Feature Selection : Menerapkan Recursive Feature Elimination with Cross-Validation (RFECV)

Anggota Kelompok

1. Cornelius Linux_122140079
2. Chandra Budi Wijaya_122140093
3. Muhammad Yusuf_122140193
4. Elma Nurul Fatika_122140069
5. Shafa Aulia_122140062
6. Harisya Miranti_122140049

Gambaran Umum Dataset dan Urgensi Feature Selection

Dataset **WADI** dan **SWAT** merupakan kumpulan data sensor industri berskala besar yang digunakan untuk mendeteksi kondisi abnormal atau serangan (attack) pada sistem kontrol industri. Keduanya berisi pembacaan multivariate time-series dari berbagai transmitter, alarm, flow meter, pressure indicator, motor valve status, hingga sensor turunan operasional. Karakteristik utamanya adalah:

- **WADI** mengandung lebih dari **120 variabel sensor**, mencakup tekanan, aliran, level, dan status aktuator.
- **SWAT** berisi **52 variabel sensor**, terdiri dari indikator proses inti dan sensor tambahan dari beberapa unit produksi.
- Kedua dataset memiliki target biner: *Normal (0)* dan *Attack (1)*.

Lingkungan industri seperti ini biasanya menghasilkan data berdimensi tinggi (high-dimensional), penuh korelasi antar sensor, dan sering kali mengandung sinyal redundan yang tidak berkontribusi pada prediksi. Masalah ini dapat menyebabkan:

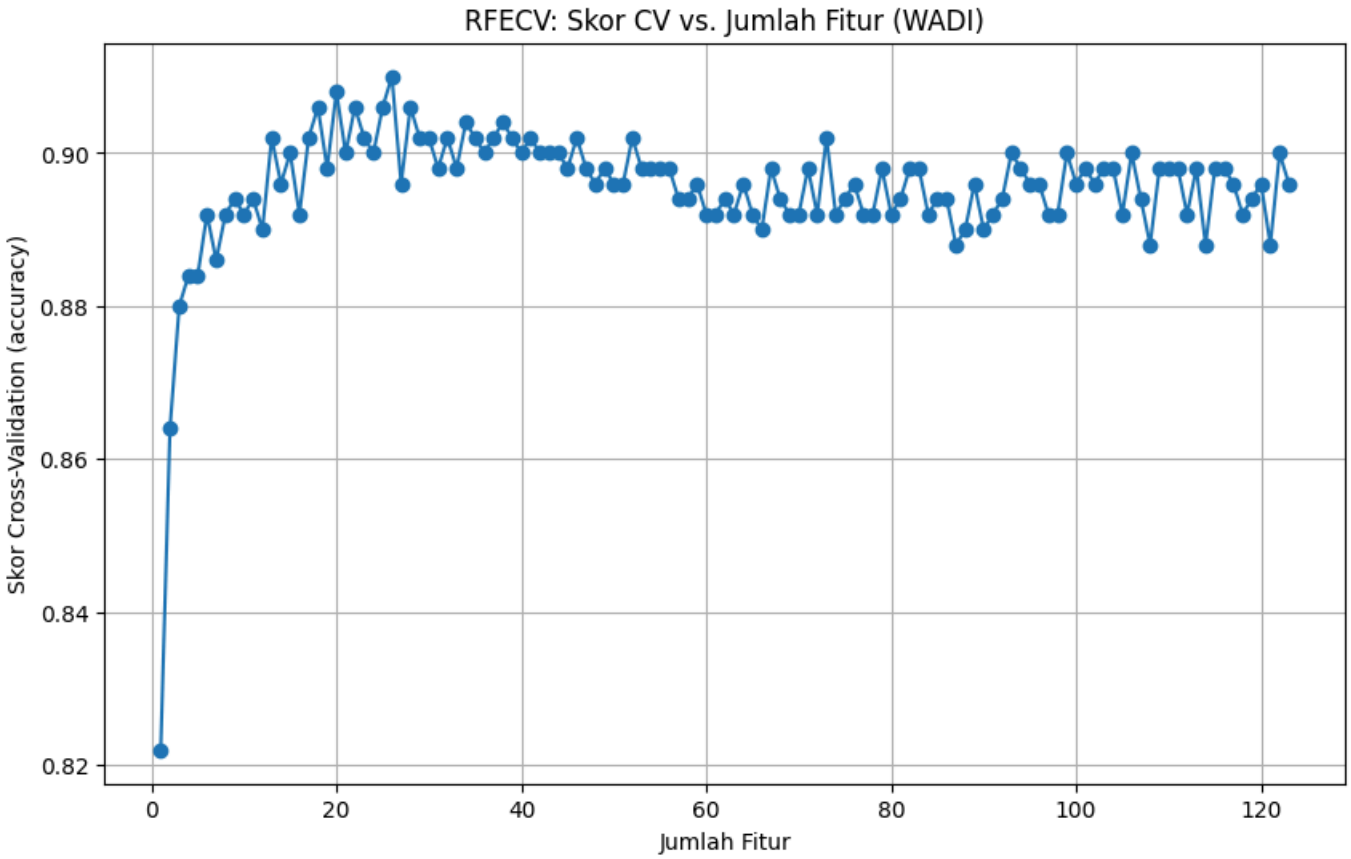
1. **Overfitting** model karena terlalu banyak fitur tidak relevan.
2. **Waktu komputasi meningkat** secara signifikan.
3. **Penurunan akurasi** akibat noise pada fitur yang tidak informatif.
4. **Kesulitan interpretasi** ketika fitur terlalu banyak.

