静态和动态相结合的病毒检测方法研究

文 / 黄馥妃　苗春雨

传统的计算机病毒检测方法主要有长度检测法、病毒签名检测法、特征代码检测法、校验法、行为监测法等。这些方法依据的原理不同，大体上分为病毒静态检测和动态检测两类，这两类计算机病毒的动态检测与静态检测各有其优缺点。但随着计算机病毒技术的不断更新，使传统的病毒检测技术已经无法有效地检测已知病毒的变种或未知病毒，由此笔者想到能否将二者结合起来，扬长避短，以实现病毒的高效检测。本文引入Dempster-Shafer证据理论用于融合病毒动态检测器和静态检测器的结果，实现计算机病毒的自动检测。

D-S证据理论由Dempster于1967年提出，其后Share将其发展并整理成一套完整的数学推理理论。D-S证据理论可以看作是有限域上对经典概率推理理论的一般化扩展，其主要特性是支持描述不同等级的精确度和直接引入了对未知不确定性的描述。D-S证据理论可以支持概率推理、诊断、风险分析以及决策支持等，并在多传感器网络、医疗诊断等应用领域内得到了具体应用。

一、病毒检测引擎

本文提出的病毒检测系统中采用多个支持向量机作为成员分类器对病毒的动态行为建模，使用多个概率神经网络作为成员分类器对病毒的静态行为建模，最后将各成员分类器的检测结果用D-S证据理论组合，形成最终的检测结论。

1.特征向量的提取

（1）抽取程序的N-gram信息。N-gram分析因其直观和易于实现，已成功地应用于语言模型和语音识别领域。将N-gram用于恶意代码分析在相关文献中也有提及，随着病毒代码数量剧增，很多病毒作者采用自动工具编写和编译恶意代码。N-gram分析是采用概率统计方法从N-gram集中挖掘出隐含其间的特征，可以用于检测出编制病毒所用的自动生产机、编译器和编程环境甚至程序作者的某些编程习惯，该法可以有效地防范病毒作者的反击。将N-gram分析应用于对Windows PE格式的程序文件的检测，其字母空间为{0，1}。实验中将滑动窗口大小取为16bit。对数据集进行预处理，可以获得的频率矩阵，见表1。

表1　频率矩阵

sample	character					
	0180	2C65	5080	635D	E010	…
virus 1	3	5	20	7	79	…
virus 2	40	6	42	23	7	…
virus 3	11	10	19	7	15	…
virus 4	7	0	13	9	22	…
virus 5	8	21	15	5	27	…
…	…	…	…	…	…	…

（2）基于API调用的特征提取。笔者将程序在加载的动态链接库((Dynamic Link Library，DLL)中调用的API函数作为待检测程序的分类特征。对样本库中的程序进行API函数调用跟踪处理后，可以得到大量的系统调用，这些不同的API函数对于病毒识别所起的作用是不一样的。当一个API函数调用在病毒文件中出现的频率非常高，而在正常程序文件中出现频率较低时，该API函数调用对识别病毒所作的贡献就比较大。因此尽量提取此类API构成特征集。在此，笔者采用概率统计学中的均方差法来进行实验。类间频率均方差能较好地体现各不同的API函数调用的“贡献程度”，一个被调用API函数的分类作用与它的类间频率均方差成正比。

