

Đề Tài

Gas sensor array low-concentration

Giảng viên: Lê Đình Long 

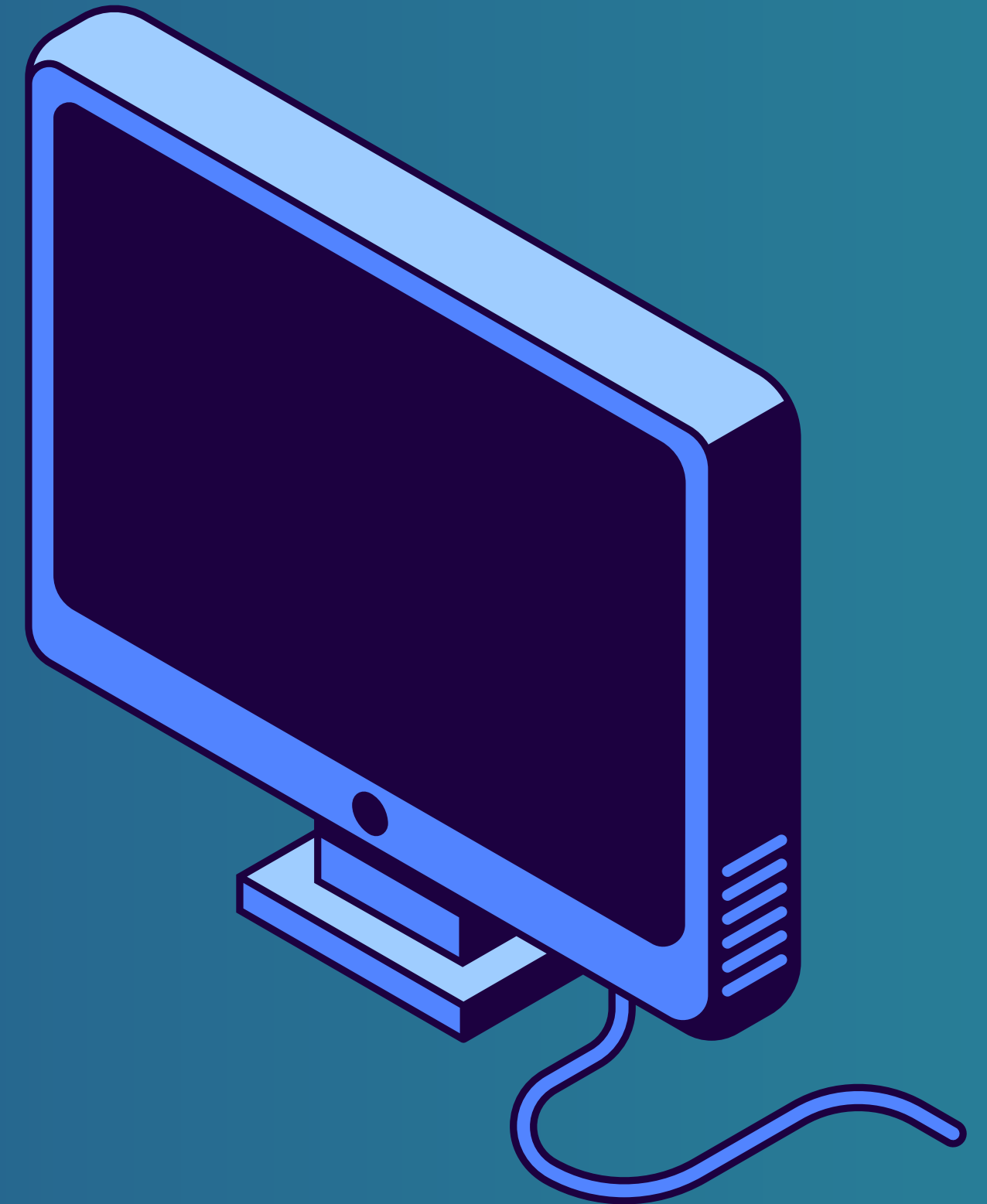
Nhóm 5

Nguyễn Hùng Anh – 21064051

Nguyễn Hạnh Bảo Ân – 21075071

Lê Phú Hào – 21073141

Dương Ngọc Anh – 21090261



Tầm quan trọng

- Phát hiện khí ở nồng độ thấp rất cần thiết trong y tế, công nghiệp, và bảo vệ môi trường.
- Ví dụ: Phát hiện khí CO trong nhà ở, khí độc trong nhà máy hóa chất, hoặc khí gây hiệu ứng nhà kính

