**1．实验目的**

（1）掌握k-means聚类的应用

（2）掌握基于密度的聚类DBSCAN方法

（3）能够对聚类结果进行适当的可视化

（4）掌握聚类性能评价方法

**2．任务描述**

2.1. 本实验需要完成的任务

（1）读取NSL-KDD 99数据，完成数据预处理。

（2）筛选有价值的字段，采用至少两种聚类算法，对该数据集聚类。

（3）利用聚类评价方法，评价聚类结果。

2.2. 实验背景描述

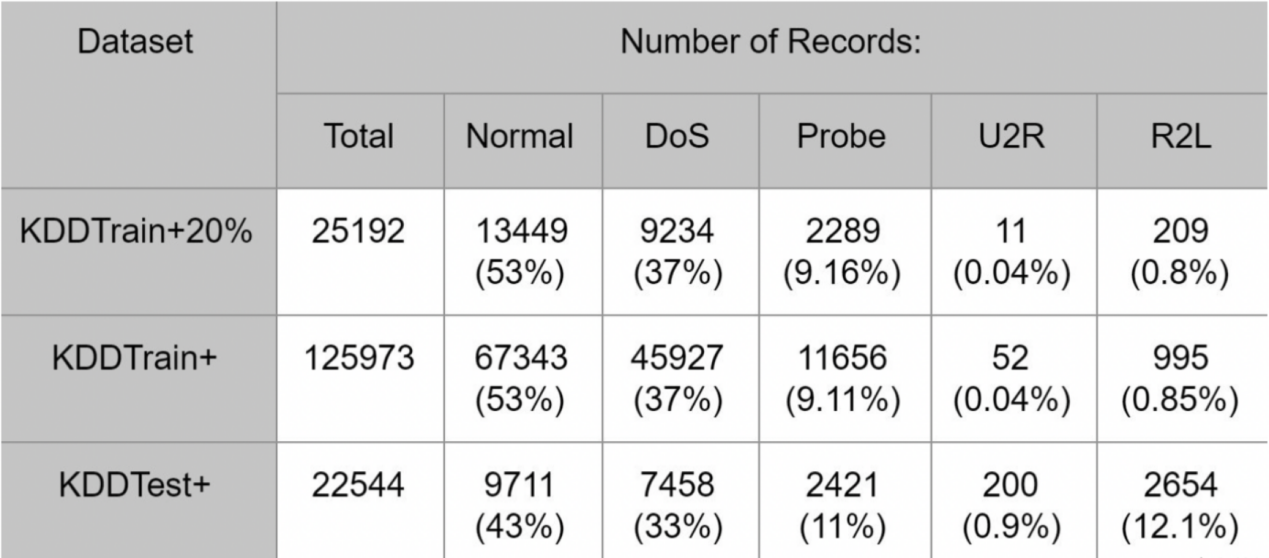
本实验的任务是进行入侵流量的聚类分析。入侵流量是指网络中传输的数据包，在网络中的传输行为受到不同攻击或入侵的影响，因此具有不同的模式和特征。聚类分析的目标是将入侵流量归为一类，并将其与正常流量进行区分。这样可以帮助识别并理解攻击流量，从而提高入侵检测的准确性和效率。

在实验中，首先需要准备包含正常流量和入侵流量的数据集。数据集应该经过预处理，包括特征提取、缺失值处理和标准化等步骤。然后，选择适当的聚类算法，如K均值聚类和DBSCAN，对数据集进行聚类分析，聚类的结果将形成两个不同的簇（clusters），一个代表正常流量，一个代表恶意流量，并生成可视化图。接着，对聚类结果进行评估，以评估聚类性能和效果。

2.3. 实验数据集介绍

NSL-KDD数据集是数据集KDDCup-99的修订版本，该数据集由四个子数据集组成：KDDTest+、KDDTest-21、KDDTrain+、KDDTrain+\_20Percent。其中KDDTest-21和 KDDTrain+\_20Percent是KDDTrain+和KDDTest+的子集。数据集每条记录包含 43个特征，其中41个特征指的是流量输入本身，最后两个是标签（正常或攻击）和攻击分数（流量输入的严重性）。

数据集中存在 4 种不同类型的攻击：拒绝服务(DoS)、探测（Probe）、用户到根(U2R) 和远程到本地(R2L)。每种攻击类型的数据分布如下：



实验使用KDDTrain+数据集进行聚类分析操作。

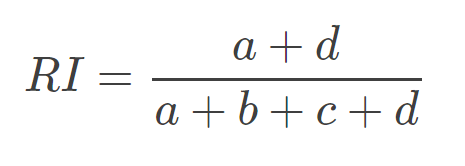
**3. 相关知识**

3.1. 兰德系数

兰德系数（Rand index）是一种用于衡量两个数据分布之间相似度的指标，通常用于聚类算法的评估。它是以其发明者，W. Rand的名字命名的。

考虑两个聚类结果：一个是算法得到的聚类结果，另一个是实际标记的真实聚类结果。兰德系数通过计算两个聚类结果中数据点的一致性和不一致性来评估它们的相似性。

兰德系数的定义如下：



兰德系数是一种简单且直观的聚类评估指标，但它对于数据集中的噪声点敏感，并且在聚类结果中簇的数量不一致时效果可能较差。因此，在实际应用中，通常会结合其他评估指标来综合评估聚类算法的性能，兰德系数的值在[0,1]之间，比例越大，说明聚类效果越好，与真实情况越拟合。

