## 3.3 分类算法 Classification algorithm

首先对收集到的文件进行分类，将提取出来的特征作为训练数据集，在这个时候，随机森林（random forests）在分类上表现了很好的优势，有效且误报率极低，并易于使用,可以很快的对数据进行分类。随机森林分类方法给出的结果是基于很多棵分类树判断结果的集合展现，每一个决策树都是在训练数据中随机选择生成的，因此，随机森林总的来说是一个集成分类器，它使用 bagged training data。通过随机选择的特征子集,，并使用该节点的训练数据，确定每个节点处的最佳分割来创建树中的每个节点。此外，每棵树都是基于一个独立的特征子集，最后，在分类过程中每一个树的投票来确定最终结果。

First, we classify the collected files by extracted features which are used as the training dataset. Random Forests Algorithm we used performs well during classification, with high effectiveness and low false positive rate. Moreover, it is easy to use and can classify data rapidly. The output of random forests classification method is the results ensemble of a multitude of decision trees which is constructed from a randomly selected subset of training data. That said, Random Forests is an ensemble classifier applying the technique of bagging training data. Each node in a decision tree is constructed based on a randomly selected subset of features, as well as the best split at each node, which is determined by training data for that node. Besides, each tree is based on an individual subset of features. Finally, the result is determined by the votes of each tree during classification.

AI引擎的重要 组成部分之一是算法，我们选取了几个实用性较好的算法来比较包括KNN邻近算法，NNET神经网络，RF随机森林和SVM支持向量机，经过多次的训练与分类实验，发现随机森林准确率高，误报率低，低延时鲁棒性良好, 和可解析性等优势，于是将其作为我们选定作为默认算法。并且经过我们的一百多次的验证，在特征发生改变的时候，随机森林准确率依然趋于一个稳定的值。

One of the important component of AI engine is the algorithm. We select several algorithms with good performance to compare including KNN, NNET, Random forests, and SVM. After multiple times of training and classification experiments, we have found that random forests algorithm demonstrates a high detection accuracy, low false positive rate and latency, and good robustness and resolvability. Therefore, we choose Random Forests as our default algorithm. After more than one hundred experiments, the accuracy rate of Random Forests tends to be a stable value even if the features change.

Table 3 算法准确率对比

Table 3 Comparison of Algorithm Accuracy Rate

|  |  |  |  |
| --- | --- | --- | --- |
| SVM | NNET | KNN | RF |
| 75.23% | 82.41% | 97.12% | 99.64% |

## 3.4 模型构建Model Construction

所提出的基于机器学习的恶意PDF文档检测的方法包括以下两个步骤，如图1所示：

1. 提取文件特征。此为基本的预处理步骤，对PDF文件的结构、内容和元数据进行解析，并做相应的向量计算，提取为一个二维的特征集，使得这些特征可以进入到基于机器学习的模型中进行训练分类。

2. 学习和分类。我们会随机选取数据的80%进行训练，通过训练之后保存训练模型，然后使用20%的文件进行预测分类，从中计算出模型的准确率，误报率等信息。

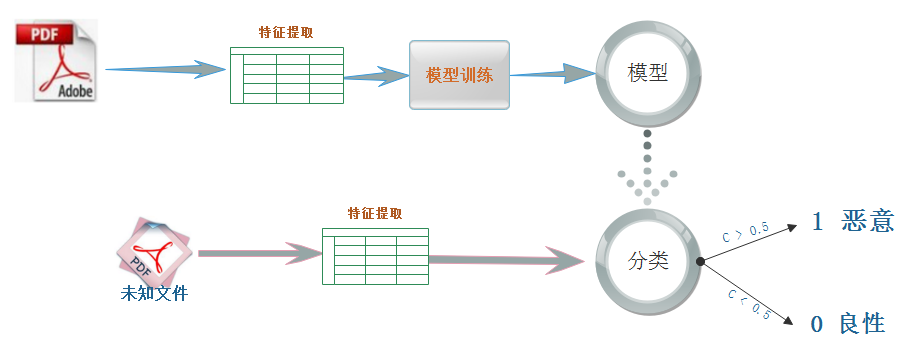
The proposed method of detecting malicious PDF file based on machine learning include the following steps, as shown in Figure 1:

1. Feature Extraction. This is a basic preprocessing step, which is to parse the structure, content and metadata in PDF files and then conduct vector computation on these objects, finally extract them as a two-dimension feature set. These features can be trained and classified in the model based on machine learning.

2. Learning and Classification. We select 80% of data randomly for training and then save the trained model. Later we use 20% of the files for prediction and classification, in order to calculate the information such as accuracy rate, false positive rate, etc.

图1 机器学习的基本框架

Figure 1 Basic Framework of Machine Learning



在我们的实验当中，一共对模型进行了4次更新，其中最开始的模型（model1）是使用peepdf为解析工具提取特征，然后通过一些对特征的计算与量化，使其可以用于机器学习训练与预测。通过peepdf提取的特征数达到133个的时候，这些特征包括有基于结构的（count\_font、size、count\_startxref），内容信息的（title\_oth、subject\_lc）和metadata(producer\_oth、producer\_len)的一些静态属性，并用随机森林算法(RF)对这些特征进行分类, 达到良好的分类效果。但这样存在问题：经过我们的研究发现，由于在一开始使用peepdf进行解析的时候，只有一半的文件可以被解析到。所以我们重新选取了解析器mimicus[2]，这个工具可以解决之前因为结构缺陷或混淆而不能正常解析的问题，我们使用了mimicus 对之前的文件进行解析，发现对所有的数据（20万）均能正常解析，并做特征提取。

在模型2的训练中，我们先使用在总的数据集中随机抽选的平衡数据集进行训练与预测，其中包含2万恶意样本与2万良性样本。并且从Model2 开始我们就使用mimicus 对文件进行特征提取，一共提取特征135个。我们的主要算法还是使用准确率较高的随机森林。经过参数调优后，Model2在多次测试中的检测率有所提高并稳定在99.99%，误报率降低为0.012%。

我们用十万级别的样本重新对模型进行训练。在4核4G的CPU上，训练时间仅需要56s。使用2万数据集进行测试，Model2准确率维持在99.81%，误报率为0.086%。

During the experiment, we have updated the model for four times. The initial one (Model 1) uses peepdf as the parser to extract features. After several computation and quantization, these features can be used for machine learning-based training and prediction. We have extracted 133 features which contain static attributes of structure (count\_font、size、count\_startxref), content (title\_oth、subject\_lc) and metadata(producer\_oth、producer\_len). We classify these extracted features via Random Forests and find a good output. But problems also exist: only half of the files can be parsed if we use peepdf. Therefore, we reselect another parser mimicus[2] which can help us tackle the failure of parsing caused by structure defect or obscuration. By using mimicus, all of the data can be parsed and features can be extracted.

For the training of Model 2, we initially use balanced dataset for training and prediction. This balanced dataset includes 20,000 malicious and 20,000 benign samples which are selected randomly from the whole dataset. And we use mimicus to extract features from the beginning of training of Model 2 and we extract 135 features in total. We mainly apply Random Forest Algorithm as it brings higher accuracy. After parameter tuning, the detection rate of Model 2 in multiple times of testing can increase and maintain at 99.99%, with a decreased FP rate of 0.012%.

We use a new sample dataset at ten-thousand level to retrain the model, with a training time being only 56 seconds by utilizing a quad core with 4G RAM. Then we test the Model 2 by using 20,000 samples, the accuracy rate still maintains at 99.81% with a FP rate of 0.086%.