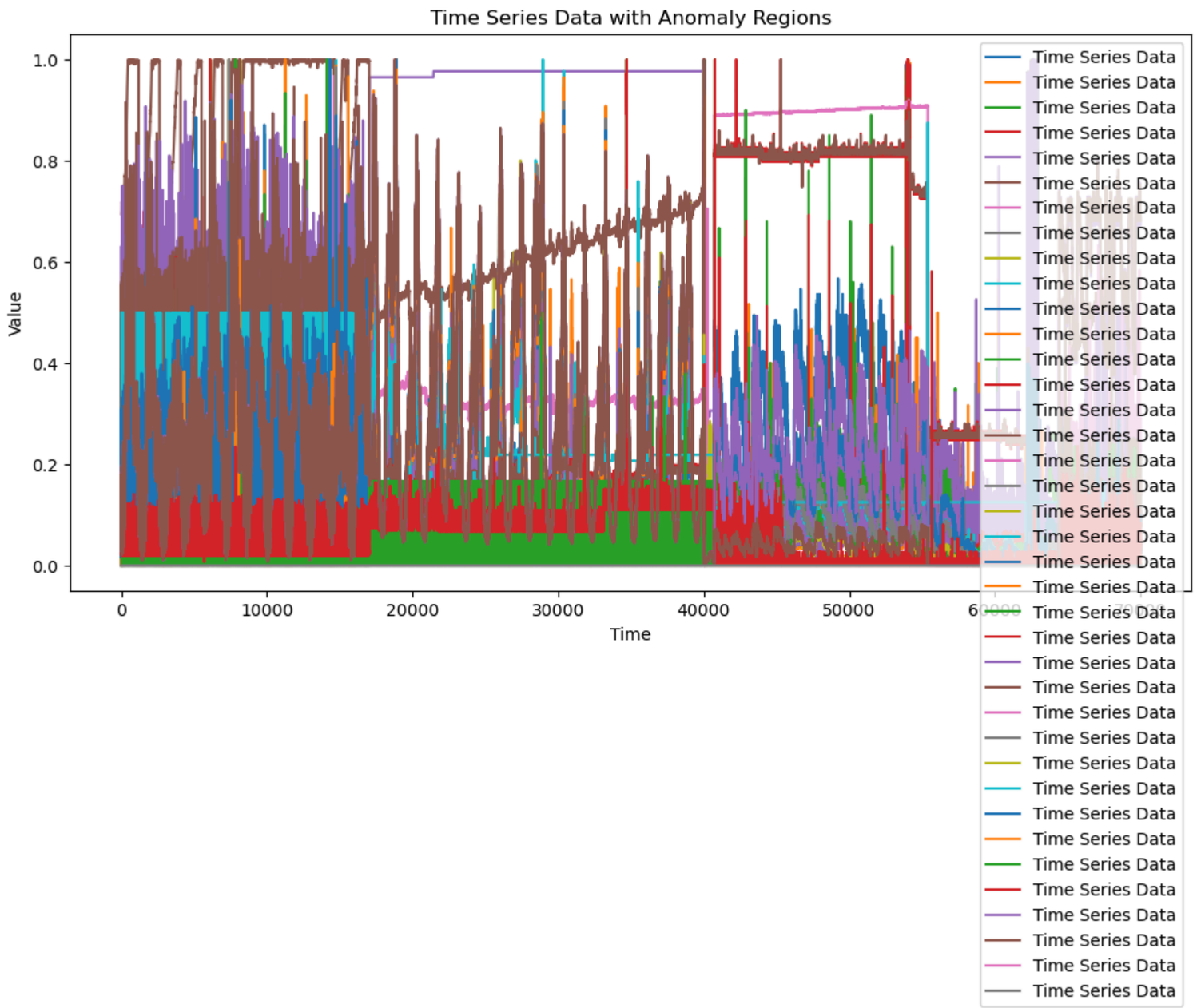
In [6]:
```python
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier

# a) Read test and label files
test_data = pd.read_csv("test.csv")
test_labels = pd.read_csv("test_label.csv")

# b) Draw time series plots with anomaly regions
def plot_with_anomaly(data, labels):
    plt.figure(figsize=(12, 6))
    plt.plot(data.index, data.values, label='Time Series Data')
    # Check if the label column exists and get its name
    label_column = labels.columns[0] if len(labels.columns) == 1 else 'Label'
    # Check if the label column contains the value 1 for anomalies
    if label_column in labels.columns and labels[label_column].nunique() > 1:
        anomalies = labels[labels[label_column] == 1]
        for _, row in anomalies.iterrows():
            # Ensure that the column names for start and end points are correct
            start = row.get('Start')  # Replace 'Start' with the actual column name if different
            end = row.get('End')  # Replace 'End' with the actual column name if different
            if pd.notna(start) and pd.notna(end):
                plt.axvspan(start, end, color='red', alpha=0.3, label='Anomaly Region')
    plt.xlabel('Time')
    plt.ylabel('Value')
    plt.title('Time Series Data with Anomaly Regions')
    plt.legend()
    plt.show()

plot_with_anomaly(test_data, test_labels)

# c) Perform EDA
print("Basic Statistical Information:")
print(test_data.describe())

# d) Find potential root causes using feature importance analysis
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(test_data, test_labels.values.ravel())  # Flatten the target variable
feature_importances = rf.feature_importances_
feature_importance_df = pd.DataFrame({'Feature': test_data.columns, 'Importance': feature_importances})
feature_importance_df = feature_importance_df.sort_values(by='Importance', ascending=False)
top_n = 5  # Number of top features to display
print(f"Top {top_n} features contributing to anomalies:")
print(feature_importance_df.head(top_n))
```




```
Basic Statistical Information:
                 0             1             2             3             4  \
count  70001.000000  70001.000000  70001.000000  70001.000000  70001.000000
mean       0.125281      0.025424      0.034415      0.037462      0.322057
std        0.148530      0.072388      0.088629      0.093520      0.456736
min        0.000000      0.000000      0.000000      0.000000      0.000000
