

ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ**КАФЕДРА ЕЛЕКТРОННИХ ОБЧИСЛЮВАЛЬНИХ МАШИН****КВАЛІФІКАЦІЙНА РОБОТА****«Інформаційна система обробки даних з
безпроводної мережі»**

Студент гр. КІУКІу-22-2

Готра Г.І.

Керівник

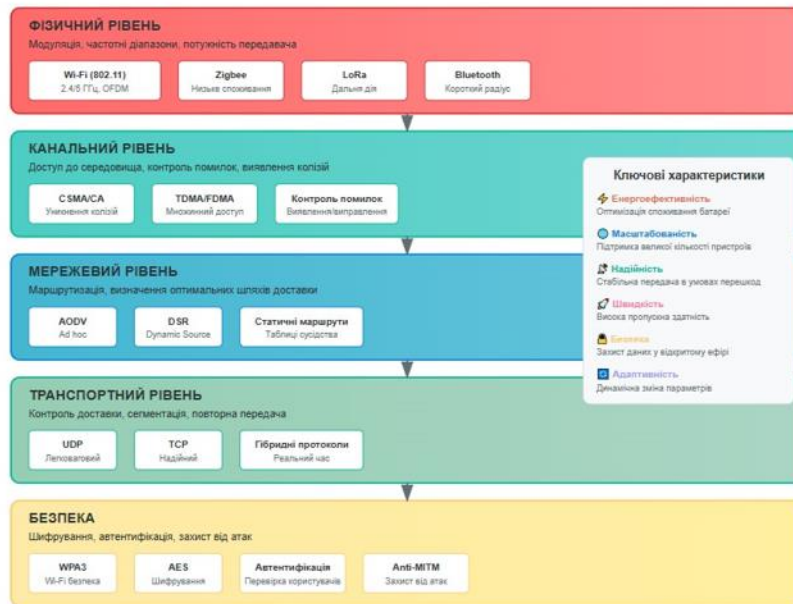
ст. викл. Дяченко В.О.**Харків 2025*****Мета та завдання кваліфікаційної роботи***

Мета кваліфікаційної роботи: створення функціональної інформаційної системи, яка забезпечує обробку та візуалізацію даних, отриманих з безпроводної мережі, з використанням програмних засобів Matlab і Google Colab.

Завдання:

- проаналізувати принципи організації безпроводних мереж;
- визначити оптимальні методи обробки даних;
- розробити алгоритми їх представлення;
- перевірити працездатність створеної системи у тестовому середовищі.

Протоколи передачі даних у безпроводних мережах



3

Обробка даних з безпроводних мереж



4

Огляд існуючих програмних засобів

ПАКЕТНІ АНАЛІЗАТОРИ

Параметр	Wireshark	Kismet
Призначення	Захоплення та аналіз пакетів	Пасивне виявлення мереж
Платформа	Windows, Linux, macOS	Linux, macOS, FreeBSD
Інтерфейс	Графічний (Qt)	Web UI (localhost:2501)
Ліцензія	Open Source	Open Source
Складність	Середня	Середня
Переваги	Потужна фільтрація, широка підтримка протоколів	Пасивний режим, GPS, wardriving
Недоліки	Великий обсяг даних, потребує досвіду	Вимагає monitor mode, складне налаштування
Типові користувачі	Мережеві адміністратори, інженери безпеки	Пентестери, дослідники безпеки

КОМЕРЦІЙНІ СИСТЕМИ МОНІТОРИНГУ

Параметр	PRTG	SolarWinds NPM	Nagios
Призначення	Універсальний моніторинг	Корпоративний моніторинг	Системний моніторинг
Платформа	Windows	Windows Server	Linux, Unix
Інтерфейс	Web UI	Web UI	Web UI + Config
Ліцензія	Freemium (100 сенсорів)	Комерційна	Open Source
Складність	Легка	Середня	Висока
Переваги	Простота, безкоштовна версія	Корпоративний рівень, масштабованість	Безкоштовний, гнучкість
Недоліки	Обмеження версії, тільки Windows	Дуже дорогий	Складна конфігурація
Типові користувачі	SMB, IT відділи	Великі корпорації, MSP	Linux адміністратори, DevOps