Mục tiêu

- Sử dụng mô hình FE-ELM để phân loại dữ liệu từ mảng cảm biến khí (GSA) trong điều kiện nồng độ thấp, nhằm phát hiện và phân biệt các loại khí như ethanol, acetone, toluene ở mức ppb (phần tỷ)

Tổng quan về dữ liệu

Cấu trúc dữ liệu:

- 89 mẫu (dòng), mỗi mẫu đại diện cho một lần đo.
- 9002 cột:
 - Cột 1: Nhãn khí (Gas) : Ethanol, Acetone, Toluene, Ethyl Acetate, Isopropanol, n-Hexane
 - Cột 2: Nồng độ (Concentration) - ví dụ: 50ppb, 100ppb, 200ppb.
 - Cột 3-9002: Phản ứng của 10 cảm biến, mỗi cảm biến có 900 điểm dữ liệu.
- "Tổng cộng có 6 loại khí, mỗi loại đo ở 3 nồng độ khác nhau, lặp lại 5 lần, tạo thành 90 mẫu, nhưng thực tế tệp chỉ có 89 mẫu."



Xem 5 dòng dữ liệu đầu tiên và kiểm tra dữ liệu

```
# Xem trước 5 dòng đầu tiên
df.head()
```

	ethanol	100ppb	0.3565	0.3345	0.3575	0.333	0.3565.1	0.3345.1	0.3565.2	0.3355	...	3.93.74	3.929.105	3.929.106	3.929.107	3.9315.53	3.935.12	3.9275.108	3.93.75	3.9325.24	3.9315.54
0	ethanol	100ppb	0.3525	0.3305	0.3525	0.3320	0.3550	0.3305	0.3525	0.3280	...	3.9190	3.9190	3.9180	3.9180	3.9180	3.9180	3.9190	3.918	3.9180	3.9205
1	ethanol	100ppb	0.3550	0.3345	0.3540	0.3320	0.3550	0.3330	0.3550	0.3320	...	3.8815	3.8840	3.8800	3.8800	3.8840	3.8840	3.8815	3.884	3.8825	3.8815
2	ethanol	100ppb	0.3415	0.3220	0.3430	0.3210	0.3430	0.3235	0.3415	0.3210	...	3.9460	3.9485	3.9410	3.9470	3.9460	3.9425	3.9460	3.946	3.9460	3.9460
3	ethanol	100ppb	0.3500	0.3245	0.3465	0.3320	0.3455	0.3245	0.3490	0.3235	...	3.9375	3.9400	3.9385	3.9375	3.9375	3.9365	3.9400	3.940	3.9365	3.9365
4	ethanol	200ppb	0.3380	0.3220	0.3365	0.3345	0.3380	0.3365	0.3380	0.3210	...	3.8730	3.8705	3.8715	3.8715	3.8730	3.8705	3.8715	3.874	3.8680	3.8730

5 rows × 9002 columns

Kiểm tra kích thước , tên cột và xác nhận không có dữ liệu thiếu

- Số lượng mẫu (dòng): 89
- Số lượng đặc trưng (cột): 9002
- Tên 10 cột đầu tiên: ['ethanol', '100ppb', '0.3565', '0.3345', '0.3575', '0.333', '0.3565.1', '0.3345.1', '0.3565.2', '0.3355']
- Số lượng giá trị bị thiếu: 0



Đổi tên cột

- Cột 1: Gas (nhãn khí).
- Cột 2: Concentration (nồng độ).
- Cột 3-9002: Đặt tên là các chỉ số từ 0 đến 8999 cho dễ xử lý.

Mã hóa nhãn

- Chuyển nhãn khí (ethanol, acetone, v.v.) thành số (0, 1, 2, ...).
- Chuyển nồng độ (50ppb, 100ppb, ...) thành số (0, 1, 2, ...).

```
# Đổi tên cột: Cột 1 là 'Gas', Cột 2 là 'Concentration', Còn lại là các chỉ số từ 0 đến 8999
df.columns = ['Gas', 'Concentration'] + list(range(9000))
```

```
# Kiểm tra giá trị thiếu
missing_values = df.isnull().sum().sum()
```

```
# Chuyển đổi nhãn khí và nồng độ thành dạng số
gas_labels = {gas: i for i, gas in enumerate(df['Gas'].unique())}
concentration_labels = {conc: i for i, conc in enumerate(df['Concentration'].unique())}
```

```
df['Gas'] = df['Gas'].map(gas_labels)
df['Concentration'] = df['Concentration'].map(concentration_labels)
```

