# 基于模糊模式与决策树融合的脚本病毒检测算法

张 涛 张 瀚* 付垒朋
(南开大学天津市智能机器人技术重点实验室 天津 300071)

**摘 要**：构建决策树进行脚本病毒检测可以全面利用训练样本的信息，在样本特征较为复杂、样本数较大的情况下会产生大量节点，计算时间复杂度高，在剪枝过程中影响分类准确度。为融合模糊模式的信息以提高分类器性能，该文设计了决策树分类基础上的融合算法。该算法将关于模糊模式贴近度的3个特性作为决策树样本信息向量中的属性。使用训练样本集，根据上述属性在划分点上的分裂信息值及信息增益率选择分裂属性，逐步构建决策树。实验结果验证了算法的稳定性与准确度，表明这种融合方法可增加属性的区分度，减少决策树的分支数。

**关键词**：脚本病毒检测；模糊模式；决策树；贴近度
**中图分类号**：TP309.5; TP181 **文献标识码**：A **文章编号**：1009-5896(2014)01-0108-06
**DOI**: 10.3724/SP.J.1146.2012.01491

# Script Virus Detection Algorithm Based on Fusion of Fuzzy Pattern and Decision Tree

Zhang Tao Zhang Han Fu Lei-peng
(*Tianjin Key Laboratory of Intelligent Robotics*, *Nankai University*, *Tianjin* 300071, *China*)

**Abstract**: The method using the decision tree for script virus detection can make full use of the information of training samples. But complex sample features and large number of samples will produce large number of nodes which result in the high algorithm time complexity and affect the classification accuracy due to the pruning process. In order to improve classification performance, a fusion algorithm using the information of fuzzy pattern is designed based on the decision tree classification algorithm. Three important characteristics of fuzzy pattern about close degree are regarded as the three attributes of sample information vector in the decision tree to build decision tree through training get. The stability and accuracy of the algorithm is verified by experiment. The experiment results show that the proposed algorithm increases discrimination of attributes and reduces the decision tree branch.

**Key words**: Script virus detection; Fuzzy pattern; Decision tree; Close degree

## 1 引言

脚本病毒[1]作为网络中的流行病毒对信息安全产生巨大威胁，脚本病毒检测技术的研究有着积极的意义。脚本与浏览器的紧密结合，简单的脚本语法使得病毒易于变形与撰写。特征代码法、校验和法等对于变种病毒检测效果不好[2]，使得统计与人工智能方法被广泛应用。

脚本病毒与可执行文件病毒等在语言与执行环境上有明显区别，脚本病毒研究有其自身特点。Choi 等人[3]采用 N-gram 分析脚本代码，结合 SVM 方法识别未知脚本病毒。Asaf 等人[4]使用机器学习理论构建分类器，通过对静态特征的分析检测恶意代码。Robert 等人[5]总结了机器学习方法提取恶意代码静态特征的研究进展，模糊理论[6]被应用到病毒检测中，可以识别未知的计算机病毒。文献[7]细致讨论了脚本病毒检测中，模糊模式的具体构造算法与参数选取。模糊模式的使用能提取病毒集的特征相互之间的规则信息，一定程度上能反映脚本语言编写特点，但是仅使用模糊模式检测病毒的效果不太理想。

决策树[8,9]是模式分类中的常用技术，文献[10]使用决策树分析可执行文件病毒的机器码。决策树算法通常[8,9,11]具有较高的准确率，分类效果与训练样本有直接关系。如果训练样本包含噪声数据过多，特征过于复杂，或统计出现波动，使得增加了不必要的分支，会产生决策树的过度适应现象，这样分支的增多会导致资源消耗的增加。

针对决策树算法的不完善及模糊模式在脚本检

2012-11-16 收到，2013-08-12 改回
国家自然科学基金(61004086)和高等学校博士学科点专项科研基金(200800551024)资助课题
*通信作者：张瀚 zhanghan@nankai.edu.cn

测中的缺点，本文提出了一种基于模糊模式和决策树相融合的脚本病毒检测算法。决策树使用脚本文件集预处理产生的特征信息训练生成；为了增加属性区分度，减少树节点个数，构建决策树时融合了模糊模式中贴近度的信息；混合使用模糊模式信息与自身特征信息可提高决策树的稳定性，改善决策树的准确性，所得决策树为新的脚本病毒分类器，最后通过实验结果验证检测效果。

