SVM_default

July 10, 2024

[1]:
```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix,␣
  ↪ConfusionMatrixDisplay, accuracy_score, precision_score, recall_score,␣
  ↪f1_score, classification_report, roc_curve, auc, RocCurveDisplay
import matplotlib.pyplot as plt
from sklearn.utils import resample
import matplotlib.colors
from sklearn.inspection import DecisionBoundaryDisplay
```

[2]:
```
%%time
df_main = pd.read_csv('../Dataset/IDS 2018 Intrusion CSVs (CSE-CIC-IDS2018)/
  ↪MainDataset/dataset_main.csv')
```

```
CPU times: total: 1min 9s
Wall time: 1min 51s
```

[3]:
```
len(df_main)
```

[3]:
```
11916113
```

[4]:
```
len(df_main[df_main['Label'] == 0])
```

[4]:
```
10564771
```

[5]:
```
len(df_main[df_main['Label'] == 1])
```

[5]:
```
1351342
```

[6]:
```
df_normal = df_main[df_main['Label'] == 0]
df_attack = df_main[df_main['Label'] == 1]
```

[7]:
```
df_normal_downsampled = resample(df_normal, replace=False, n_samples=500000,␣
  ↪random_state=42)
len(df_normal_downsampled)
```

[7]:
```
500000
```

[8]:
```
df_attack_downsampled = resample(df_attack, replace=False, n_samples=500000,␣
 ↪random_state=42)
len(df_attack_downsampled)
```

[8]:
```
500000
```

[9]:
```
df_downsample = pd.concat([df_normal_downsampled, df_attack_downsampled])
len(df_downsample)
```

[9]:
```
1000000
```

Split data

[10]:
```
X = df_downsample.drop(columns='Label')
y = df_downsample['Label']

# Chia dữ liệu thành tập huấn luyện và tập kiểm tra
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)
```

Chuẩn hóa dữ liệu

[11]:
```
scaler = StandardScaler()
X_train_scaled  = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Huấn luyện mô hình

[12]:
```
%%time
# Evaluate normal model
clf_svm = SVC(random_state=42)
clf_svm.fit(X_train_scaled, y_train)
```

```
CPU times: total: 5h
Wall time: 5h 39min 50s
```

[12]:
```
SVC(random_state=42)
```

[13]:
```
clf_svm.C
```

[13]:
```
1.0
```

[14]:
```
clf_svm.gamma
```

[14]:
```
'scale'
```

[15]:
```
clf_svm.kernel
```

[15]:
```
'rbf'
```

[16]:
```
clf_svm.decision_function_shape
```

[16]:
```
'ovr'
```

Đánh giá mô hình

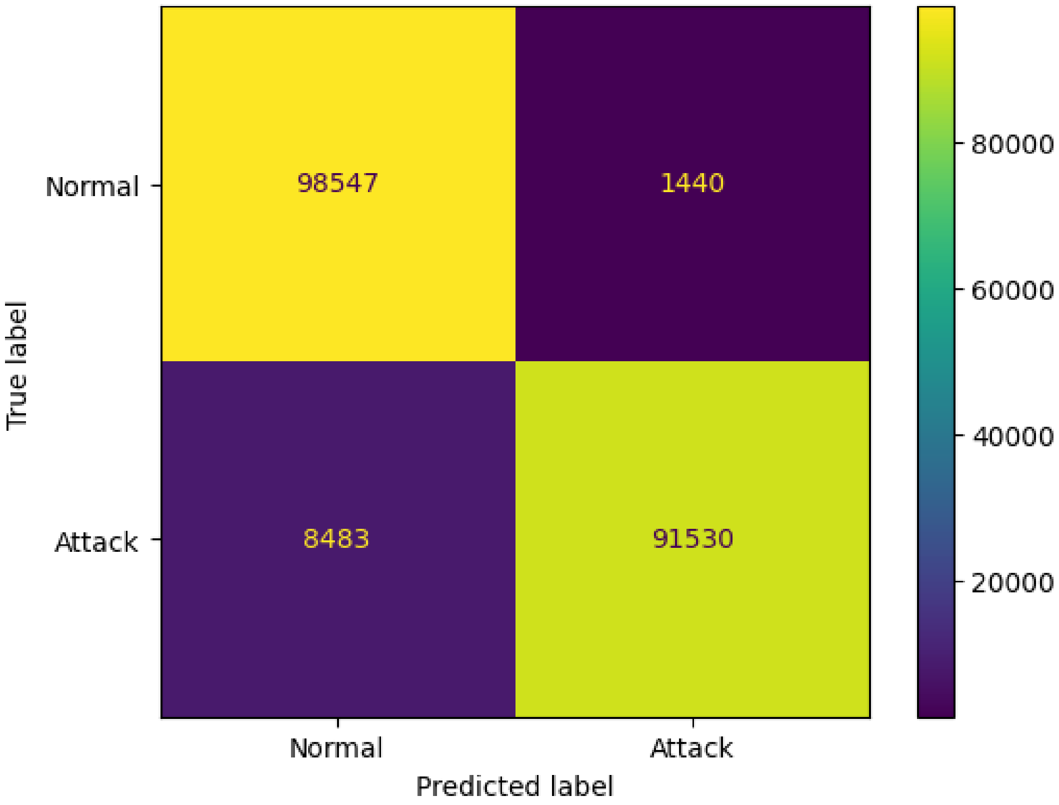
[17]:
```
y_test_pred = clf_svm.predict(X_test_scaled)
accuracy = accuracy_score(y_test, y_test_pred)
precision = precision_score(y_test, y_test_pred)
recall = recall_score(y_test, y_test_pred)
f1 = f1_score(y_test, y_test_pred)
print(f'Accuracy: {accuracy}')
print(f'Precision: {precision}')
print(f'Recall: {recall}')
print(f'F1-Score: {f1}')
```

