## 4.3 模型更新

针对逃逸样本，我们采用两种抗攻击的方法来更新之前的模型，1.增加训练样本的个数当训练样本达到一定的数量，就会避免数据过拟合，实现局部最优的情况；2.重新调整特征集也可以使检测率有所提高。

如果我们的特征集已被攻击者利用，我们可通过改变特征集，修改权值，或删除重要特征等操作，重新训练模型。如图2 所示，是模型训练后，按照特征重要性排序的前30个特征，我们可以看到count\_font，count\_javascript，size，count\_obj，count\_endobj这几个特征在分类中占有较多的权值，同时也是非常容易被攻击者利用，来对解析器和分类器进行逃逸，于是我们在训练时就删除了这几个特征，然后重新训练模型，预测结果如表7 所示。

图2 前30个重要特征分布图



表7 是对前5个特征修改后的模型准确率。由表中可知，当分类器使用全部特征进行训练时，模型准确率高达99.82%。当我们将第一个重要的特征在训练的时候删去，检测率基本没有太大的波动，当减到count\_endobj前5个特征时，模型准确率的波动可忽略不计。这也说明了我们的模型可以对抗一些基于特征的攻击，即使对手知道我们分类器使用的特征，模型同样可以达到99%的精度 。

表7 对前5个特征依次删除后的模型准确率

|  |  |
| --- | --- |
| Feature delete train | 准确率 |
| None | 99.82% |
| count\_font | 99.52% |
| count\_javascript | 99.52% |
| Size | 99.64% |
| count\_obj | 99.64% |
| count\_endobj | 99.64% |

同时我们还通过“自减重要特征”的方式来评估模型的鲁棒性，对特征有效性进行研究。我们将模型的特征进行重要性排序，然后依次将最重要的特征逐一删减，并重新使用新特征集重新训练模型。图3是特征在不断自减时所对应的准确度曲线。从图中可知，当特征减少至100个时，重新训练后的模型准确率依然高达90%，这说明：

* 单个特征纵然权重高，当此类特征被删除时，模型准确度会下降，但降幅不大；
* “中等权重”特征的互相作用和叠加，可以使模型健壮，且抵消单个重要特征的缺失影响；
* “中等权重”特征能有效抵御通过改变特征数值的分类器逃逸攻击；

图3 特征自减后的识别率（模型3）



## 4.4 性能评估

为了评估模型的预测性能，我们把数据集随机分为训练（90％）和测试（10％）两部分，并采用10-Fold 交叉验证(Cross Validation）的方法来评估模型。 图4为ROC曲线图，由图可知，ROC曲线下的面积约为1，这表明模型具有良好的预测性能。模型准确度超过99％，与此同时误报率低于0.01％。

图4 ROC曲线图



特征提取是最耗时的操作，因为它需要从硬盘加载所有的文件，并对文件进行逐个解析。于是我们将文件解析与训练分步处理，作为中间结果保存，不仅可以减少CPU 内存占用，同时也可以使模型在更新训练时更为快速。对训练样本的解析（十万级别）共耗时约22分钟。表8 是不同算法之间的训练与预测时间对比，由表可知，随机森林在此任务中不止有良好的准确率，并且预测时间也维持在秒级别。

表8 训练时间与预测时间

|  |  |  |  |
| --- | --- | --- | --- |
|  | 训练时间 | 预测时间 | 准确率 |
| Random Forest | 56s | 1s | 99% |
| Decision Tree | 4s | 1s | 97% |
| SVM | 58m 18s | 12s | 75% |