Trích xuất 4 đặc trưng (trung bình, độ lệch chuẩn, độ lệch, độ nhọn) từ 900 điểm của mỗi cảm biến, tạo thành 40 đặc trưng/mẫu

```
def extract_features(sensor_data):  
    features = []  
    for i in range(10): # 10 cảm biến  
        start = i * 900  
        end = (i + 1) * 900  
        segment = sensor_data[:, start:end]  
        features.append(np.mean(segment, axis=1))  
        features.append(np.std(segment, axis=1))  
        features.append(skew(segment, axis=1))  
        features.append(kurtosis(segment, axis=1))  
    return np.column_stack(features)  
  
# Áp dụng trích xuất đặc trưng  
X = extract_features(sensor_data)  
y = df['Gas'].values # Nhãn khí  
  
# Kiểm tra kích thước dữ liệu sau trích xuất  
X.shape, y.shape
```





Huấn luyện mô hình FE-ELM

```
# Chia dữ liệu (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

# Chuẩn hóa dữ liệu (Standardization)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Huấn luyện mô hình FE-ELM (MLP với một hidden layer)
model = MLPClassifier(hidden_layer_sizes=(50,), activation='relu', solver='adam', max_iter=1000, random_state=42)
model.fit(X_train, y_train)

# Dự đoán trên tập kiểm tra
y_pred = model.predict(X_test)
```



Đánh giá mô hình

- **Độ chính xác: 83.3% trên tập test.**
- **Báo cáo chi tiết:**
 - Nhãn 0 (ethanol): F1-score = 0.80.
 - Nhãn 1: F1-score = 1.00 (phân loại hoàn hảo).
 - Nhãn 2: F1-score = 0.86.
 - Nhãn 3: F1-score = 0.75.
 - Nhãn 4: F1-score = 0.80.
 - Nhãn 5: F1-score = 0.80.
- **Nhận xét:**
 - Mô hình đạt hiệu suất tốt, đặc biệt với nhãn 1 (độ chính xác 100%).
 - Một số nhãn (như nhãn 3) có hiệu suất thấp hơn, có thể do dữ liệu không đủ đa dạng hoặc đặc trưng chưa tối ưu.

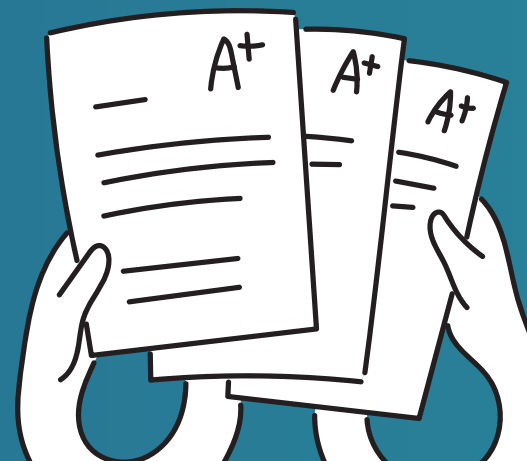
```
[ ] # Đánh giá mô hình
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)

print(f"Độ chính xác tổng thể: {accuracy:.1%}")
print(report)
```

```
⇒ Độ chính xác tổng thể: 83.3%
              precision    recall  f1-score   support

     0         1.00        0.67        0.80         3
     1         1.00        1.00        1.00         3
     2         0.75        1.00        0.86         3
     3         0.60        1.00        0.75         3
     4         1.00        0.67        0.80         3
     5         1.00        0.67        0.80         3

 accuracy                   0.83         18
 macro avg                0.89         18
 weighted avg             0.89         18
```



Kết luận:

- Dự án đã thành công trong việc áp dụng FE-ELM để phân loại khí với nồng độ thấp, đạt độ chính xác 83.3%.
- Quy trình tiền xử lý và trích xuất đặc trưng hiệu quả, giảm số lượng đặc trưng từ 9000 xuống 40 mà vẫn giữ được thông tin quan trọng.

Hướng phát triển:

- Tăng số lượng mẫu dữ liệu để cải thiện độ chính xác.
- Thử nghiệm các phương pháp trích xuất đặc trưng khác (ví dụ: PCA, wavelet transform).
- Tinh chỉnh tham số mô hình hoặc thử các mô hình khác như Random Forest, SVM để so sánh.





THANK
YOU