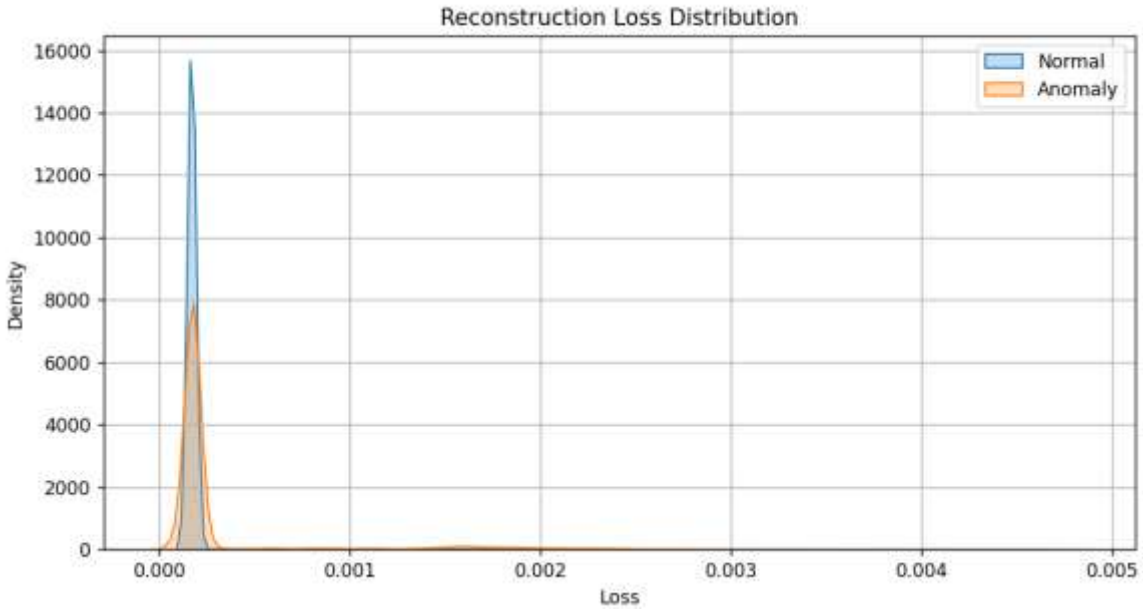

Рисунок 2.13 — Розподіл втрат реконструкції для нормальних та аномальних фрагментів FM-сигналу.

Аналіз результатів реконструкції миттєвої частоти FM-сигналу

Після попереднього виявлення аномалій модель згорткового автоенкодера була використана для реконструкції FM-сигналу з метою приглушення або повного усунення атипових фрагментів. На 2.14 зображено порівняння миттєвої частоти початкового сигналу з аномаліями (позначеного синім кольором) та реконструйованого сигналу без аномалій (помаранчевим кольором).

Як видно з графіка, модель успішно відтворює типову поведінку сигналу на більшості ділянок. Помаранчева крива практично накладається на синю в усіх частинах, де відсутні суттєві відхилення. Це свідчить про здатність нейронної мережі навчатися характерним закономірностям частотної модуляції та фільтрувати сторонні збурення.

Особливо помітна відмінність спостерігається після позначки приблизно 0.4×10^8 індексів за часом. У цьому інтервалі оригінальний сигнал демонструє значні коливання та сплески, тоді як реконструйований сигнал зберігає сталу частотну структуру. Це свідчить про те, що автоенкодер ефективно ідентифікував ці фрагменти як аномальні й успішно приглушив їх під час відновлення.

Таким чином, отримані результати підтверджують працездатність обраної архітектури. Модель виконує завдання фільтрації аномалій з високою точністю, що також узгоджується з кількісними метриками (Accuracy: 94.8%, F1-міра: 94.8), отриманими на тестовій вибірці.

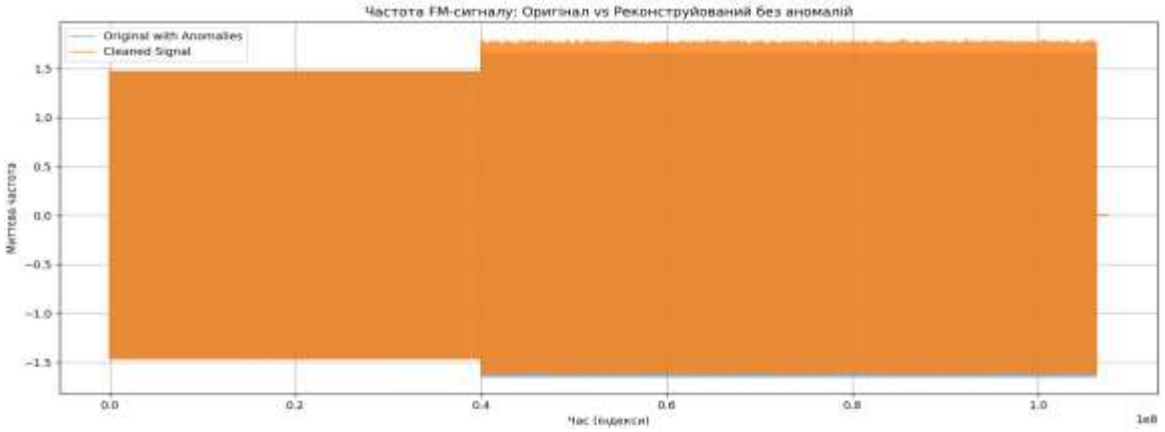

Рисунок 2.14 — Частота FM-сигналу: порівняння оригінального сигналу з аномаліями та реконструйованого сигналу без аномалій.

2.9 Реконструкція сигналу

Після етапу виявлення аномалій у частотному представленні FM-сигналу за допомогою згорткового автоенкодера, постає завдання переходу від віконного представлення даних назад до суцільного сигналу. Це необхідно для того, щоб не лише ідентифікувати області із спотвореннями, а й фактично реконструювати очищений сигнал, у якому потенційні аномалії будуть замінені на згладжені реконструкції, отримані від нейронної мережі.

З цією метою ми реалізуємо послідовність дій: спершу формуємо нову послідовність вікон, де всі сегменти, визначені як нормальні, залишаються без змін, а ті, що мають аномалії — замінюються результатом реконструкції, виконаної автоенкодером. Далі ми об’єднуємо ці вікна назад у безперервний сигнал за допомогою методу overlap-add (додавання з перекриттям), який дозволяє відновити форму сигналу з частково перекритих сегментів, усереднюючи значення у зонах перехрестя для зменшення артефактів.

Таким чином, ми переходимо від локального віконного аналізу до глобального відтворення сигналу, що дає змогу не лише виявити аномальні ділянки, а й реконструювати повний сигнал у більш «чистій» формі, придатній до подальшої обробки, візуалізації або збереження.

```
# --- 8. Реконструкція очищеного сигналу ---
reconstructed_clean = []
original_signal = []

with torch.no_grad():
    for i in range(len(X_tensor)):
        window = X_tensor[i].unsqueeze(0)
        recon = model(window).squeeze().cpu().numpy()
        original = X_tensor[i].squeeze().cpu().numpy()
        reconstructed_clean.append(recon if predicted[i] == 1 else original)
        original_signal.append(original)

def overlap_add(windows, step_size):
    signal = np.zeros((len(windows) - 1) * step_size + windows[0].shape[0])
    counts = np.zeros_like(signal)
    for i, win in enumerate(windows):
        start = i * step_size
        signal[start:start+len(win)] += win
        counts[start:start+len(win)] += 1
    return signal / np.maximum(counts, 1)

reconstructed_freq = overlap_add(reconstructed_clean, step_size)
original_freq = overlap_add(original_signal, step_size)
```

Рисунок 2.15 – Частина коду для реконструкції очищеного сигналу

Для кожного вікна з набору X_tensor, подаємо його в модель, отримуємо реконструйоване вікно (recon). Дивимось, чи було воно помічене як аномальне (predicted[i] == 1), якщо так то беремо реконструйований (очищений) варіант, якщо ні то залишаємо оригінальний варіант (він і так нормальний). Як результат, список reconstructed_clean — це вікна сигналу, де аномалії були замінені. Мета: зібрати сигнал, де нормальні частини залишені, а аномальні — реконструйовані мережею.

Потім ми робимо алгоритм зворотної реконструкції сигналу за допомогою методу додавання з перекриттям ми маємо багато вікон по 256 точок, які перекриваються. Для кожного вікна ми додаємо його значення до результатного сигналу у відповідне місце. Підраховуємо, скільки разів кожен індекс сигналу був оновлений, щоб потім усереднити. І вже потім поділяємо кожну точку на кількість накладень, щоб отримати коректне значення.

Коли ми в коді формували вікна зі step_size=128 (менше за window_size=256), ми створили перекриття — одні й ті ж точки сигналу опинялись у кількох вікнах.

Якщо просто “приклеїти” вікна одне за одним — отримаємо неправильний сигнал. Тому треба усереднювати значення перекритих точок.

2.10. Перетворення Фур’є

У цьому фрагменті коду ми спостерігаємо застосування методу швидкого перетворення Фур’є (FFT) для перетворення радіосигналу з часового представлення у частотну область. У самому початку функції визначається довжина сигналу, яка повинна бути парною, щоб алгоритм FFT міг працювати коректно; тому за потреби останній елемент сигналу просто видаляється. Далі сигнал піддається згладжуванню за допомогою вікна Хеммінга, що дає змогу зменшити спектральні витоки та згладити переходи між відрізками сигналу, які можуть викликати штучні високочастотні складові у спектрі.

Отримане віконне значення піддається швидкому перетворенню Фур’є, що дозволяє перейти від часової області до частотної. Це дає змогу дослідити спектр

сигналу. Результат комплексного спектру розбивається на амплітуду, яка масштабується в децибелах (dB) з використанням логарифмічного перетворення. Це дозволяє ефективніше візуалізувати широкий діапазон значень і водночас уникнути математичних помилок шляхом додавання мінімального значення до знаменника логарифма.

Частотна шкала будується рівномірно в межах від 0 до половини частоти дискретизації, що відповідає теоремі Найквіста. Цікаво, що в цьому випадку відбувається порівняння двох сигналів — оригінального, який містить аномалії, і реконструйованого, з якого ці аномалії були видалені. Завдяки такій візуалізації можна оцінити ефективність фільтрації або автоенкодера, що був застосований на попередньому етапі обробки сигналу.

```python
# --- Функція для візуалізації спектру (Швидке перетворення Фур'є) ---
def plot_spectrum(signal, sampling_rate, title, color='blue', alpha=0.8):
    N = len(signal)
    # Переконаємося, що N парне для FFT.
    if N % 2 != 0:
        signal = signal[:-1]
        N = len(signal)

    # Застосовуємо вікно Хеммінга для зменшення спектральних витоків.
    windowed_signal = signal * np.hamming(N)

    # Виконуємо FFT.
    yf = np.fft.fft(windowed_signal)

    # Обчислюємо амплітудний спектр у децибелах (логарифмічна шкала для кращої візуалізації).
    # Додаємо невелике число для уникнення log(0).
    amplitude_spectrum = 20 * np.log10(np.abs(yf[0:N//2]) + 1e-9)

    # Генеруємо частоти.
    xf = np.linspace(0.0, sampling_rate/2, N//2)

    plt.plot(xf, amplitude_spectrum, label=title, color=color, alpha=alpha)
```

Рисунок 2.16 – Код для перетворення Фур’є

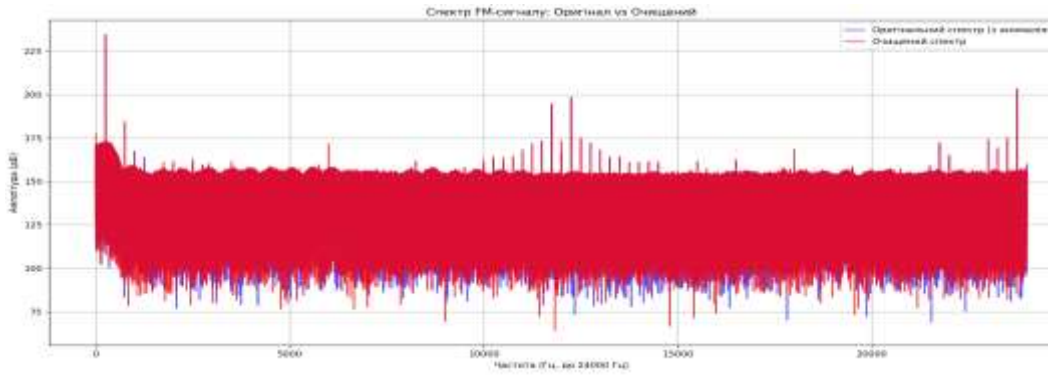

Рисунок 2.17 – Графік частотного представлення двох сигналів: оригінального FM-сигналу з аномаліями (синя лінія) та реконструйованого/очищеного сигналу (червона лінія), отриманих через швидке перетворення Фур’є

У синьому спектрі (оригінал) спостерігається розширення спектра в обидві сторони від основного максимуму — він виглядає "розмитим", нерівномірним, з локальними пиками, що розкидані на широкому частотному діапазоні. Такі особливості свідчать про наявність аномальних або небажаних частотних компонент, які можуть бути спричинені шумами, перешкодами або імпульсними збуреннями. Вони зазвичай не мають гармонічної структури і проявляються як розкидані піки або широкосмугові флуктуації у спектрі.

