SSB IsolationForest Modeli - 1

May 8, 2025

[6]:
```
import pandas as pd

# Dosya yolu
file_path = "C:\\Users\\Lenovo\\Desktop\\OneDrive_2025-05-07\\TON_IoT␣
 ↪datasets\\Processed_datasets\\Processed_IoT_dataset\\IoT_Fridge.csv"

# Veriyi oku
df = pd.read_csv(file_path)

# İlk 5 satırı görüntüle
df.head()
```

[6]:

	date	time	fridge_temperature	temp_condition	label	type
0	31-Mar-19	12:36:52	13.10	high	0	normal
1	31-Mar-19	12:36:53	8.65	high	0	normal
2	31-Mar-19	12:36:54	2.00	low	0	normal
3	31-Mar-19	12:36:55	4.80	low	0	normal
4	31-Mar-19	12:36:56	10.70	high	0	normal

[7]:
```
# Sütun adlarını ve veri tiplerini kontrol et
print(df.dtypes)

# Eksik değer kontrolü
print(df.isnull().sum())

# Eksik varsa satırları sil (ya da dilersen dolgu da yapılabilir)
df.dropna(inplace=True)
```

```
date                 object
time                 object
fridge_temperature  float64
temp_condition       object
label                 int64
type                 object
dtype: object
date                 0
time                 0
fridge_temperature   0
```

```
temp_condition        0
label                 0
type                  0
dtype: int64
```

[8]:
```
# Gereksiz sütunları çıkar (tarih, saat, saldırı tipi vs.)
df.drop(['date', 'time', 'type'], axis=1, inplace=True)
```

[9]:
```
# 'label' sütunu zaten 0 (normal) ve 1 (attack) şeklinde → kontrol amaçlı göster
print(df['label'].value_counts())
```

```
0    500827
1     86249
Name: label, dtype: int64
```

[10]:
```
# Kategorik verileri dönüştür (temp_condition)
# temp_condition (low, high) gibi kategorik verileri sayısal forma çevir
df = pd.get_dummies(df, columns=['temp_condition'], drop_first=True)
```

[11]:
```
# Temizlenmiş veri setini gör
print(df.head())

# Giriş ve hedefi ayır
X = df.drop('label', axis=1)
y = df['label']
```

```
   fridge_temperature  label  temp_condition_high   temp_condition_high    \
0               13.10      0                     1                     0
1                8.65      0                     1                     0
2                2.00      0                     0                     0
3                4.80      0                     0                     0
4               10.70      0                     1                     0

   temp_condition_low  temp_condition_low   temp_condition_low
0                   0                   0                   0
1                   0                   0                   0
2                   0                   1                   0
3                   0                   1                   0
4                   0                   0                   0
```

[16]:
```
# Giriş ve hedefi ayır
X = df_clean.drop('label', axis=1)
y = df_clean['label']

# Sayısal veriyi ölçekle
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

```python
# Isolation Forest modeli
from sklearn.ensemble import IsolationForest
from sklearn.metrics import classification_report, confusion_matrix

model = IsolationForest(n_estimators=100, contamination='auto', random_state=42)
model.fit(X_scaled)

# Tahmin (-1 anomali, 1 normal → 1 = saldırı, 0 = normal)
y_pred = model.predict(X_scaled)
y_pred = [1 if val == -1 else 0 for val in y_pred]

# Performans çıktıları
print(confusion_matrix(y, y_pred))
print(classification_report(y, y_pred))
```

```
[[342842 157985]
 [ 48997  37252]]
              precision    recall  f1-score   support

