Outlier Detection in HAR Data

By Rahila Mohammed Ilegbodu
Student Number: 224308922
Advanced Data Mining

Introduction:

This report presents an analysis of a Python code designed to detect outliers in HAR data using the Isolation Forest algorithm.

Code Overview:

The provided Python code is designed to perform outlier detection in HAR data using Isolation Forest. Here's an overview of the code's functionality:

Dataset Loading:

- The code loads HAR datasets stored as CSV files. Each CSV file represents the activity data of a single subject. The load_dataset function reads the CSV files, extracts the data, and organizes it into numpy arrays for further processing.

Outlier Detection using Isolation Forest:

- Isolation Forest is a machine learning algorithm used for anomaly detection. The code utilizes the Isolation Forest implementation from the scikit-learn library.
- The detect_outliers_isolation_forest function applies Isolation Forest to the subject's data to identify outliers. The algorithm assigns anomaly scores to each data point, with lower scores indicating a higher likelihood of being an outlier.

Visualization:

- The code provides visualizations to assist in understanding the data and the detected outliers.
- Scatter plots are generated to visualize the entire subject's data. Outliers are highlighted on the scatter plots using red crosses, making it easier to identify anomalous data points.

Implementation:

The implementation of the code follows a systematic approach:

Data Loading:

- The load_dataset function iterates through a directory containing CSV files, reads each file using the Pandas library, and extracts the data into numpy arrays.

```
import numpy as np
from os import listdir
from pandas import read_csv

# Load sequence for each subject, returns a list of numpy arrays
def load_dataset(prefix='./'):
    subjects = []
    directory = prefix
    for name in listdir(directory):
        filename = directory + name
        if filename.endswith('.csv'):
            df = read_csv(filename, header=None)
            # drop row number
            values = df.values[:, 1:]
            subjects.append(values)
    return subjects
```

Outlier Detection:

- The detect_outliers_isolation_forest function applies Isolation Forest to the subject's data to detect outliers. An Isolation Forest model is trained on the data, and outliers are identified based on their anomaly scores.

```
from sklearn.ensemble import IsolationForest

# Outlier detection function using Isolation Forest
def detect_outliers_isolation_forest(subject):
    iso_forest = IsolationForest(contamination=0.1)
    outliers = iso_forest.fit_predict(subject)
    return outliers
```

Visualization:

- The scatter_plot_subject_with_outliers function creates scatter plots to visualize the entire subject's data. Outliers are marked on the scatter plots using red crosses for easy identification.

```
import matplotlib.pyplot as plt

# Scatter plot for the entire subject's data with outliers marked
def scatter_plot_subject_with_outliers(subject, outliers):
    plt.figure()
    plt.scatter(range(len(subject)), subject, label='Data Points',
color='purple')

    if outliers is not None:
        outliers_mask = outliers == -1
        flattened_outliers_mask = outliers_mask.flatten()[:len(subject)]  #
Adjust the mask length
        plt.scatter(
            np.where(flattened_outliers_mask)[0],  # x values
            subject[flattened_outliers_mask],  # y values
            label='Outliers',
            color='red',
            marker='x'
        )

    plt.xlabel('Data Point Index')
    plt.ylabel('Values')
    plt.title('Scatter Plot for the Entire Subject with Outliers')
    plt.legend()
    plt.show()
```

Conclusion:

The implemented code offers an effective solution for detecting outliers in HAR data using Isolation Forest. By leveraging machine learning techniques, the code enables the identification of anomalous activities, contributing to the improvement of HAR systems' accuracy and reliability. The visualizations provided aid in the interpretation of outlier detection results, facilitating further analysis and decision-making in HAR applications.