Oleh sebab itu, proses *feature selection* menjadi penting untuk menyaring fitur yang benar-benar relevan. Dengan memilih subset fitur terbaik, model machine learning dapat bekerja lebih cepat, lebih stabil, dan memberikan keputusan yang lebih mudah dipahami. Pendekatan ini sejalan dengan penerapan teknik seleksi fitur pada penelitian Yadav et al. (2025) yang menunjukkan bahwa pemilihan fitur mampu meningkatkan performa model ensemble pada sistem klasifikasi.

In [4]:
```
# Menerapkan Recursive Feature Elimination with Cross-Validation (RFECV)
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_selection import RFECV
from sklearn.model_selection import StratifiedKFold

df = pd.read_csv('WADI.csv')
target_column = 'Attack'

df_clean = df.drop(columns=['Date', 'Time']).copy()
X = df_clean.drop(columns=[target_column])
y = df_clean[target_column]

X = X.fillna(0)

estimator = RandomForestClassifier(n_estimators=100, random_state=42, n_jobs=-1)
cv_strategy = StratifiedKFold(5)
rfecv = RFECV(
    estimator=estimator,
    step=1,
    cv=cv_strategy,
    scoring='accuracy',
    min_features_to_select=1
)

print("Memulai proses RFECV...")
rfecv.fit(X, y)

optimal_feature_count = rfecv.n_features_
selected_features_mask = rfecv.support_
selected_feature_names = X.columns[selected_features_mask].tolist()

X_selected = X[selected_feature_names]

print("\nProses RFECV Selesai.")
print(f"Jumlah Fitur Awal: {X.shape[1]}")
```

```
print(f"Jumlah Fitur Optimal: {optimal_feature_count}")
print(f"Fitur-Fitur yang Terpilih ({optimal_feature_count} fitur):")
for feature in selected_feature_names:
    print(f"- {feature}")

# Bagian kode untuk visualisasi
scores = rfecv.cv_results_['mean_test_score']
num_features = range(1, len(scores) + 1)
plt.figure(figsize=(10, 6))
plt.plot(num_features, scores, marker='o')
plt.xlabel("Jumlah Fitur")
plt.ylabel(f"Skor Cross-Validation ({rfecv.scoring})")
plt.title("RFECV: Skor CV vs. Jumlah Fitur (WADI)")
plt.grid(True)
plt.show()
```