SDR ТА МОДЕЛЮВАННЯ

Параметр	GNU Radio	MATLAB Wireless Toolbox
Призначення	SDR блокові схеми	5G/LTE моделювання
Платформа	Linux, Windows, macOS	Windows, Linux, macOS
Інтерфейс	GNU Radio Companion	MATLAB IDE + Apps
Ліцензія	Open Source	Комерційна
Складність	Висока	Середня
Переваги	Гнучкість, наукова підтримка	Професійний, готові моделі
Недоліки	Крута крива навчання	Дорогий, потребує ліцензії
Типові користувачі	Інженери-дослідники, радіоаматори	Інженери 5G/LTE, корпорації

5

Вимоги до інформаційної системи

СПЕЦИФІКАЦІЯ ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ оброби даних в бездротових мережах

ФУНКЦІОНАЛЬНІ ВИМОГИ

Збір та прийом даних з мережевих вузлів <small>Бездротові мережі та мережеві вузли</small>	Поліаритмічна обробка сигналів <small>Бінарні, аналогові, цифрові</small>
Аналітична обробка та моделювання <small>Статистичні методи, машинне навчання</small>	Візуалізація та генерація звітів <small>Дашборди, графіки, інтерактивні звіти</small>

ТЕХНІЧНІ ВИМОГИ

Програмне середовище MATLAB <small>Інтеграція з MATLAB, чистий код</small>	Платформа Google Cloud <small>Гнучка архітектура, масштабованість</small>
Інтеграція з апаратним забезпеченням <small>Специфічні адаптери, драйвери</small>	Підтримка потокових даних <small>Розподілені системи, масштабованість</small>

ВИМОГИ ДО ПРОДУКТИВНОСТІ

Обробка в реальному часі <small>Масштабованість, оптимізація алгоритмів</small>	Оптимізація алгоритмічної складності <small>Аналіз складності, вибір оптимальних алгоритмів</small>
Масштабованість системи <small>Масштабованість, розподілені системи</small>	Ресурсна ефективність <small>Ефективне використання ресурсів</small>

НАДІЙНІСТЬ ТА СПІВІСНУВАННЯ ДО ЗБОЇВ

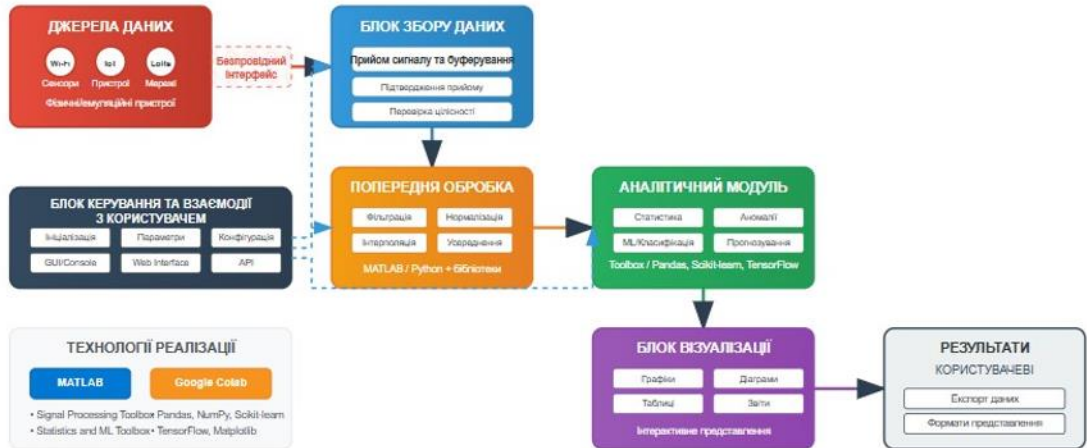
Обробка некоректних даних <small>Валидация, очищення, фільтрація</small>	Автоматичне відновлення з'єднань <small>Повторні спроби, таймауты</small>
Система резервного копіювання <small>Резервні копії, відновлення даних</small>	Моніторинг та діагностика <small>Моніторинг стану, діагностика помилок</small>