3.2. 互信息

互信息（Mutual Information，MI）是一种用于衡量两个随机变量之间相互依赖程度的信息论指标。在聚类算法评估中，互信息常用于度量聚类结果与真实标签之间的相似性，进而评估聚在聚类算法评估中，互信息用于比较聚类结果与真实标签之间的相似性，计算两者之间的互信息值。较高的互信息值表示聚类结果与真实标签之间的相似性较高，聚类效果较好；而较低的互信息值则表示聚类结果与真实标签之间的一致性较差。

需要注意的是，互信息可能会受到聚类结果中簇的数量和标签数量的影响。在使用互信息进行聚类评估时，通常需要对数据进行适当的预处理，确保聚类结果与真实标签之间的对应关系正确。取值范围为[0,1]，值越大意味着聚类结果与真实情况越吻合。

**4.实验步骤与要求**

4.1 实验步骤

（1）读取数据集并查看数据特征。

代码如下：

df = pd.read\_csv('handledKDDTest.csv')

df.info()

（2）使用describe()函数统计数据集中各特征的信息，便于更好地数据观察以及预处理。

示例代码：

df.describe()

（3）对数据预处理，分为xdata和ylabel两个部分，来去除标记对聚类的影响。

示例代码：

ylabel = df['1.5']

xdata = df.drop('1.5', axis = 1)

（4）对特征数据进行归一化，使用MinMaxScaler将数据缩放到[0, 1]范围内，并使用PCA进行降维：使用PCA将特征数据xdata降至2维，以便后续可视化。

示例代码：

min\_max\_scaler = preprocessing.MinMaxScaler()

xdata = min\_max\_scaler.fit\_transform(df)

xdata = PCA(n\_components=2).fit\_transform(xdata)

（5）K-means聚类：使用K-means算法将降维后的特征数据xdata聚为两个簇。

示例代码：

k\_model = KMeans(n\_clusters=2)

k\_model.fit(xdata)

predict\_y = k\_model.predict(xdata)

（6）评估K-means聚类结果，使用兰德系数、调整互信息、均一性、完整性和V-measure等指标对K-means聚类结果与真实标签ylabel进行评估。

示例代码：

from sklearn.metrics import adjusted\_rand\_score

ari = adjusted\_rand\_score(ylabel, label\_pred)

print('K-Means 兰德系数为：%f' % (ari))

from sklearn.metrics import adjusted\_mutual\_info\_score

ami = adjusted\_mutual\_info\_score(ylabel, label\_pred)

print('K-Means 互信息为：%f' % (ami))

from sklearn import metrics

h\_score = metrics.homogeneity\_score(ylabel, label\_pred)

c\_score = metrics.completeness\_score(ylabel, label\_pred)

V\_measure = metrics.v\_measure\_score(ylabel, label\_pred)

print('K-Means h\_score为：%f \nK-Means c\_score为：%f \nK-Means V\_measure为：%f' % (h\_score, c\_score, V\_measure))

（7）使用DBSCAN聚类算法对降维后的数据xdata进行聚类。

示例代码：

from sklearn.cluster import DBSCAN

clustering = DBSCAN(eps=0.065, min\_samples=2).fit(xdata)

（8）评估DBSCAN聚类结果：使用兰德系数、互信息、均一性（Homogeneity）、完整性和V-measure等指标对DBSCAN聚类结果与真实标签ylabel进行评估。

示例代码：

from sklearn.metrics import adjusted\_rand\_score

ari = adjusted\_rand\_score(ylabel, label\_pred2)

print('DBScan 兰德系数为：%f' % (ari))

from sklearn.metrics import adjusted\_mutual\_info\_score

ami = adjusted\_mutual\_info\_score(ylabel, label\_pred2)

print('DBScan 互信息为：%f' % (ami))

from sklearn import metrics

h\_score = metrics.homogeneity\_score(ylabel, label\_pred2)

c\_score = metrics.completeness\_score(ylabel, label\_pred2)

V\_measure = metrics.v\_measure\_score(ylabel, label\_pred2)

print('DBScan h\_score为：%f \nDBScan c\_score为：%f \nDBScan V\_measure为：%f' % (h\_score, c\_score, V\_measure))

**5. 测试说明**

（1）数据集读取，并查看数据特征。

（2）统计数据集特征信息。

（3）将数据集分类，去除标记对聚类的影响。

（4）对特征数据进行归一化处理并降维。

（5）使用K-means聚类。

（6）评估K-means聚类结果。

（7）使用DBSCAN聚类。

（8）评估DBSCAN结果。

6．分析与讨论

（1）分析聚类的可视化结果，是否可以通过可视化结果直观地观察聚类算法的效果。

（2）讨论聚类与分类方法的优缺点和应用范围。

（3）讨论评估指标如何量化聚类结果的好坏，是否可以帮助我们选择合适的聚类算法和参数。