大家好，我们的大作业课题为：电信网络威胁检测。

在本次实验中，我们对基于两份电信网络流量数据集，设计了基于特征化网络流量和机器学习方法的电信网络威胁检测系统（**IDS**）。我们对实验的数据集进行了数据预处理，对特征进行了进一步的筛选和精细处理，使用于训练的特征数据能够更好的拟合出结果；在此基础上，我们采用了多种不同的机器学习方法对特征数据集进行训练和测试，并对不同的模型性能进行分析。

以下是我们的报告内容。

换页。

首先先介绍一下网络威胁检测的含义。

网络威胁（网络入侵）是一系列旨在破坏计算平台上资源的完整性、机密性或可用性的行为。网络威胁检测系统（**IDS**）用于监控网络流量以寻找可疑流量并在发现此类流量的时候发出警报，并在检测到异常的时候采取行动，比如阻止来自可疑地 **IP** 地址的流量等等。

网络威胁检测通常有两种检测技术：

1. 异常检测：基于与正常行为偏离的偏差识别恶意活动，并将这些偏差视为攻击；
2. 特征检测（又称滥用检测）：检测假设入侵者活动可以用一种模式来表示，系统的目标是检测主体活动是否符合这些模式。 它可以将已有的入侵方法检查出来，但对新的入侵方法无能为力。而且此类检测方法的误报率高。

换页：

网络威胁检测系统（**IDS**）的设计通常包括以下四个步骤：

1. 收集数据：为了进行**IDS**，我们需要收集有关网络流量的信息，如流量类型、主机和协议详情。更详细的信息比如 **IP** 地址，设备，请求内容，时间等等。
2. 特征选择：从数据特征中提取有用的特征；
3. 分析数据：对所选特征数据进行分析，以确定数据是否异常；
4. 执行操作：当发生攻击时，**IDS**通过系统管理员生成警报或警报，并告知攻击类型。**IDS**还通过关闭网络端又和终止进程参与控制攻击。

换页：

网络威胁检测面临的挑战网络威胁检测系统面临的挑战包括但不只如下几种：

1. 攻击流量的种类越来越多样化；
2. 攻击流量常常具有伪装性和隐藏性，传统检测手段束手无策；

3. IDS需要处理庞大的数据流，因此需要具备高吞吐量和低延迟；
4. 需要降低威胁检测的误报率以提升效率；

换页：

这是我们本次实验所采用的数据集。

共有两份电信网络威胁数据集，其中给了50000多个特征化的网络流量，其中，‘研判’为数据标签列，包含的标签有：攻击、非攻击、无法判定。

换页：

以上是数据集中部分特征的取值，以及每种特征的取值个数。

换页：

接下来我来介绍一下我们是如何对数据集进行数据预处理工作的。

首先，我们观察数据特征的类型，

事件名称，事件类型，攻击结果，响应码，设备来源，规则ID，响应码，设备动作，研判为离散格式，考虑先编码为整数格式(后采用独热编码)。

威胁等级取值有轻微，一般，较大，重大，为连续整数，考虑分别赋值为 0, 1, 2, 3；

然后我们去掉生成时间和结束时间这两个对研判结果相关性不大的列，再考虑删除重复的行（因为，数据集中的很多攻击都不止进行了一次）。

此时数据库的响应体还有一些为 null，考虑将其替换为空串。

换页：

接下来考虑如何处理请求体和响应体这两个部分的内容：先查看一下研判结果为攻击的行如左图所示，在查看部分研判结果为非攻击的行如右图所示。

可以发现攻击数据和非攻击数据在请求体和响应体上的差异很大，考虑解析这两个部分。

换页：

这是我们找到的3个很具有代表性的，攻击流量中的请求体内容。

可以发现请求体主要包含：请求行(**rstline**)，请求头(**headers**)，请求体(**body**)。
以上述三个为例：

- 请求行可能出现注入攻击：此时请求行的长度比往常的长，可以考虑统计长度和字符出现的频率/方差进行判断。
- 请求体中也可能注入恶意数据，还可能伪装用户代理信息等，可以考虑统计字符出现的频率/方差进行判断。此外还应该重点关注 **Host** 和 **User-Agent** 两行。在本次作业中，决定取 **User-Agent** 行的长度，字符频率方差，**Host** 长度，字符频率方差，**Headers** 字符频率方差共**5**个特征。
- 请求体可能出现远程命令执行攻击：此时请求体相比于别的较为特殊，可以考虑统计字符出现的频率/方差进行判断。

换页：

经过替换后的特征如下，此时共有**23**个特征。

但还没有结束。

换页：

可以发现数据集中，原端口和目的端口的数量最多都达到了**100**，因为他们的值各不相同，而且端口的取值范围也很大，显然采用它们的值或者作为向量作为特征输入到模型中并不是很好的方法。考虑用源端口的数量和目的端口的数量替换原端口，目的端口这两个特征。

接下来查看 **IP** 地址的组成，如右图所示。

对于上述结构，我们考虑将 **IP** 地址转化为整数表示，比如 **151.80.13.43** 转化为 **2538605867**，然后将城市部分提取为字符串格式，用这两个新特征替换原先的特征。

最终的特征如下图所示。

换页：

通过观察数据发现，“无法研判”的行中的请求体往往是一些乱码，而且该类标签对于做网络威胁检测的意义不大，为了方便进行分类问题，首先删除所有研判结果为“无法研判”的行。

至此，数据预处理已经全部完成。右图为数据集中两类标签的占比。

其中非攻击流量有 **98.3** 而攻击流量仅有 **1.7%**。

我们认为这个数据集中攻击流量太少，容易导致训练效果不佳，尤其是会降低召回率（因为模型会认为绝大部分的数据都是非攻击的）。

换页：

接下来我们来定义一下机器学习模型的评价指标。

我们主要采用了输出准确率，精确率，召回率对模型进行评价。

定义模型评估函数：输出准确率，精确率，召回率。

这三个评价指标的定义如下：（正例为攻击）

1. 准确率(**Accuracy**)：

   定义：模型正确预测的样本数量与总样本数量之比。

   公式：$\frac{TP+TN}{TP+FP+TN+FN}$

2. 精确率(**Precision**)：

   定义：所有被模型预测为正类别的样本中，实际为正类别的样本所占的比例。

   公式：$\frac{TP}{TP+FP}$

3. 召回率(**Recall**)：

   定义：所有实际正类别的样本中，模型成功预测为正类别的样本所占的比例。

   公式：$\frac{TP}{TP+FN}$

精确率可以反应模型的是否不容易误判，而召回率可以反应模型对威胁检测的全面性。

换页：

接下来展示模型的效果。

对于决策树模型，我们分别设置了最大深度为 **3**， **6**， **10.**

进行测试。

采用三种最大深度跑出来结果并没有明显的差异，但深度越深，召回率有微小提升，精确率的提升在 **1%** 左右。但是模型明显复杂了很多。

换页：

以下为决策树分析的对于分类问题最重要最大的几个特征。

可以发现源端又的重要性最大，我认为这个可能原因是该特征的取值最多，所以自然带来的信息增益比较大。而响应码的取值虽然不多，但是依然带来很大的信息增益。
此外发生次数，设备动作，规则**ID**，设备来源和请求行长度也起一定作用，其中 请求行长度 **(rstline_len)** 为新分离出的特征，说明其对于分类问题是有价值的。

换页：

KNN 算法的召回率很低，不适合处理该问题。

我们认为一个很重要的原因是训练集中的正例点太少了，在进行对 **KNN** 中的投票结果很不利。

此外我们还尝试了朴素贝叶斯法以及支持向量机算法（其中 **svm** 我们使用了线性核与 **RBF** 核两种核函数进行测试）但是遗憾的是，这两个模型的召回率为**0**。也不适合该分类问题。

其中 **Baiyes** 法失败的原因可能是先验分布不够合理，并且极大似然估计的时候正例点不够，导致估计出来的概率几乎为 **0**。
而 **SVM** 则可能是因为特征空间中无关维度太多，如果经过一定的降维并且增加训练集中的正例点可能可以提升模型能力。

换页：

相比决策树算法，随机森林的精确度高了不少，但是召回率却降低地很明显。

我们认为对于特征较少的小规模数据集，集成算法反而不如单模型来的优秀。

再看右图为随机森林计算出的较为重要的特征。可以发现，由于建立了更多的决策树，对结果产生影响的特征变得更多了。

换页：

eXtreme Gradient Boosting 是一种将弱分类算法提升为一个强分类算法的算法，主要是用决策树模型进行集成，通过反复迭代训练弱学习器来提高模型的性能。每一轮迭代都根据前一轮的结果来调整模型，以减小预测误差。

它在目标函数中引入了正则化项，帮助控制模型的复杂性，防止过拟合。

可以发现，**eGboost** 算法在本分类问题上的的准确率、精确度、召回率的综合表现在上述几个模型中是最好的。

换页：

最后，我们还对模型的泛化能力进行了测试。

我们首先单独在 **0406** 数据集上进行了决策树算法的测试，又把 **0406** 的数据集作为训练集把 **0509** 的数据集作为测试集再进行了一次决策树的测试。两次测试结果如图所示。

可以发现，在 **0406** 数据集上的测试结果在合并后数据集上好，但是用于测试 **0509** 的效果却很不好，说明模型的泛化能力并不够好。
要想要得到泛化能力更好的模型，应该更多的数据，尤其需要不同时间的数据进行训练。而单日的数据集中的网络攻击数量少，种类少，特征单一，不具有很好的可总结性，因此分类效果并不好。

换页：

总结来说，对于特征化流量网络，我们认为决策树算法已经梯度提升树算法的拟合效果是最好的。决策树类算法能够很好的计算出重要的特征用于分类，对于特征网络的适应力极强。而诸如 **SVM**，**KNN**，贝叶斯估计，神经网络等算法，都需要对特征空间进行很好的设计，在算力和数据集规模不够，正例点数量显著不足，特征多且不易处理的情况下，这些算法在设计上要困难许多，得到的结果也不尽如人意。在 **IDS** 的设计中，好的特征是能够更好的做分类问题的至关重要的因素，应当充分利用现有的流量特征提取关键信息。为了应对网络攻击的多样化，我们应当基于更庞大的数据集进行训练才可以得到效果最好的模型。