           0       0.87      0.68      0.77    500827
           1       0.19      0.43      0.26     86249

    accuracy                           0.65    587076
   macro avg       0.53      0.56      0.52    587076
weighted avg       0.77      0.65      0.69    587076
```

1 Sonuçları Kısaca Yorumlayalım:

0 (Normal) 1 (Anormal / Saldırı)

Precision 0.87 (çok iyi) 0.19 (düşük) Recall 0.68 (ortalama) 0.43 (fena değil, iyileştirilebilir) F1-Score 0.77 0.26 Accuracy 0.65 (yani %65 doğruluk)

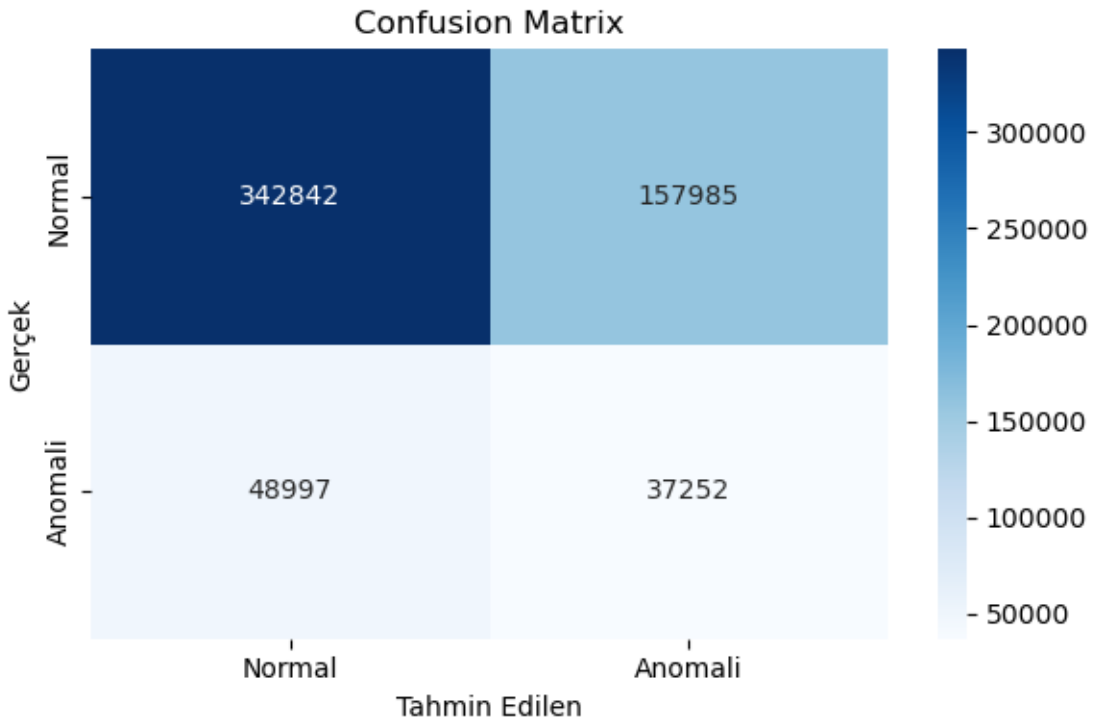
2 Ne Anlama Geliyor?

Model, normal veriyi iyi öğrenmiş, ama anomalileri ayırmakta zorlanıyor. Bu, genelde etiketsiz veri ile çalışıldığında (unsupervised) normal. Çünkü IsolationForest modeli saldırıyı görmeden sadece uç davranışları ayırt eder.

[19]:
```python
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# Gerçek ve tahmin edilen değerlerle Confusion Matrix oluştur
cm = confusion_matrix(y, y_pred)
```

