****

**HUNAN UNIVERSITY**

数据挖掘

KDD CUP99 实验报告

**小组成员：曹杰（S1710W0821）**

**小组成员：倪雪纯（S1710W0836）**

**学 院：信息科学与工程学院**

**指导老师：陈湘涛**

**2018年1月6日**

**一、实验目的**

培养学生数据挖掘的动手能力、实践能力。

**二、实验工具**

Python。

**三、实验数据**

1.KDDCup99网络入侵检测数据介绍

KDDCup99的原始数据来自于1998年的DARPA入侵检测评估项目，所有的网络数据来自于一个模拟的美国空军局域网，网络中加了很多模拟的攻击。实验的训练数据为7周的网络流量，这些网络流量包含有约500万条网络连接;实验的测试数据为2周的网络流量，包含有约200万条网络连接。该网络环境中模拟的攻击分为4大类，如下表所示，正常的网络连接标记为normal。

|  |  |  |
| --- | --- | --- |
| **标识类型** | **含义** | **具体分类标识** |
| Normal | 正常记录 | normal |
| DOS | 拒绝服务攻击 | back、land、neptune、pod、smurf、teardrop |
| Probing | 监视和其他探测活动 | ipsweep、nmap、portsweep、satan |
| R2L | 来自远程机器的非法访问 | ftp\_write、guess\_passwd、imap、multihop、phf、spy、warezclient、warezmaster |
| U2R | 普通用户对本地超级用户特权的非法访问 | buffer\_overflow、loadmodule、perl、  rootkit |

KDDCup99训练数据集中每个连接记录包含了41个固定的特征属性和1个类标识，标识用来表示该条连接记录是正常的，或是某个具体的攻击类型。在41个固定的特征属性中，9个特征属性为离散(symbolic)型,其他均为连续(continuous)型。

**属性：**duration,protocol\_type,service,flag,src\_bytes,dst\_bytes,land,

wrong\_fragment,urgent,ho,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,class

**实例**：

0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,254,1.00,0.01,0.00,0.00,0.00,0.00,0.00,0.00,normal.

0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,254,1.00,0.01,0.00,0.00,0.00,0.00,0.00,0.00,snmpgetattack.

2.实验数据选取

在实验研究中,使用KDDCup99中的网络入侵检测数据包corrected、做为训练集kddcup\_data\_10percent做为测试集。kddcup\_data\_10percent数据包是对kddcup\_data数据包(约490万条数据记录)10%的抽样。

**三、数据预处理**

1. 数据标识

解压数据包后，得到的是纯文本文件，其中的数据均用逗号隔开，且各个变量没有标识，因此需要先对文件添加csv后缀名，然后用python为数据添加标识python代码在文件中给出。保存文件后，文件便可以利用Weka打开，但在此之前，我们必须对数据进行一些处理。

为了便于分类算法的进行，有必要将数据集（包括测试集）中所有的攻击类别标识统一化，我选择了用0,1,2,3,4分别代表Normal，DoS，Probe，R2L，U2R五个大类，将所有分类标识均用这五个数字代替。由于label的变量值比较多，用python处理比较麻烦，该过程可以利用Excel完成。

2.**特征选择**

对于预处理后的数据，我们需要进行简要的分析，对其中的属性进行筛选。

根据信息熵可知，某一属性的值全相同数时，其带来的信息是为1的，所以是必然发生的事件，据此在属性中，在num\_outbound\_cmds 和 is\_hot\_login 这两个属性上的取值相同均为0；land 属性取值为 1 的 22 条连接记录中有 21 条均属于 land 攻击， 且 land 攻击和 normal 之间， 除land 属性之外的其他属性之间并无明显区别， 因此可以忽略；urgent属性值只有4条非0；删除su\_attempted属性；删除num\_shells属性；删除num\_shells属性。通过数据集中的数据表层信息，已经可以删除其中的 6 个属性和 2 个攻击类型。此时，数据集中还剩余 35 个特征属性和 20 种攻击类型和 1 个 normal 正常行为。**训练集与测试集都要进行上述步骤。**

**所选属性为:1,23,4,5,6,8,10,11,12,13,16,17,18,19,22,23,24,25,26,27,**

**28,29,30,31,32,33,34,35,36,37,38,39,40,41**

**3.数据归一化**

由于训练集和测试集中大部分属性的取值范围不同，无法取得相同的离散区间，因此，在进行离散化之前，需要将部分数据进行归一化。需要进行归一化的数据是取值连续且无固定上下界的数据，如duration，src\_byte等，利用过滤算法中的unsupervised-instance-normalize算法，将区间定为[0,1]，可将数据归一化。

**四、模型评估**

**模型的评估方法比较简单，只用了测试集的准确率和验证集数据的准确率来评估。**

**五、实验方法**

1.随机森林定义

**随机森林是一种比较新的机器学习模型。随机森林对多元共线性不敏感，结果对缺失数据和非平衡的数据比较稳健，可以很好地预测多达几千个解释变量的作用（Breiman 2001b），被誉为当前最好的算法之一（Iverson et al. 2008）。**

**随机森林顾名思义，是用随机的方式建立一个森林，森林里面有很多的决策树组成，随机森林的每一棵决策树之间是没有关联的。在得到森林之后，当有一个新的输入样本进入的时候，就让森林中的每一棵决策树分别进行一下判断，看看这个样本应该属于哪一类（对于分类算法），然后看看哪一类被选择最多，就预测这个样本为那一类。**

1. 随机森林方法理论介绍

2.1 随机森林算法

2.1.1 决策树

决策树（decision tree）是一个树结构（可以是二叉树或非二叉树）。其每个非叶节点表示一个特征属性上的测试，每个分支代表这个特征属性在某个值域上的输出，而每个叶节点存放一个类别。使用决策树进行决策的过程就是从根节点开始，测试待分类项中相应的特征属性，并按照其值选择输出分支，直到到达叶子节点，将叶子节点存放的类别作为决策结果。

随机森林是用随机的方式建立一个森林，森林里面有很多的决策树组成，随机森林的每一棵决策树之间是没有关联的。在得到森林之后，当有一个新的输入样本进入的时候，就让森林中的每一棵决策树分别进行一下判断，看看这个样本应该属于哪一类，然后看看哪一类被选择最多，就预测这个样本为那一类。

在建立每一棵决策树的过程中，有两点需要注意采样与完全分裂。首先是两个随机采样的过程，random forest对输入的数据要进行行、列的采样。对于行采样，采用有放回的方式，也就是在采样得到的样本集合中，可能有重复的样本。假设输入样本为N个，那么采样的样本也为N个。这样使得在训练的时候，每一棵树的输入样本都不是全部的样本，使得相对不容易出现over-fitting。然后进行列采样，从M个feature中，选择m个（m << M）。之后就是对采样之后的数据使用完全分裂的方式建立出决策树，这样决策树的某一个叶子节点要么是无法继续分裂的，要么里面的所有样本的都是指向的同一个分类。一般很多的决策树算法都一个重要的步骤——剪枝，但是这里不这样干，由于之前的两个随机采样的过程保证了随机性，所以就算不剪枝，也不会出现over-fitting。

2.1.2 随机森林实现过程

随机森林中的每一棵分类树为二叉树，其生成遵循自顶向下的递归分裂原则，即从根节点开始依次对训练集进行划分；在二叉树中，根节点包含全部训练数据， 按照节点纯度最小原则，分裂为左节点和右节点，它们分别包含训练数据的一个子集，按照同样的规则节点继续分裂，直到满足分支停止规则而停止生长。若节点n上的分类数据全部来自于同一类别，则此节点的纯度I(n)=0，纯度度量方法是Gini准则，即假设P(Xj)是节点n上属于Xj 类样本个数占训练。

具体实现过程如下：

（1）原始训练集为N，应用bootstrap法有放回地随机抽取k个新的自助样本集，并由此构建k棵分类树，每次未被抽到的样本组成了k个袋外数据；

（2）设有mall个变量，则在每一棵树的每个节点处随机抽取mtry个变量(mtry n mall)，然后在mtry中选择一个最具有分类能力的变量，变量分类的阈值通过检查每一个分类点确定；

（3）每棵树最大限度地生长, 不做任何修剪；

（4）将生成的多棵分类树组成随机森林，用随机森林分类器对新的数据进行判别与分类，分类结果按树分类器的投票多少而定。

**六、实验结果**

**（属性为:1,23,4,5,6,8,10,11,12,13,16,17,18,19,22,23,24,25,26,27,**

**28,29,30,31,32,33,34,35,36,37,38,39,40,41结果。**