2.系统框架

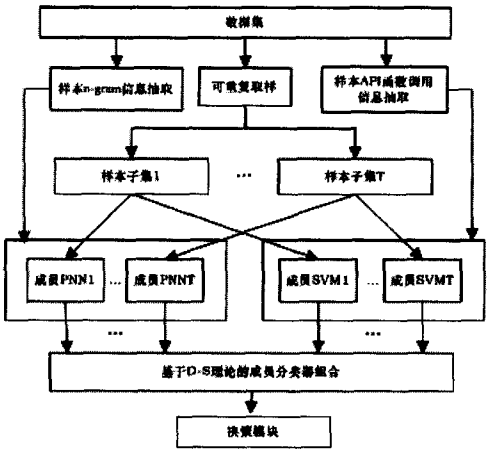

图1 系统框架图

基于D-S证据理论的病毒检测系统框架如图1所示。系统综合考虑程序的动态行为特性与静态特性，提取了两类特征向量来表示样程序的模式，一类是被程序所引用的API函数，另一类是从PE程序中静态抽取的N-gram信息。检测引擎监视程序的行为并进行分析，能有效地检测未知病毒和各种多态病毒。静态分析过程采用概率统计方法从N-gram集中挖掘出隐含其间的信息，可以用于检测出编制病毒所用的自动生产机、编译器和编程环境甚至程序作者的某些编程习惯，并且还可有效地防范病毒作者的反击。本系统中的成员分类器有概率神经网络（Probabilistic Neural Network，PNN）和支持向量机（Support Vector Machine，SVM）两种，PNN是径向基网络的一种变化形式，具有结构简单、训练快捷等特点，应用广泛，特别适合于模式分类问题的解决。它的优势在于可以利用线性学习算法来完成以往非线性算法所做的工作，同时又可以保持非线性算法高精度的特性。PNN的特点是人为调节的参数少，只需要通过SPREAD这一个参数来调节，且网络的学习基本依赖于样本。而统计理论中的SVM算法主要用于解决有限样本学习问题，而且对数据的维数和多变性不敏感，具有较好的分类精度和泛化能力。SVM方法已被成功用于孤立的手写体识别、文本分类、人脸识别、垃圾邮件过滤、入侵检测等领域并显示了巨大的优越性。最后使用Bagging算法生成参与集成的成员分类器，成员分类器的组合方法则基于Dempster-Shafer证据理论。在训练各成员分类器过程中输入PNN网络的特征量为API函数调用信息，输入SVM的特征量是程序的N-gram信息，增大了成员分类器间的差异性和不相关性。

3.实现方法

从检测系统的时间开销角度考虑，选择Bagging方法生成个体成员分类器，Bagging方法的基础是可重复取样从原始训练集中随机抽取的若干示例来训练个体成员分类器。通过重复选取训练集增加了个体分类器的差异度，从而提高泛化能力，而且该方法成员分类器的训练可以并行进行，具体做法如下：

给定训练集S，通过可重复取样获得一系列子训练集$S1$，$S2$，…，Sn。然后在各个子训练集上用信息增益算法挑选出对分类起重要作用的静态属性作为输入PNN网络训练出个体成员PNN网络。同时挑选出对分类起重要作用的动态属性输入SVM，训练出个体成员SVM分类器。最后，对于成员分类器的组合，选择D-S证据理论进行融合，并与投票法的结果进行了比较。算法伪代码如下：

A训练过程

输入：训练集=$S=\{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$其中$x_i \in X, y_i \in Y=\{1, 2, \cdots k\}=k$；学习机$C^{PNN}$, C^{SVM}；迭代次数T；每Bag的大小d。

1. W_{IG}=IG_Ngram(S_i)

2. W_{API}=MSE_API(S_i)

3. *for* $r=1, 2\cdots$, $T\{$

4. $S_i \leftarrow$从S中随机重采样d个样本；

5. $C_i^{PNN} \leftarrow C^{PNN}(S_i, W_{IG})$;//利用$S_i$对$C^{PNN}$进行学习，训练基于PNN神经网络的成员分类器

6. $C_i^{SVM} \leftarrow C^{SVM}(S_i, W_{API})$;//利用$S_i$对$C^{SVM}$进行学习，训练基于SVM的成员分类器

}

输出：C^{PNN}, C^{SVM}

B测试过程

输入：测试样本x。

输出：$E(x) = \underset{i \in K}{\operatorname{argmax}}\{\mathrm{Bel}(\theta_i) \mid \forall i, \mathrm{bel}(\theta_i^-) \leqslant \alpha\}$