```
# Grafik çizimi
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=['Normal', 'Anomali'],
            yticklabels=['Normal', 'Anomali'])
plt.title("Confusion Matrix")
plt.xlabel("Tahmin Edilen")
plt.ylabel("Gerçek")
plt.tight_layout()
plt.show()
```



3 Açıklamalar

annot=True: Hücre içlerine sayıları yazdırır. fmt='d': Sayılar tam sayı (integer) formatında gösterilir. cmap='Blues': Renk tonunu belirler, istersen Reds, Greens, Purples da yazabilirsin.

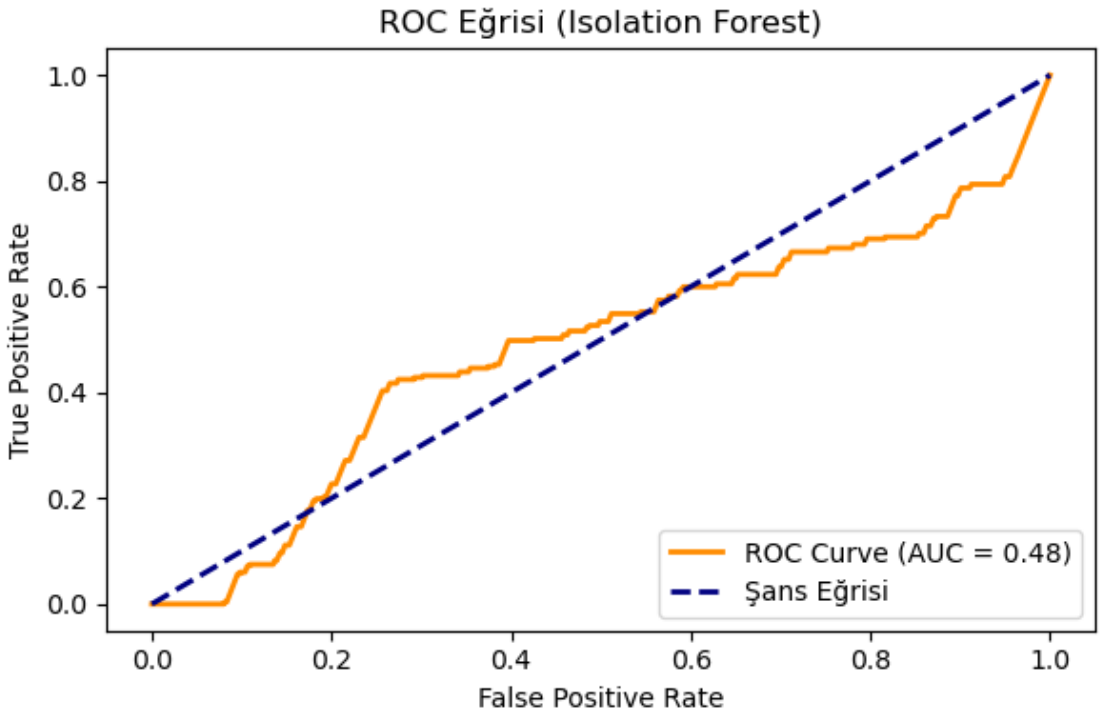
[20]:
```
from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt

# Skorları al (yüksek skor = daha normal → ters çevirmek gerek)
scores = model.decision_function(X_scaled)

# ROC eğrisi için ters çevirerek anomali skoru yapıyoruz
```

```
fpr, tpr, _ = roc_curve(y, -scores)
roc_auc = auc(fpr, tpr)

# ROC grafiği
plt.figure(figsize=(6, 4))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC Curve (AUC = {roc_auc:.
↪2f})')
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--', label='Şans␣
↪Eğrisi')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Eğrisi (Isolation Forest)')
plt.legend(loc="lower right")
plt.tight_layout()
plt.show()
```



Bu kod, Isolation Forest algoritmasının anomali tespit performansını görsel olarak değerlendirmek için ROC (Receiver Operating Characteristic) eğrisini üretir. Modelin decision_function() çıktısı kullanılarak elde edilen skorlar, normal verilerde yüksek, anomalilerde düşük olduğu için skorlar negatifleştirilerek ROC eğrisi çizilmiştir. Grafik Yorumu: -True Positive Rate (TPR): Gerçek anomali verilerinin doğru şekilde tespit edilme oranı. -False Positive Rate (FPR): Gerçek normal verilerin yanlışlıkla anomali olarak algılanma oranı. -Eğri, (0,1) noktasına ne kadar yakınsa model o kadar iyi çalışıyor demektir. -AUC (Area Under Curve) değeri bu eğrinin altındaki alanı temsil eder: –AUC 1 → Mükemmel model –AUC 0.5 → Tahmin gücü yok (şansa bağlı)

[ ]: