西安電子科技大學
XIDIAN UNIVERSITY

人工智能学院

智能数据挖掘课程作业报告

KDD CUP 网络入侵数据集小波分解

姓名：杨文韬

学号：18020100245

班级：1920012

2022 年 03 月 29 日

目录

1 KDD CUP99 数据集简介 **1**

2 原理分析 **1**

2.1 小波变换 1

2.2 离散小波变换 2

3 实验过程 **2**

4 附录 **3**

KDD CUP 网络入侵数据集小波分解

1 KDD CUP99 数据集简介

这是用于第三届国际知识发现和数据挖掘工具竞赛的数据集，该竞赛与 KDD-99 第五届知识发现和数据挖掘国际会议同时举行。竞赛任务是建立一个网络入侵检测器，一个能够区分“坏”连接（称为入侵或攻击）和“好”正常连接的预测模型。该数据库包含一组要审计的标准数据，其中包括在军事网络环境中模拟的各种入侵。由于数据集较大，我使用完整数据 10% 的子集来进行实验。

2 原理分析

2.1 小波变换

在进行深度学习训练时会使用到大量的数据，这些数据中有会存在一些噪声，小波变换 (Wavelet Transform, WT) 可以用来去除数据中的噪声。小波变换继承和发展了短时傅立叶变换局部化的思想，同时又克服了窗口大小不随频率变化等缺点，能够提供一个随频率改变的“时间-频率”窗口，是进行信号时频分析和处理的理想工具。小波变换有两个变量：尺度 a 和平移量 b。尺度 a 控制小波函数的伸缩，平移量 b 控制小波函数的平移。尺度就对应于频率 (反比)，平移量 b 就对应于时间。

$$w(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(t) \cdot \psi \left(\frac{t-b}{a} \right) dt \tag{1}$$

小波分解的意义就在于能够在不同尺度上对信号进行分解，而且对不同尺度的选择可以根据不同的目标来确定。

对于许多信号，低频成分相当重要，它常常蕴含着信号的特征，而高频成分则给出信号的细节或差别。人的话音如果去掉高频成分，听起来与以前可能不同，但仍能知道所说的内容；如果去掉足够的低频成分，则听到的是一些没有意义的声音。在小波分析中经常用到近似与细节。近似表示信号的高尺度，即低频信息；细节表示信号的低尺度，即高频信息。因此，原始信号通过两个相互滤波器产生两个信号。

通过不断的分解过程，将近似信号连续分解，就可以将信号分解成许多低分辨率成分。理论上分解可以无限制的进行下去，但事实上，分解可以进行到细节（高频）只包含单个样本为止。因此，在实际应用中，一般依据信号的特征或者合适的标准来选择适当的分解层数。

2.2 离散小波变换

在数字图像处理中，需要将连续的小波及其小波变换离散化。一般计算机实现中使用二进制离散处理，将经过这种离散化的小波及其相应的小波变换成为离散小波变换 (Discrete Wavelet Transform, DWT)。实际上，离散小波变换是对连续小波变换的尺度、位移按照 2 的幂次进行离散化得到的，所以也称之为二进制小波变换。

3 实验过程

首先导入数据集并预处理添加属性名，用 `DataFrame._get_numeric_data Examples` 获取定量型数据。

然后使用 pywt 库中的单级离散小波变换函数 `cA, cD = pywt.dwt(data, wavelet, mode='symmetric', axis=-1)`，其中 cA 表示近似系数 (approximation coefficients)，cD 表示细节系数 (detail coefficient)，一般近似系数代表信号中的低频信息，细节系数代表信号中的高频信息，低频信息则代表整段信号的整体特征，高频信息则代表信号中的细节特征，对于 axis 参数，axis = 0 代表对横轴操作，也就是第 0 轴；axis = 1 代表对纵轴操作，也就是第 1 轴。实验中我采用 db1 小波基分解，具体结果见附录。

4 附录

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import pywt
%matplotlib inline
```

In [2]:
```
sns.set_style("darkgrid", {"grid.color": ".6", "grid.linestyle": ":"})
sns.set_theme(font='Times New Roman', font_scale=1.2)
plt.rc("figure", autolayout=True)
# Chinese support
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False
```

In [3]:
```
col_names = ["duration","protocol_type","service","flag","src_bytes",
    "dst_bytes","land","wrong_fragment","urgent","hot","num_failed_logins",
    "logged_in","num_compromised","root_shell","su_attempted","num_root",
    "num_file_creations","num_shells","num_access_files","num_outbound_cmds",
    "is_host_login","is_guest_login","count","srv_count","serror_rate",
    "srv_serror_rate","rerror_rate","srv_rerror_rate","same_srv_rate",
    "diff_srv_rate","srv_diff_host_rate","dst_host_count","dst_host_srv_count",
    "dst_host_same_srv_rate","dst_host_diff_srv_rate","dst_host_same_src_port_rate",
    "dst_host_srv_diff_host_rate","dst_host_serror_rate","dst_host_srv_serror_rate",
    "dst_host_rerror_rate","dst_host_srv_rerror_rate","label"]
df = pd.read_csv('./kddcup.data_10_percent_corrected', header=None, names=col_names)
print(df.shape)
# df.describe() # 结果太长不添加在附录中
```