## 2　脚本文件集的来源与预处理

实验对象为VBScript脚本构成的病毒脚本集与正常脚本集，通过 VirusTotal, VirusScan 标记。其中病毒脚本 1277 个，来源为 VX Heavens[12]网站下载的病毒实例与脚本病毒生成器生成的典型病毒实例，及知名安全网站看雪、剑盟、卡饭、http://malwareurl.com 等。病毒脚本的种类涉及后门程序、自动运行病毒、邮件蠕虫、漏洞攻击代码、P2P 蠕虫、局域网蠕虫、WORD 宏病毒、EXCEL 宏病毒、加密变型病毒等。正常脚本集 4041 个，由版本 1.15.4 的 Heritrix 工具爬取网页获得，接近现实网络中的分布，包括网站的特效脚本代码、word 宏、excel 宏和 http://www.computerperformance.co.uk/vbscript/网站的示例代码。

脚本病毒的行为主要通过调用如 CreateObject, CreateObject 的关键的系统函数和如 FileSystem Object 的对象来实现，Zou 等人[13]使用它们来表示病毒行为。文中这两类作为关键词被提取，对病毒脚本集和正常脚本集进行关键词的统计分析[7]，按其在病毒集和正常集中出现频率的均方误差大小排序，本文选取均方误差显著较大的关键词。样本信息由矩阵描述，行表示某样本的向量信息，元素为样本的属性取值，即关键词的出现频率。

## 3　模糊模式与决策树算法融合的检测算法

由文献[7]可知基于模糊模式的脚本病毒检测算法稳定性好，但对病毒文件检测正确率不太理想。基于决策树算法的脚本病毒检测算法分类效果同训练样本有关，全面的样本集会带来较高的准确率；但如果训练样本噪声过多，统计出现波动，都会出现决策树过度适应现象，产生的多余分支会使得分类出现错误。在决策树修剪环节中，决策树过于复杂会增加难度。精确的决策树需要较多的节点，训练建树时间较长，将模糊模式和决策树相融合，模糊模式的使用可增加属性的区分度，减少决策树的分支数，即减小计算时间复杂度，增加检测的稳定性。因此，本文设计了利用模糊贴近度信息的融合算法。

### 3.1 模糊化与贴近度的计算

设第 $i$ 个关键词 $K_i$ 在病毒脚本模型中出现的频度均值为 $\mathrm{E}(\mathrm{V}_i)$，在正常脚本模型中出现的频度均值为 $\mathrm{E}(\mathrm{N}_i)$。根据文献[7]，模糊化与贴近度按如下步骤计算：

步骤 1　使用训练集样本，选择文献[7]实验中检测率较高的正态偏大型隶属函数[14] $\mu(x)=1-e^{-k\sqrt{d_i}x^2}$，$k$ 为可以调整的系数，$d_i$ 为第 $i$ 个关键词在病毒模式和正常模式中频率之差的绝对值。可得病毒脚本集模糊模式 $\boldsymbol{V}'=\{\mu_{\mathrm{V}}(\mathrm{E}(\mathrm{V}_1)),\mu_{\mathrm{V}}(\mathrm{E}(\mathrm{V}_2)),\cdots,\mu_{\mathrm{V}}(\mathrm{E}(\mathrm{V}_n))\}$ 与正常脚本集模糊模式 $\boldsymbol{N}'=\{\mu_{\mathrm{N}}(\mathrm{E}(\mathrm{N}_1)),\mu_{\mathrm{N}}(\mathrm{E}(\mathrm{N}_2)),\cdots,\mu_{\mathrm{N}}(\mathrm{E}(\mathrm{N}_n))\}$；

步骤 2　当有某个待测样本时，由关键词频度得此样本的模糊模式 $\boldsymbol{M}'=\{\mu_1,\mu_2,\cdots,\mu_n\}$，按式(1)和式(2)计算此样本模式同病毒模式的欧氏贴近度 $d_{\mathrm{V}}$ 与待检测模式同正常模式的贴近度 $d_{\mathrm{N}}$，其中 $\mu_{\mathrm{V}_i}=\mu_{\mathrm{V}}(\mathrm{E}(\mathrm{V}_i)),\mu_{\mathrm{N}_i}=\mu_{\mathrm{N}}(\mathrm{E}(\mathrm{N}_i)),i=1,2,\cdots,n$。

$$d_{\mathrm{V}}=1-\sqrt{\left[(\mu_1-\mu_{\mathrm{V}_1})^2+(\mu_2-\mu_{\mathrm{V}_2})^2+\cdots+(\mu_n-\mu_{\mathrm{V}_n})^2\right]/n} \tag{1}$$

$$d_{\mathrm{N}}=1-\sqrt{\left[(\mu_1-\mu_{\mathrm{N}_1})^2+(\mu_2-\mu_{\mathrm{N}_2})^2+\cdots+(\mu_n-\mu_{\mathrm{N}_n})^2\right]/n} \tag{2}$$

### 3.2 融合算法

**3.2.1 训练算法描述**　隶属度函数中的参数 $k$ 根据文献[7]取实验效果较好的 0.7。在训练集预处理之后，分别构建病毒脚本集和正常脚本集的模糊模式，计算样本有关贴近度的 3 个值作为样本信息向量中的 3 个属性：样本同病毒模式的贴近度 $d_{\mathrm{V}}$，样本同正常模式的贴近度 $d_{\mathrm{N}}$ 和贴近度比值 $d_{\mathrm{r}}=d_{\mathrm{V}}/d_{\mathrm{N}}$。样本的信息向量的维数变成了 $n+3$，即增加了模糊模式中待检测样本对正常与病毒两类集合的距离信息。训练算法主要步骤描述如下：

步骤 1　计算训练样本集中正常文件和病毒文件关键词平均频率信息 $\{\mathrm{E}(\mathrm{N}_1),\mathrm{E}(\mathrm{N}_2),\cdots,\mathrm{E}(\mathrm{N}_n)\}$ 与 $\{\mathrm{E}(\mathrm{V}_1),\mathrm{E}(\mathrm{V}_2),\cdots,\mathrm{E}(\mathrm{V}_n)\}$；

步骤 2　由选定的隶属度函数生成病毒文件模糊模式和正常文件模糊模式，计算所有样本的 3 个模糊特征值：样本同病毒模糊模式的贴近度 $d_{\mathrm{V}}$，样本同正常模糊模式的贴近度 $d_{\mathrm{N}}$，样本的贴近度值之比 $d_{\mathrm{r}}=d_{\mathrm{V}}/d_{\mathrm{N}}$；

步骤 3　根据样本集中样本的 $n+3$ 维文件标识向量 $\{p_{K_1},p_{K_2},\cdots,p_{K_n},d_{\mathrm{V}},d_{\mathrm{N}},d_{\mathrm{r}}\}$ 取值组成的矩阵数据，构建决策树，决策树构建的子算法描述见 3.2.2 节。

**3.2.2 决策树构建子算法描述** 依据文件预处理后的关键词及其模糊属性信息生成 C4.5 决策树[8,9,11]，并对决策树进行悲观剪枝[15]，以控制生成决策树规模，提高准确率，算法主要步骤如下：

步骤 1 按式(3)计算出包含病毒集和正常集两部分的样本集合 $S$ 的信息熵，

$$\mathrm{I}(S)=-(N_{\mathrm{N}}/M)\log_2(N_{\mathrm{N}}/M)-(N_{\mathrm{V}}/M)\log_2(N_{\mathrm{V}}/M) \tag{3}$$

其中 $N_{\mathrm{V}}$ 为病毒个数，$N_{\mathrm{N}}$ 为正常文件个数，$M=N_{\mathrm{N}}+N_{\mathrm{V}}$；

步骤 2 每个样本向量包含 $n$ 个连续型属性 $K_i$，$i=1,\cdots,n$。对于 $K_i$，使用 $X$ 个划分点等量划分属性 $K_i$，按式(4)，式(5)，式(6)和式(7)计算 $X$ 个划分点 $b_j, j=1,\cdots,X$ 的信息熵、信息增益、分裂信息值、信息增益率。$b_j$ 将数据集划分为 $S_1$ 和 $S_2$，数量为 $M_1, M_2$，集合 $S_1$ 与 $S_2$ 中病毒和正常文件数量分别为 $M_{1\mathrm{V}}$，$M_{1\mathrm{N}}$ 与 $M_{2\mathrm{V}}$，$M_{2\mathrm{N}}$，其中 $M_1=M_{1\mathrm{V}}+M_{1\mathrm{N}}$，$M_2=M_{2\mathrm{V}}+M_{2\mathrm{N}}$。选取该属性所有划分点最大的信息增益率以及对应划分点为分裂节点。

$$\begin{aligned}\mathrm{I}(b_j)=&\frac{M_1}{M}\left(-\frac{M_{1\mathrm{V}}}{M_1}\log_2\frac{M_{1\mathrm{V}}}{M_1}-\frac{M_{1\mathrm{N}}}{M_1}\log_2\frac{M_{1\mathrm{N}}}{M_1}\right)\\&+\frac{M_2}{M}\left(-\frac{M_{2\mathrm{V}}}{M_2}\log_2\frac{M_{2\mathrm{V}}}{M_2}-\frac{M_{2\mathrm{N}}}{M_2}\log_2\frac{M_{2\mathrm{N}}}{M_2}\right)\end{aligned} \tag{4}$$