МОДУЛЬНІСТЬ І РОЗШИРЮВАНІСТЬ АРХІТЕКТУРИ

Підключення нових типів сенсорів <small>Універсальні адаптери, стандартизовані дані</small>	Адаптація протоколів зв'язку <small>Гнучкість до змін протоколів зв'язку</small>	Впровадження нових алгоритмів <small>Модульна архітектура, розширюваність</small>
Гнучкість до змін вимог <small>Модульність, стандартизовані дані</small>		Компонентна архітектура <small>Модульність, стандартизовані дані</small>

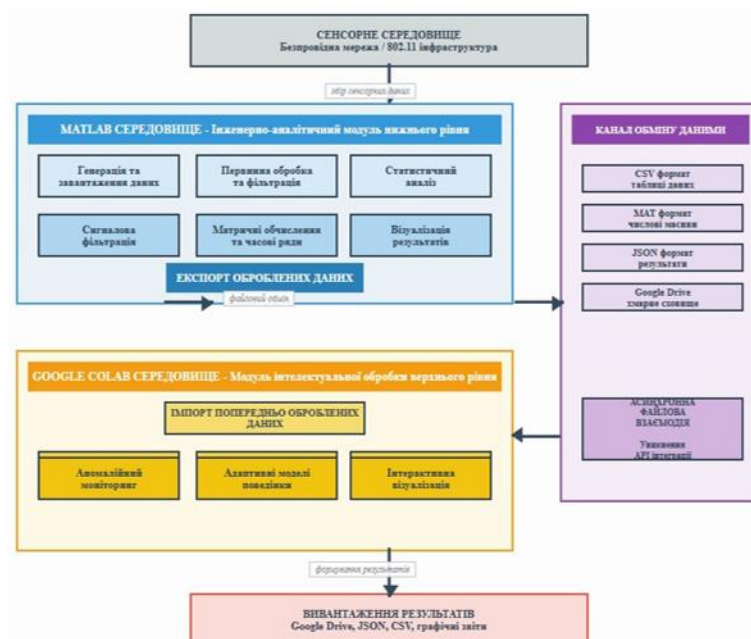
6

Структурна схема інформаційної системи



7

Архітектурна схема інформаційної системи обробки даних безпроводної мережі



8

Реалізація в Matlab та Google Colab

```

1  %%% --- 1. СТВОРЕННЯ СИНТЕТИЧНИХ ДАНИХ ---
2  N = 1000;
3  timestamps = datetime('now') + seconds(0:N-1);
4  temperature = 22 + 2*sin(2*pi*(0:N-1)/200) + randn(N,1)*0.5;
5  pressure = 101325 + (0.5*(0:N-1))' + randn(N,1)*20;
6  humidity = 60 + 5*sin(2*pi*(0:N-1)/150) + randn(N,1)*2;
7
8  sensorsdata = table(timestamps, temperature, pressure, humidity, ...
9  'VariableNames', {'Timestamps', 'Temperature', 'Pressure', 'Humidity'});
10
11 save('sensorsdata.mat', 'sensorsdata');
12 writetable(sensorsdata, 'sensorsdata.csv');
13 disp('✓ Синтетичний датасет створено.')
14 % Завантаження даних
15 load sensorsdata
16
17 % Огляд структури
18 head(sensorsdata)
19 time = sensorsdata.Timestamps;
20 temperature = sensorsdata.Temperature;
21 pressure = sensorsdata.Pressure;
22 humidity = sensorsdata.Humidity;
23
24 %%% --- ВІЗУАЛІЗАЦІЯ ПОЧАТКОВИХ ДАНИХ ---
25 figure('Name','Raw Sensor Data','NumberTitle','off')
26 subplot(3,1,1)
27 plot(time, temperature), title('Температура'), ylabel('°C'), grid on
28 subplot(3,1,2)
29 plot(time, pressure), title('Тиск'), ylabel('Pa'), grid on
30 subplot(3,1,3)

```

```

[ ] # Importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.fft import fft
from sklearn.preprocessing import StandardScaler
from google.colab import files
import io
import statsmodels.api as sm

```

```

[ ] # Upload CSV file
uploaded = files.upload()
df = pd.read_csv(io.BytesIO(list(uploaded.values())[0]))
df['Timestamps'] = pd.to_datetime(df['Timestamps'])
df.set_index('Timestamps', inplace=True)
df.head()

```

Orna sensorsdata.csv
sensorsdata.csv(application/vnd.ms-excel) - 71704 bytes, last modified: n/a - 100% done
Saving sensorsdata.csv to sensorsdata.csv

Timestamps	Temperature	Pressure	Humidity
2025-07-11 08:29:32	22.435299	101351.112462	59.144514
2025-07-11 08:29:33	22.228202	101345.179391	59.050632
2025-07-11 08:29:34	21.451633	101300.972273	62.270326
2025-07-11 08:29:35	22.962153	101322.904924	60.637687
2025-07-11 08:29:36	21.942348	101312.131871	59.564862

Подальші дії: [Переглянути рекомендовані графіки](#) [New interactive sheet](#)

```

[ ] # Smoothing
df['Temperature_smooth'] = df['Temperature'].rolling(10).mean()
df['Pressure_smooth'] = df['Pressure'].rolling(10).mean()
df['Humidity_smooth'] = df['Humidity'].rolling(10).mean()

```

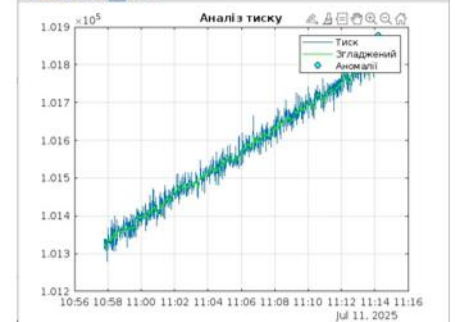
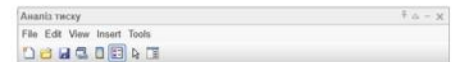
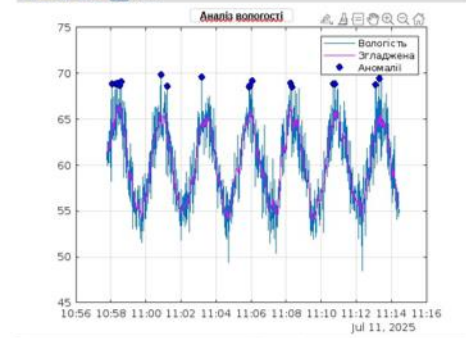
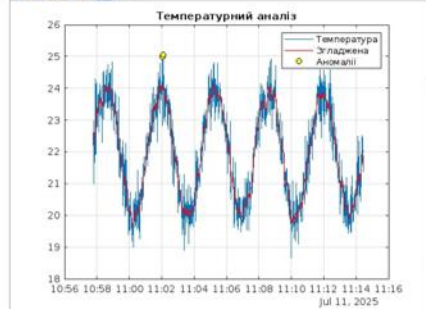
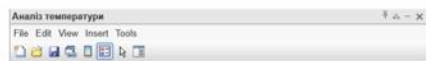
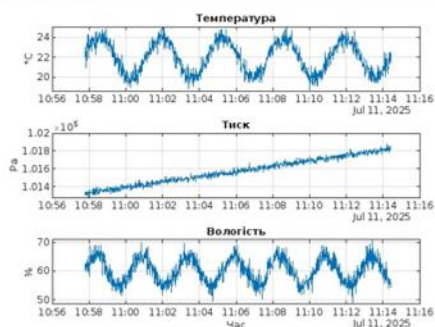
```

[ ] # Anomaly detection
def detect_anomalies(series):