25%        0.010309      0.001495      0.001873      0.002546      0.000000
50%        0.070707      0.004785      0.005242      0.006364      0.000000
75%        0.191919      0.020927      0.029270      0.031797      0.976471
max        1.000000      1.000000      1.000000      1.000000      0.976471

                 5             6        7             8             9  ...  \
count  70001.000000  70001.000000  70001.0  70001.000000  70001.000000  ...
mean       0.459721      0.336344      0.0      0.011226      0.000749  ...
std        0.347015      0.312518      0.0      0.056158      0.015199  ...
min        0.000000      0.000000      0.0      0.000000      0.000000  ...
25%        0.068121      0.060259      0.0      0.000058      0.000000  ...
50%        0.534963      0.309052      0.0      0.001867      0.000000  ...
75%        0.685973      0.346061      0.0      0.006221      0.000000  ...
max        1.000000      1.000000      0.0      1.000000      1.000000  ...

                28            29            30            31            32  \
count  70001.000000  70001.000000  70001.000000  70001.000000  70001.000000
mean       0.000008      0.169050      0.191246      0.104893      0.018708
std        0.001740      0.075581      0.170423      0.187695      0.054665
min        0.000000      0.000000      0.000000      0.000000      0.000000
25%        0.000000      0.125000      0.061520      0.000000      0.000000
50%        0.000000      0.149425      0.137931      0.001101      0.000000
75%        0.000000      0.218391      0.254096      0.114316      0.000000
max        0.455867      0.875000      1.000000      1.000000      1.000000

                33            34            35       36       37
count  70001.000000  70001.000000  70001.000000  70001.0  70001.0
mean       0.051599      0.269691      0.229657      0.0      0.0
std        0.047420      0.213860      0.230937      0.0      0.0
min        0.000000      0.000000      0.000000      0.0      0.0
25%        0.013514      0.099265      0.042573      0.0      0.0
50%        0.028623      0.224793      0.137634      0.0      0.0
75%        0.089041      0.396694      0.391304      0.0      0.0
max        1.000000      1.000000      1.000000      0.0      0.0

[8 rows x 38 columns]
Top 5 features contributing to anomalies:
   Feature  Importance
5        5    0.129956
2        2    0.064763
3        3    0.064378
```

```
1        1    0.059550
35      35    0.051002
```

In [ ]: