utils.py

```python
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay, f1_score, recall_score, precision_score, accuracy_score, roc_curve, roc_auc_sc
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np


def draw_roc(test_label, rss):
    fpr, tpr, thresholds = roc_curve(test_label, rss, pos_label=1)
    auc = roc_auc_score(test_label, rss)

    d = {"fpr": fpr, "tpr": tpr, "thresholds": thresholds}
    pd.DataFrame(d).to_csv("ROC.csv")

    plt.figure()
    lw = 2
    plt.plot(
        fpr,
        tpr,
        color="darkorange",
        lw=lw,
        label="ROC curve (area = %0.2f)" % auc,
    )
    plt.plot([0, 1], [0, 1], color="navy", lw=lw, linestyle="--")
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel("False Positive Rate")
    plt.ylabel("True Positive Rate")
    plt.title("Receiver operating characteristic example")
    plt.legend(loc="lower right")
    plt.show()

    i = np.arange(len(tpr))
    roc = pd.DataFrame({'tf': pd.Series(tpr - (1 - fpr), index=i), 'threshold': pd.Series(thresholds, index=i)})
    roc_t = roc.iloc[(roc.tf - 0).abs().argsort()[:1]]

    return list(roc_t['threshold'])[0]


def get_metric(test_label, rss, opt_threshold):
    pred = np.zeros_like(rss)
    pred[rss > opt_threshold] = 1
    print(pred, test_label)

    cf = confusion_matrix(test_label, pred)
    disp = ConfusionMatrixDisplay(cf, display_labels=["normal", "anomaly"])
    disp.plot()

    f1 = f1_score(test_label, pred)
    recall = recall_score(test_label, pred)
    precis = precision_score(test_label, pred)
    accu = accuracy_score(test_label, pred)
    plt.show()

    return accu, f1, recall, precis
```