```

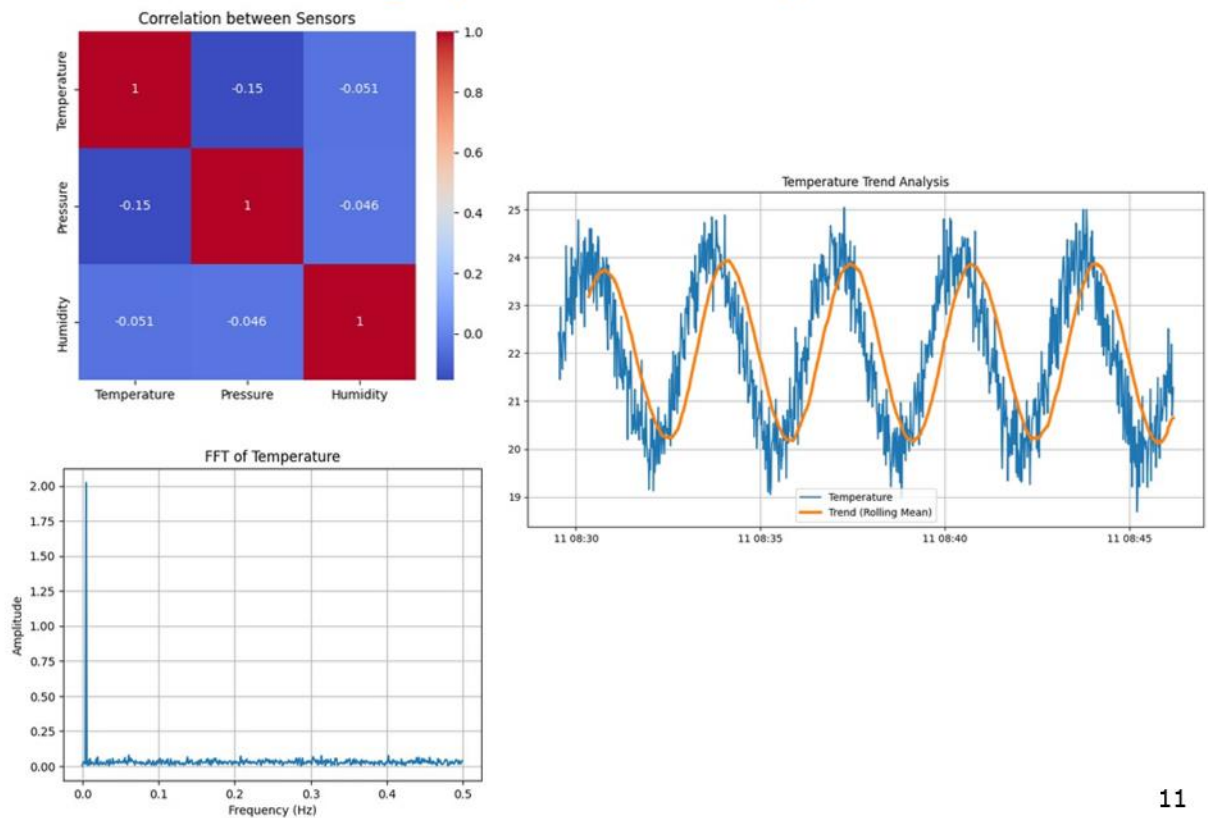
9

Аналіз результатів в Matlab



10

Аналіз результатів в Google Colab



11

Висновки

У ході виконання бакалаврської кваліфікаційної роботи було розроблено та реалізовано інформаційну систему обробки даних з безпроводної сенсорної мережі, яка функціонує на базі двох комплементарних середовищ: Matlab та Google Colab. Архітектура системи побудована з урахуванням функціонального розмежування: Matlab відповідає за моделювання даних, первинну обробку та базову візуалізацію, тоді як Google Colab реалізує розширену інтелектуальну аналітику, включаючи кореляційний аналіз, виявлення аномалій, спектральну декомпозицію та трендову оцінку.

У Matlab було синтезовано набір сенсорних даних, що включає змінні температури, тиску та вологості, з урахуванням імовірнісної природи вимірювань та шумових складових. Проведено згладжування сигналів, статистичну обробку та візуалізацію з метою ідентифікації динаміки змін. Результати демонструють наявність чітких періодичних компонент у температурі та вологості, а також висхідний тренд у тиску.

У Google Colab реалізовано обробку імпортованого CSV-файлу з використанням бібліотек Python. Здійснено виявлення аномалій за допомогою стандартного статистичного критерію, побудовано теплову карту кореляційної матриці, яка підтвердила слабкий лінійний зв'язок між сенсорними параметрами. Проведено швидке перетворення Фур'є температурного сигналу, результати якого засвідчили наявність домінуючої низькочастотної компоненти, що відповідає періодичній природі даних. Окрім того, реалізовано трендовий аналіз із використанням ковзного середнього, що дозволив візуалізувати основну динаміку коливань температури без шумових викривлень. Отримані результати підтверджують працездатність розробленої інформаційної системи, її здатність адаптуватися до різних джерел даних, виявляти ключові закономірності та забезпечувати основу для подальшого розширення з метою прогнозування, автоматичного моніторингу або інтеграції в складніші IoT-інфраструктури. Розділення обробки між Matlab та Colab довело свою ефективність і дозволило поєднати інженерну точність із гнучкістю машинного аналізу.

12

ДОДАТОК Б

Програмний код

Б.1 Лістинг файлу Gotra.m

```

%% --- 1. СТВОРЕННЯ СИНТЕТИЧНИХ ДАНИХ ---
N = 1000;
timestamps = datetime('now') + seconds(0:N-1)';
temperature = 22 + 2*sin(2*pi*(0:N-1)/200)' + randn(N,1)*0.5;
pressure = 101325 + (0.5*(0:N-1))' + randn(N,1)*20;
humidity = 60 + 5*sin(2*pi*(0:N-1)/150)' + randn(N,1)*2;

sensorsdata = table(timestamps, temperature, pressure, humidity,
    ...
    'VariableNames', {'Timestamps', 'Temperature', 'Pressure',
    'Humidity'});

save('sensorsdata.mat', 'sensorsdata');
writetable(sensorsdata, 'sensorsdata.csv');
disp('✓ Синтетичний датасет створено.')
% Завантаження даних
load sensorsdata

% Огляд структури
head(sensorsdata)
time = sensorsdata.Timestamps;
temperature = sensorsdata.Temperature;
pressure = sensorsdata.Pressure;
humidity = sensorsdata.Humidity;

%% --- ВІЗУАЛІЗАЦІЯ ПОЧАТКОВИХ ДАНИХ ---
figure('Name', 'Raw Sensor Data', 'NumberTitle', 'off')
subplot(3,1,1)
plot(time, temperature), title('Температура'), ylabel('°C'),
grid on
subplot(3,1,2)
plot(time, pressure), title('Тиск'), ylabel('Pa'), grid on
subplot(3,1,3)
plot(time, humidity), title('Вологість'), ylabel('%'), grid on
xlabel('Час')

%% --- ЗГЛАДЖУВАННЯ ТА ФІЛЬТРАЦІЯ ---
window = 10; % розмір ковзного вікна

temperature_smooth = movmean(temperature, window);
pressure_smooth = movmean(pressure, window);
humidity_smooth = movmean(humidity, window);

```

```

%% --- СТАТИСТИКА ---
fprintf('\nСЕРЕДНІ ЗНАЧЕННЯ:\n');
fprintf('Температура: %.2f °C | Тиск: %.2f Pa | Вологість: %.2f
%%\n', ...
    mean(temperature), mean(pressure), mean(humidity));

fprintf('\nСТАНДАРТНЕ ВІДХИЛЕННЯ:\n');
fprintf('Темп: %.2f | Тиск: %.2f | Вологість: %.2f\n', ...
    std(temperature), std(pressure), std(humidity));

%% --- ВИЯВЛЕННЯ АНОМАЛІЙ ---
t_thresh = mean(temperature) + 2*std(temperature);
p_thresh = mean(pressure) + 2*std(pressure);
h_thresh = mean(humidity) + 2*std(humidity);

t_anom = temperature > t_thresh;
p_anom = pressure > p_thresh;
h_anom = humidity > h_thresh;

%% --- ГРАФІКИ З АНАЛІЗОМ ---
figure('Name','Аналіз температури','NumberTitle','off')
plot(time, temperature), hold on
plot(time, temperature_smooth, 'r', 'LineWidth', 1.2)
plot(time(t_anom), temperature(t_anom), 'ko',
'MarkerFaceColor','y')
legend('Температура','Згладжена','Аномалії')
title('Температурний аналіз'), grid on

figure('Name','Аналіз тиску','NumberTitle','off')
plot(time, pressure), hold on
plot(time, pressure_smooth, 'g', 'LineWidth', 1.2)
plot(time(p_anom), pressure(p_anom), 'ko',
'MarkerFaceColor','c')
legend('Тиск','Згладжений','Аномалії')
title('Аналіз тиску'), grid on

figure('Name','Аналіз вологості','NumberTitle','off')
plot(time, humidity), hold on
plot(time, humidity_smooth, 'm', 'LineWidth', 1.2)
plot(time(h_anom), humidity(h_anom), 'ko',
'MarkerFaceColor','b')
legend('Вологість','Згладжена','Аномалії')
title('Аналіз вологості'), grid on

%% --- СПЕКТРАЛЬНИЙ АНАЛІЗ ---
fs = 1; % частота дискретизації (1 Гц - якщо 1 точка/с)
L = length(temperature);
f = fs*(0:(L/2))/L;

Y = fft(temperature - mean(temperature));
P2 = abs(Y/L);
P1 = P2(1:L/2+1);

```

```

P1(2:end-1) = 2*P1(2:end-1);

figure('Name','Спектральний аналіз','NumberTitle','off')
plot(f, P1), title('FFT температури')
xlabel('Частота (Гц)'), ylabel('|A(f)|'), grid on

%% --- ЗБЕРЕЖЕННЯ ---
saveas(gcf, 'fft_temp.png')
writetable(sensorsdata, 'exported_sensorsdata.csv')

disp('✓ Обробка завершена. Результати збережено.')
writetable(sensorsdata, 'sensorsdata.csv');

```

Б.2 Лістинг коду в Google Colab

```

# Importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.fft import fft
from sklearn.preprocessing import StandardScaler
from google.colab import files
import io
import statsmodels.api as sm
# Upload CSV file
uploaded = files.upload()
df = pd.read_csv(io.BytesIO(list(uploaded.values())[0]))
df['Timestamps'] = pd.to_datetime(df['Timestamps'])
df.set_index('Timestamps', inplace=True)
df.head()
# Smoothing
df['Temperature_smooth'] = df['Temperature'].rolling(10).mean()
df['Pressure_smooth'] = df['Pressure'].rolling(10).mean()
df['Humidity_smooth'] = df['Humidity'].rolling(10).mean()
# Anomaly detection
def detect_anomalies(series):
    mean = series.mean()
    std = series.std()
    return series > (mean + 2 * std)

df['Temp_anom'] = detect_anomalies(df['Temperature'])
df['Press_anom'] = detect_anomalies(df['Pressure'])
df['Hum_anom'] = detect_anomalies(df['Humidity'])
# Correlation matrix
plt.figure(figsize=(6, 5))
sns.heatmap(df[['Temperature', 'Pressure',
'Humidity']].corr(), annot=True, cmap='coolwarm')
plt.title('Correlation between Sensors')
plt.show()

```

```

# Summary table of anomalies
anomaly_summary = pd.DataFrame({
    'Total': [df['Temperature'].count(),
df['Pressure'].count(), df['Humidity'].count()],
    'Anomalies': [df['Temp_anom'].sum(),
df['Press_anom'].sum(), df['Hum_anom'].sum()]
}, index=['Temperature', 'Pressure', 'Humidity'])
anomaly_summary['Anomaly %'] =
(anomaly_summary['Anomalies'] /
anomaly_summary['Total']) * 100
anomaly_summary
# Trend analysis using rolling mean
plt.figure(figsize=(12,6))
plt.plot(df.index, df['Temperature'],
label='Temperature')
plt.plot(df.index,
df['Temperature'].rolling(50).mean(), label='Trend
(Rolling Mean)', linewidth=3)
plt.title('Temperature Trend Analysis')
plt.grid(), plt.legend()
plt.show()
# FFT analysis
N = len(df['Temperature'])
T = 1.0
yf = fft(df['Temperature'].fillna(0) -
df['Temperature'].mean())
xf = np.fft.fftfreq(N, T)[:N//2]
plt.plot(xf, 2.0/N * np.abs(yf[0:N//2]))
plt.title('FFT of Temperature'), plt.grid()
plt.xlabel('Frequency (Hz)'), plt.ylabel('Amplitude')
plt.show()

```