```
Accuracy: 0.950385
Precision: 0.9845111326234269
Recall: 0.9151810264665593
F1-Score: 0.9485809630900132
```

[18]:
```
# Tính ma trận nhầm lẫn
cm = confusion_matrix(y_test, clf_svm.predict(X_test_scaled))

# Hiển thị ma trận nhầm lẫn
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=["Normal",␣
  ↪"Attack"])
disp.plot(values_format='d')
```

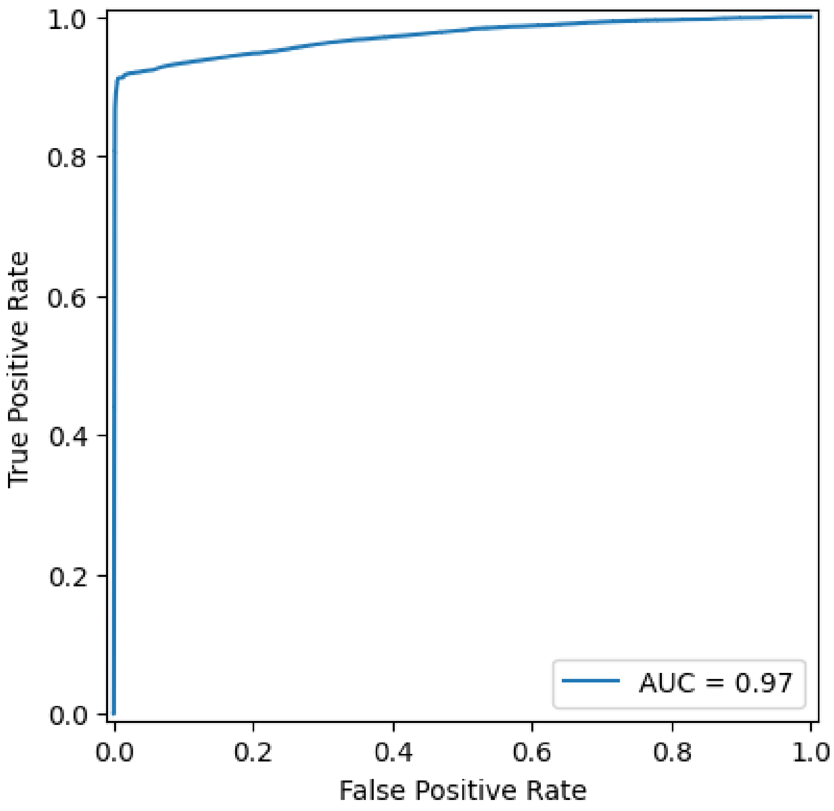
[18]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x21d59af6660>
```



```
[19]: y_score = clf_svm.decision_function(X_test_scaled)
      fpr, tpr, _ = roc_curve(y_test, y_score)
      roc_auc = auc(fpr, tpr)
      print(f'ROC AUC: {roc_auc}')
      RocCurveDisplay(fpr=fpr, tpr=tpr, roc_auc=roc_auc).plot()
```

```
ROC AUC: 0.9719377140757475
```

```
[19]: <sklearn.metrics._plot.roc_curve.RocCurveDisplay at 0x21d59d18e60>
```



[20]:

```
print(classification_report(y_test, y_test_pred))
```

```
              precision    recall  f1-score   support

           0       0.92      0.99      0.95     99987
           1       0.98      0.92      0.95    100013

    accuracy                           0.95    200000
   macro avg       0.95      0.95      0.95    200000
weighted avg       0.95      0.95      0.95    200000
```