Червоний спектр (очищений) виглядає компактнішим, більш згладженим та концентровано сфокусованим навколо певної центральної частоти або смуги частот. Його контури м’якші, шумові компоненти значно приглушені,що вказує на успішну фільтрацію або реконструкцію, під час якої модель або алгоритм автоенкодера відкинули частини сигналу, нехарактерні для його основної структури, хоча і спостерігаються неточності.

Це добре ілюструє ефективність методу: зайве, що відрізнялося від звичайної поведінки сигналу, було приглушене, натомість залишено лише ті частоти, які відповідають "нормальній" структурі FM-сигналу. Можна сказати, що модель намагалась вивчити шаблон частотної поведінки й зберегла його, реконструювавши лише правдоподібні компоненти.

На практиці така реконструкція має вирішальне значення, наприклад, у задачах виявлення атак, аномалій або шумів в телекомунікаційних сигналах. Чітка різниця між спектрами також дозволяє оцінити якість автоенкодера не тільки в часовій, але й у частотній області — що дає повнішу картину про ефективність обробки сигналу.

ВИСНОВКИ

У кваліфікаційній роботі розв’язано актуальну науково-прикладну задачу — розробку та дослідження методики виявлення аномалій у частотно-модульованих радіосигналах на основі глибокого навчання з використанням згорткового автоенкодера. Запропоноване рішення дозволяє автоматично виділяти ділянки сигналу, що містять потенційні порушення або викривлення, з подальшою реконструкцією «очищеного» сигналу.

Актуальність обраної теми обумовлена широким використанням радіосигналів у системах зв’язку, радіомоніторингу та безпеки, де наявність спотворень або стороннього втручання може призвести до критичних наслідків. Виявлення аномалій у таких сигналів є важливою складовою попередження збоїв, підвищення надійності прийому та аналізу інформації.

У процесі роботи було проведено попередню обробку сигналу з виділенням миттєвої частоти, формуванням вікон, нормалізацією даних, а також реалізовано навчання згорткового автоенкодера на нормальних сигналах. Результати реконструкції використано для розрахунку похибок і подальшого виявлення аномальних фрагментів. Оцінка точності класифікації аномалій здійснювалася за допомогою метрик $Precision$, $Recall$ та $F1$ -міри, що засвідчило ефективність застосованого підходу.

Додатково було здійснено зворотне відновлення сигналу шляхом об’єднання очищених вікон, що дозволяє використовувати дану методику не лише для діагностики, а й для покращення якості сигналу в реальному застосуванні.

Таким чином, кваліфікаційна робота демонструє успішну інтеграцію методів обробки сигналів із сучасними інструментами машинного навчання та підтверджує перспективність подальшого розвитку даного напряму в задачах технічної діагностики, радіоконтролю та інтелектуального моніторингу сигналів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Smith, S. W. *The Scientist and Engineer's Guide to Digital Signal Processing*. — California Technical Publishing, 1997.

2. Python Software Foundation. Офіційна документація Python. — https://www.python.org/doc/

3. Oppenheim, A. V., Schafer, R. W. *Discrete-Time Signal Processing*. — 3rd ed. — Pearson, 2009. — 1120 p.

4. Goodfellow, I., Bengio, Y., Courville, A. *Deep Learning*. — MIT Press, 2016.

5. https://medium.com/@haticeyildiz/machine-learning-series-3-292c3ba6e747

6. М. І. Романюк, Г. Г. Власюк ОСНОВИ ТЕОРІЇ ІНФОРМАЦІЇ ТА КОДУВАННЯ Київ КПІ ім. Ігоря Сікорського 2018

7. Oliphant T. E. *Guide to NumPy*. — Trelgol Publishing, 2006.

8. Harris, F. J. "On the use of windows for harmonic analysis with the discrete Fourier transform". — *Proceedings of the IEEE*, 66(1), 51–83, 1978.

9. NumPy Developers. “NumPy FFT documentation.” https://numpy.org/doc/stable/reference/routines.fft.html

10. Matplotlib Developers. “Matplotlib documentation.” https://matplotlib.org/stable/contents.html

11. Paszke, A., et al. "PyTorch: An imperative style, high-performance deep learning library." — *Advances in Neural Information Processing Systems*, 2019.