Lab: Clustering on IOT-detection and NSL-KDD

By : Djallel DILMI

Objective:

In this lab, you will learn how to apply two basic clustering algorithms—**K-means** and **Hierarchical Clustering (CAH)**—on two datasets: **NSL-KDD** (network intrusion detection) and **IoT-detection** (device activity detection). By the end of the lab, you should be able to run these algorithms and interpret the resulting clusters.

Part 1: Introduction to Clustering

Clustering is a way of grouping similar data points together. For example, if you have data about network traffic, you might want to group the traffic into "normal" behavior and "suspicious" activity. Similarly, you could group IoT devices based on their activity patterns.

There are many clustering algorithms, but today, we'll focus on:

1. **K-means**: This algorithm divides data into a specified number of clusters by finding groups that are similar.
2. **Hierarchical Clustering (CAH)**: This method builds clusters step-by-step by merging similar data points.

Part 2: Loading the Data

You'll be working with two datasets:

- **NSL-KDD**: This dataset is used for detecting network intrusions (e.g., detecting attacks).
- **IoT-detection**: This dataset contains activity data from IoT devices, which you can group into clusters based on behavior.

For simplicity, use the datasets you have pre-processed in previous labs.

Part 3: K-means Clustering

K-means works by dividing the data into a number of clusters. The algorithm tries to find clusters where the data points are close to each other.

1. **Run the K-means Algorithm:**
    - We will use **3 clusters** for simplicity.
    - After running the algorithm, we will visualize the clusters in a 2D plot using two features from the dataset.
2. **Code for K-means Clustering:**

```
import pandas as pd
from sklearn.cluster import KMeans
from matplotlib import pyplot as plt
# Load the dataset (NSL-KDD or IoT-detection)
data = pd.read_csv('nsl_kdd_simple.csv')  # Or use 'iot_detection.csv'
# K-means Clustering with 3 clusters
kmeans = KMeans(n_clusters=3)
kmeans_labels = kmeans.fit_predict(data)
# Plotting the clusters using the first two features
plt.scatter(data.iloc[:, 0], data.iloc[:, 1], c=kmeans_labels, cmap='rainbow')
plt.title('K-means Clustering')
plt.show()
```

Part 4: Hierarchical Clustering (CAH)

Hierarchical clustering works by progressively merging the most similar data points into clusters. You can visualize this process using a **dendrogram**, which shows how clusters are formed step-by-step.

1. **Run the Hierarchical Clustering Algorithm:**
    - We'll use the **Ward method** to group similar points.
    - The dendrogram will help you visualize the clustering process.
2. **Code for Hierarchical Clustering (CAH):**

```
from scipy.cluster.hierarchy import dendrogram, linkage
from matplotlib import pyplot as plt
# Create the linkage matrix using the Ward method
linked = linkage(data, method='ward')
# Plot the dendrogram
plt.figure(figsize=(10, 7))
dendrogram(linked)
plt.title('Dendrogram - Hierarchical Clustering')
plt.show()
```

Part 5: Questions to Answer

After running the K-means and Hierarchical Clustering algorithms, answer the following questions:

1. **K-means Clustering**:
    - How many clusters did K-means create?
    - Can you see any clear separation between clusters in the plot?
2. **Hierarchical Clustering (CAH)**:
    - Look at the dendrogram. How many clusters do you think are present based on the dendrogram?
    - How does Hierarchical Clustering differ from K-means in grouping the data?