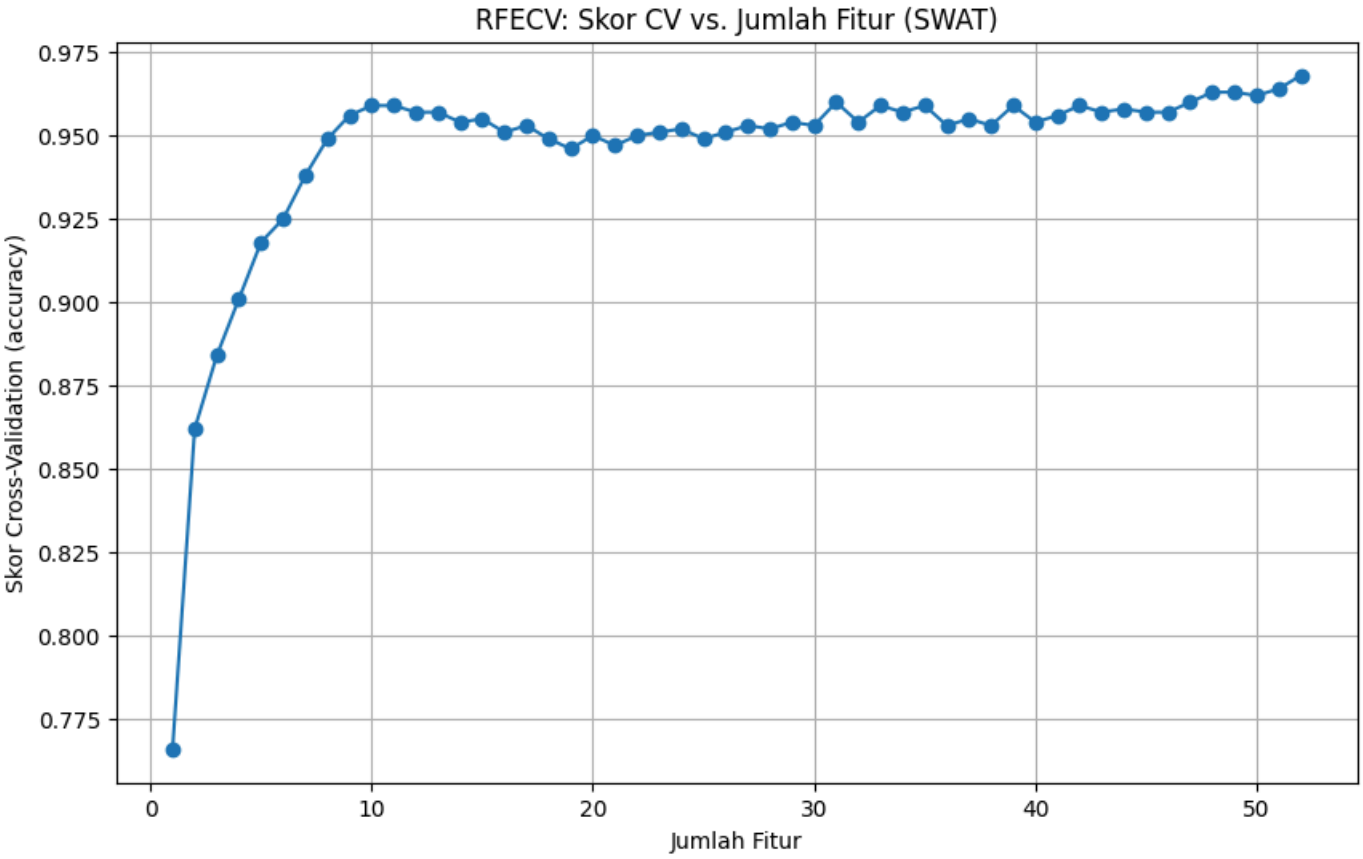
```
Memulai proses RFECV...

Proses RFECV Selesai.
Jumlah Fitur Awal: 123
Jumlah Fitur Optimal: 26
Fitur-Fitur yang Terpilih (26 fitur):
- 1_AIT_003_PV
- 1_LS_002_AL
- 1_P_002_STATUS
- 1_P_006_STATUS
- Sensor_30
- Sensor_33
- Sensor_37
- Sensor_38
- Sensor_47
- Sensor_52
- Sensor_53
- Sensor_59
- Sensor_67
- Sensor_68
- Sensor_72
- Sensor_74
- Sensor_75
- Sensor_79
- Sensor_89
- Sensor_90
- Sensor_92
- Sensor_97
- Sensor_107
- Sensor_114
- Sensor_117
- LEAK_DIFF_PRESSURE
```



In [ ]:

```python
# Menerapkan Recursive Feature Elimination with Cross-Validation (RFECV)
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_selection import RFECV
from sklearn.model_selection import StratifiedKFold

df = pd.read_csv('SWAT_Dataset.csv')
target_column = 'Normal/Attack'

X = df.drop(columns=[target_column])
y_raw = df[target_column]

# Mengubah kolom target (string) menjadi numerik biner: Normal=0, Attack=1
y = y_raw.apply(lambda x: 1 if x.strip() == 'Attack' else 0)

X = X.fillna(0)

estimator = RandomForestClassifier(n_estimators=100, random_state=42, n_jobs=-1)
cv_strategy = StratifiedKFold(5, shuffle=True, random_state=42)
rfecv = RFECV(
    estimator=estimator,
    step=1,
    cv=cv_strategy,
    scoring='accuracy',
    min_features_to_select=1
)

print("Memulai proses RFECV pada dataset SWAT...")
rfecv.fit(X, y)

optimal_feature_count = rfecv.n_features_
selected_features_mask = rfecv.support_
selected_feature_names = X.columns[selected_features_mask].tolist()

X_selected = X[selected_feature_names]

print("\nProses RFECV Selesai.")
print(f"Jumlah Fitur Awal: {X.shape[1]}")
print(f"Jumlah Fitur Optimal: {optimal_feature_count}")
print(f"Fitur-Fitur yang Terpilih ({optimal_feature_count} fitur):")
for feature in selected_feature_names:
    print(f"- {feature}")

# Bagian kode untuk visualisasi
scores = rfecv.cv_results_['mean_test_score']
num_features = range(1, len(scores) + 1)
plt.figure(figsize=(10, 6))
plt.plot(num_features, scores, marker='o')
plt.xlabel("Jumlah Fitur")
plt.ylabel(f"Skor Cross-Validation ({rfecv.scoring})")
plt.title("RFECV: Skor CV vs. Jumlah Fitur (SWAT)")
plt.grid(True)
plt.show()
```

```
Memulai proses RFECV pada dataset SWAT...

Proses RFECV Selesai.
Jumlah Fitur Awal: 52
Jumlah Fitur Optimal: 52
Fitur-Fitur yang Terpilih (52 fitur):
- P206
- DPIT301
- FIT301
- LIT301
- MV301
- MV302
- MV303
- MV304
- P301
- P302
- AIT401
- AIT402
- FIT401
- LIT401
- P401
- P402
- P403
- P404
- UV401
- AIT501
- AIT502
- AIT503
- AIT504
- FIT501
- FIT502
- FIT503
- FIT504
- P501
- P502
- PIT501
- PIT502
- PIT503
- FIT601
- P601
- P602
- P603
- Sensor_0
- Sensor_1
- Sensor_2
- Sensor_3
- Sensor_4
- Sensor_5
- Sensor_6
- Sensor_7
- Sensor_8
- Sensor_9
- Sensor_10
- Sensor_11
- Sensor_12
- Sensor_13
- Sensor_14
- Sensor_15
```



Hasil Penerapan RFECV

Recursive Feature Elimination with Cross-Validation (RFECV) adalah metode seleksi fitur bertahap yang bekerja dengan menghapus fitur paling lemah secara berulang. Pada setiap iterasi, model dievaluasi menggunakan *cross-validation* untuk memastikan bahwa fitur yang tersisa benar-benar meningkatkan performa. Tujuan akhirnya adalah menemukan jumlah fitur optimal yang memberikan skor prediksi terbaik.

Metode ini dipilih karena mampu:

- Menangani dataset berdimensi tinggi,
- Mengurangi fitur redundan,
- Meningkatkan generalisasi model,
- Menghasilkan subset fitur yang lebih stabil dan mudah ditafsirkan.

Hasil RFECV pada Dataset WADI

- Jumlah fitur awal: **123**
- Jumlah fitur optimal: **26**
- Fitur yang terpilih mencakup sensor status pompa, level alarm, dan sejumlah sensor derivatif (Sensor_30, Sensor_38, Sensor_97, dll.)
- Grafik menunjukkan bahwa akurasi CV meningkat cepat hingga sekitar 20–30 fitur, kemudian cenderung datar dan sedikit menurun. Hal ini menunjukkan bahwa sebagian besar fitur tambahan tidak berkontribusi signifikan pada prediksi anomali.

Hasil ini mencerminkan bahwa dataset WADI memang kaya fitur redundan serta sensor yang tidak memiliki pengaruh kuat terhadap deteksi kondisi attack.

Hasil RFECV pada Dataset SWAT

- Jumlah fitur awal: **52**
- Jumlah fitur optimal: **52**
- Semua fitur sensor dipertahankan oleh RFECV.
- Kurva skor CV menunjukkan stabilitas tinggi sejak jumlah fitur sekitar 10, dengan sedikit peningkatan ketika seluruh fitur dipertahankan.

Hal ini menunjukkan bahwa variabel pada SWAT memiliki kontribusi kuat dan saling melengkapi, sehingga menghapus fitur justru menurunkan performa model.

Interpretasi Umum

- **WADI**: Banyak fitur redundan → pemilihan fitur sangat membantu, mengurangi jumlah fitur hingga ~80% tanpa kehilangan akurasi.

- **SWAT**: Fitur-fitur bersifat komplementer → seluruh fitur tetap diperlukan untuk akurasi terbaik.

Hasil ini memperkuat temuan dalam penelitian Yadav et al. (2025), di mana teknik seleksi fitur membantu meningkatkan performa model terutama pada dataset dengan jumlah fitur besar dan struktur sensor kompleks.

Referensi

P. S. Yadav, R. S. Rao, A. Mishra, and M. Gupta (2024), "Ensemble methods with feature selection and data balancing for improved code smells classification performance," Engineering Applications of Artificial Intelligence, vol. 139, 2025, Art. no. 109527.