$$\mathrm{G}(b_j)=\mathrm{I}(S)-\mathrm{I}_{K_i}(b_j) \tag{5}$$

$$\begin{aligned}\mathrm{D}(b_j)=&-(M_1/M)\log_2(M_1/M)\\&-(M_2/M)\log_2(M_2/M)\end{aligned} \tag{6}$$

$$\mathrm{R}(b_j)=\mathrm{G}(b_j)\big/\mathrm{D}(b_j) \tag{7}$$

步骤 3 对所有属性使用上述方法，选取具有最大信息增益率的属性作为此次分裂的属性，并将集合划分为两个子集；

步骤 4 对划分的两个子集使用上述方法递归进行决策树的生长；

步骤 5 设置叶节点，停止决策树的生长。以下 3 种情况设为叶节点：节点数据集的纯度达到 98%；节点数据集与整个数据集样本数比值小于 1%，即数据集过小；没有用来分割的属性。当该节点集合中病毒比重较大时，将此叶节点的类别设置病毒，反之设置为正常文件。

**3.2.3 分类算法描述**

步骤 1 计算待检测文件中关键词出现的频度 $\{p'_{K_1}, p'_{K_2}, \cdots, p'_{K_n}\}$ 与贴近度相关的 3 个模糊特征值 $d'_{\mathrm{V}}$，$d'_{\mathrm{N}}$，$d'_{\mathrm{r}}$，得 $n+3$ 维标识向量 $\{p'_{K_1}, p'_{K_2}, \cdots, p'_{K_n}, d'_{\mathrm{V}}, d'_{\mathrm{N}}, d'_{\mathrm{r}}\}$。

步骤 2 使用构建好的决策树自根节点向下进行检测，设决策树的根节点为 $K_i$，其分裂点的值为 $Y$，若对应的 $p'_{K_i}>Y$，则将该向量放入其右子树中递归向下判断，反之放入左子树，直到该向量落入叶节点。如果叶节点类别为病毒则为病毒文件，反之为正常文件。

## 4 实验结果分析

### 4.1 融合算法的实验结果

使用 10 次交叉验证实验方式[16]，共 5318 个样本。每次随机提取 1772 个实验样本作为测试数据集，其中病毒文件 425 个，正常文件 1347 个，其余 3546 个样本作为训练集。决策树停止生长的条件设置为：最小的叶准确率为 98%，最下的叶分支比例为 0.001，连续属性的划分数设置为 30。

样本集中病毒集与正常集数量比例为1:3.16，为非平衡数据集。文中采用如下性能指标：TP 正常文件判断为正常的数量；TN 病毒文件判断为病毒的数量；漏报数 FP 是病毒文件判断为正常的数量；虚警数 FN 是正常文件判断为病毒的数量；漏报率 $\mathrm{FP}_{\mathrm{rate}}=\mathrm{FP}/(\mathrm{TN}+\mathrm{FP})$；虚警率 $\mathrm{FN}_{\mathrm{rate}}=\mathrm{FN}/(\mathrm{TP}+\mathrm{FN})$；正常文件判别正确率 $\mathrm{TP}_{\mathrm{rate}}=\mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$；病毒文件判别正确率 $\mathrm{TN}_{\mathrm{rate}}=\mathrm{TN}/(\mathrm{TN}+\mathrm{FP})$；评价非平衡数据集中总体分类性能的几何均值 $G_{-\mathrm{mean}}=\sqrt{[\mathrm{TN}/(\mathrm{TN}+\mathrm{FP})]\times[\mathrm{TP}/(\mathrm{TP}+\mathrm{FN})]}$；评价非平衡数据集中病毒类分类性能的 $F_{\beta-\mathrm{measure}}=(1+\beta^2)P\times R/(P+\beta^2R)$，是病毒文件判别正确率即召回率 $R=\mathrm{TN}/(\mathrm{TN}+\mathrm{FP})$ 与查准率 $P=\mathrm{TN}/(\mathrm{TN}+\mathrm{FN})$ 的函数，常取 $\beta=1$，此时 $F_{1-\mathrm{measure}}=2P\times R/(P+R)$；后 4 个指标尤为重要。

首先确定在本类型样本集下实验所需最少的关键词数量，也即属性数。考察融合算法中属性数与实验正确率的关系，对基础的 $\mathrm{TP}_{\mathrm{rate}}$ 与 $\mathrm{TN}_{\mathrm{rate}}$ 进行研究。由图 1 可见，在属性数达到 53 个之后，正常文件判别正确率与病毒文件判别正确率都趋于平稳，分别为 0.9636 与 0.9694 附近，再增加属性数已不能提高最重要的病毒文件判别正确率，实验中选取 53 个关键词。