```
(494021, 42)
```

In [4]:
```
df.head()
```

Out[4]:

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	\
0	0	tcp	http	SF	181	5450	0	
1	0	tcp	http	SF	239	486	0	
2	0	tcp	http	SF	235	1337	0	

```
3         0           tcp    http   SF        219       1337     0
4         0           tcp    http   SF        217       2032     0

   wrong_fragment  urgent  hot  ...  dst_host_srv_count  \
0               0       0    0  ...                   9
1               0       0    0  ...                  19
2               0       0    0  ...                  29
3               0       0    0  ...                  39
4               0       0    0  ...                  49

   dst_host_same_srv_rate  dst_host_diff_srv_rate  \
0                     1.0                     0.0
1                     1.0                     0.0
2                     1.0                     0.0
3                     1.0                     0.0
4                     1.0                     0.0

   dst_host_same_src_port_rate  dst_host_srv_diff_host_rate  \
0                         0.11                          0.0
1                         0.05                          0.0
2                         0.03                          0.0
3                         0.03                          0.0
4                         0.02                          0.0

   dst_host_serror_rate  dst_host_srv_serror_rate  dst_host_rerror_rate  \
0                   0.0                       0.0                   0.0
1                   0.0                       0.0                   0.0
2                   0.0                       0.0                   0.0
3                   0.0                       0.0                   0.0
4                   0.0                       0.0                   0.0

   dst_host_srv_rerror_rate    label
0                       0.0  normal.
1                       0.0  normal.
2                       0.0  normal.
3                       0.0  normal.
4                       0.0  normal.
```

```
        [5 rows x 42 columns]
```

```
In [5]: nums_data = df._get_numeric_data() # 获取定量型数据
        nums_data.values
```

```
Out[5]: array([[0.000e+00, 1.810e+02, 5.450e+03, ..., 0.000e+00, 0.000e+00,
                0.000e+00],
               [0.000e+00, 2.390e+02, 4.860e+02, ..., 0.000e+00, 0.000e+00,
                0.000e+00],
               [0.000e+00, 2.350e+02, 1.337e+03, ..., 0.000e+00, 0.000e+00,
                0.000e+00],
               ...,
               [0.000e+00, 2.030e+02, 1.200e+03, ..., 1.000e-02, 0.000e+00,
                0.000e+00],
               [0.000e+00, 2.910e+02, 1.200e+03, ..., 1.000e-02, 0.000e+00,
                0.000e+00],
               [0.000e+00, 2.190e+02, 1.234e+03, ..., 1.000e-02, 0.000e+00,
                0.000e+00]])
```

```
In [6]: # cA: 近似系数 cD: 细节系数
        # 近似系数：低频信息 细节系数：高频信息
        # 低频信息：整段信号的整体特征 高频信息：信号中的细节特征
        (cA, cD) = pywt.dwt(nums_data, 'db1', axis=0) # 离散小波变换
```

```
In [7]: cA
```

```
Out[7]: array([[0.00000000e+00, 2.96984848e+02, 4.19738585e+03, ...,
                0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
               [0.00000000e+00, 3.21026479e+02, 1.89080353e+03, ...,
                0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
               [0.00000000e+00, 3.06884343e+02, 2.87368196e+03, ...,
                0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
               ...,
               [0.00000000e+00, 4.18607214e+02, 2.94651396e+03, ...,
                1.41421356e-02, 0.00000000e+00, 0.00000000e+00],
               [0.00000000e+00, 3.49310750e+02, 1.69705627e+03, ...,
                1.41421356e-02, 0.00000000e+00, 0.00000000e+00],
               [0.00000000e+00, 3.09712770e+02, 1.74513954e+03, ...,
                1.41421356e-02, 0.00000000e+00, 0.00000000e+00]])
```

```
In [8]: cD
```

```
Out[8]: array([[   0.        ,  -41.01219331, 3510.07806181, ...,    0.        ,
                   0.        ,    0.        ],
               [   0.        ,   11.3137085 ,    0.        , ...,    0.        ,
                   0.        ,    0.        ],
               [   0.        ,    0.        ,    0.        , ...,    0.        ,
                   0.        ,    0.        ],
               ...,
               [   0.        ,   19.79898987, -286.37824638, ...,    0.        ,
                   0.        ,    0.        ],
               [   0.        , -62.22539674,    0.        , ...,    0.        ,
                   0.        ,    0.        ],
               [   0.        ,    0.        ,    0.        , ...,    0.        ,
                   0.        ,    0.        ]])
```