式中$0<\alpha<1$，其目的是在低错误率约束下达到尽可能高的识别率。

二、成员分类器组合

在证据理论中概率信度函数的确定是以后推理的基本前提，对于多分类器组合系统，可以将各个成员分类器的分类性能结果作为各成员的基本信度函数值。可以选择类间距离作为各成员分类器的证据信度分配依据。

考虑N个分类器…应用于K分类问题，每一类用，$k=1$，2，…，K表示，在D-S证据理论下的识别框架为…设…的训练样本矩阵，将不同分类器的特征选择模块提取的特征矩阵记为…，在此各分类器可以是同构的或异构的，不同分类器的特征空间可以不同。为表述方便，对各分类器在不同特征空间下对样本的表示抽象为一个建模函数，记为：

$$\Gamma^{(n)}(X_k) = I_k^n, k = 1,2,\cdots,K, n = 1,2,\cdots,N.$$

当然，根据分类器的原理与方法不同，各建模函数也各异。

对于一测试样本x，各成员分类器对其建模表示为：

$$\Gamma^{(n)}(x) = \gamma^{(n)}, n = 1,2,\cdots,N.$$

现在根据对训练样本和测试样本的不同表示，每一个分类器$e^{(n)}$均可计算出测试样本与不同类型训练样本间的距离，并将其归一化，记为：

$$\mathrm{Distance}^{(n)}(I_k^n, \gamma^{(n)}), k = 1,2,\cdots,K$$

对于任一分类器$e^{(n)}$，可以得到K个类间距离值，记为：

$$d^{(n)} = [d_1^{(n)} d_2^{(n)} \cdots d_k^{(n)}]^T$$

若分类器$e^{(n)}$对测试样例x分类结果为θ_j，其信度(BBA)赋值根据下式计算：

$$m^{(n)}(\theta_j) = \mathrm{logsig}(\mathrm{variance}[d^{(n)}])$$

式中variance[$d^{(n)}$]为类间距离方差，logsig(　)是一个递增S形函数，将变量值映射到[0，1]。然后根据Dempster规则将各成员分类器BPA组合：

$$m(\theta_k) = m^{(1)} \oplus m^{(2)} \oplus \cdots \oplus m^{(N)}(\theta_k)$$

最终组合分类器决策规则为：

$$\theta_j = \underset{k}{\operatorname{argmax}}(m(\theta_k))$$

算法步骤有如下六步。

第一步，用训练数据对N个成员分类器进行训练。

第二步，各成员分类器对测试样例x建模。

第三步，各成员分类器计算类间距离。

第四步，对每一个成员分类器进行信度赋值。

第五步，通过Dempster组合规则对各分类器信度组合。

第六步，应用组合分类器决策规则将测试样例x分类；

三、实验结果与分析

本文的实验数据由从Vx.netlux.org下载的743个病毒文件和568个从新装的微软WindowsXP系统目录下的所有PE文件组成，把其中20%作为测试数据，80%作为训练数据。实验主要关注病毒代码被识别为正常程序的数量(False Negative，FN)和正常程序被识别为病毒代码的数量(False Positive，FP)。支持向量机的两个主要参数核函数和错分惩罚因子C的选择，选用了RBF核函数，C和Y的取值为2与0.015。而PNN中可调参数只有一个SPREAD，易于确定。实验中我们比较了病毒静态检测方法、动态检测方法以及将二者综合起来的检测方法的性能，结果见表2。

表2　实验结果对比表　　单位：%

Detection Methods	Detection Precision	FN	FP
dynamic detection	89	4.8	6.2
static detection	87	6.7	5.3
combined detection	93	3.4	3.6

结论：病毒动态检测与静态检测相结合方法的检测准确率比仅使用单一的动态或静态检测方法得到的准确率高。其原因在于输入集成分类器的特征量分别为程序的API信息和N-gram信息，它们之间无相关性，这就最大程度地扩大了各成员分类器的差异性，因而能提高检测系统的性能。