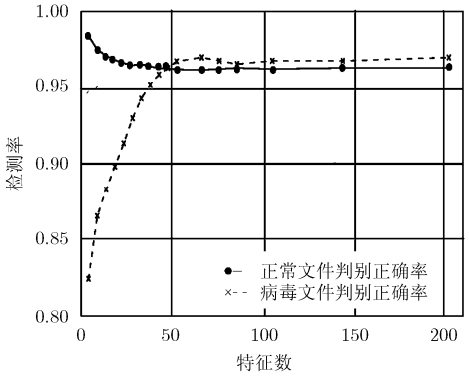


图1 融合算法中关键词数与实验正确率的关系图

表 1 为只使用决策树的检测模型实验结果，正常文件平均检测率为 0.9584，病毒文件为 0.9506，效果较好。代表非平衡样本集总分类效果的$G_{-\mathrm{mean}}$值为 0.9545，代表病毒类分类效果的$F_{1-\mathrm{measure}}$值为 0.9130，效果较好。缺点是生成的节点数较多，平均约为 47 个，决策树检测方法在达到一定准确率的要求下，需要对属性细致划分以致生成较多的节点，实验中表现为以节点数量换取高检测率的决策树。而决策树训练学习的时间复杂度与决策树的节点数呈线性正相关[8]，节点数量体现了病毒分类问题的计算时间复杂度。

表 2 为融合算法实验结果，正常文件的检测率为 0.9638；病毒文件的检测率为 0.9680；漏报率$\mathrm{FP}_{\mathrm{rate}}$ 10 次交叉均值为 0.0320，大大低于模糊模式的 0.1774，也低于决策树的 0.0494；虚警率$\mathrm{FN}_{\mathrm{rate}}$均值为 0.0362，已经很低，低于决策树算法的 0.0416，但是高于模糊模式算法的 0.0126，原因在于模糊模式的构造倾向于判定样本为正常，虚警率的微幅提高是以大幅降低最具价值的病毒检测率为代价；$G_{-\mathrm{mean}}$ 值为 0.9659；$F_{1-\mathrm{measure}}$ 值为 0.9295，非平衡集整体分类效果与病毒分类效果均有提高；生成的节点数较少，均值约为 13 个。

### 4.2 3 种检测模型的对比分析

**4.2.1 正常文件的检测率**　由表 1，表 2 将融合算法与决策树算法、文献[7]的模糊模式算法比较，正常文件检测率均值依次为 0.9638, 0.9584, 0.9875，总体样本差依次为 0.0011, 0.0024, 0.0020。模糊模式算法对于正常文件检测效果稍强于其它两个检测模型，因模糊模式算法构造时倾向于判断为正常文件。

**4.2.2 病毒文件的检测率与$F_{1-\mathrm{measure}}$值**　由表 1，表 2 融合算法对病毒文件的检测率均值 0.9680 要高于

**表 1 决策树模式 10 次交叉实验结果**

| 序号 | TP | TN | FP | FN | $\mathrm{TP}_{\mathrm{rate}}$ | $\mathrm{TN}_{\mathrm{rate}}$ | $\mathrm{FP}_{\mathrm{rate}}$ | $\mathrm{FN}_{\mathrm{rate}}$ | 节点数 | $G_{-\mathrm{mean}}$ | $F_{1-\mathrm{measure}}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1292 | 399 | 26 | 55 | 0.9592 | 0.9388 | 0.0612 | 0.0408 | 44 | 0.9489 | 0.9078 |
| 2 | 1289 | 396 | 29 | 58 | 0.9569 | 0.9318 | 0.0682 | 0.0431 | 43 | 0.9443 | 0.9010 |
| 3 | 1286 | 402 | 23 | 61 | 0.9547 | 0.9459 | 0.0541 | 0.0453 | 46 | 0.9503 | 0.9054 |
| 4 | 1287 | 416 | 9 | 60 | 0.9555 | 0.9788 | 0.0212 | 0.0445 | 43 | 0.9671 | 0.9234 |
| 5 | 1289 | 406 | 19 | 58 | 0.9569 | 0.9553 | 0.0447 | 0.0431 | 47 | 0.9561 | 0.9134 |
| 6 | 1289 | 396 | 29 | 58 | 0.9569 | 0.9318 | 0.0682 | 0.0431 | 41 | 0.9443 | 0.9010 |
| 7 | 1295 | 412 | 13 | 52 | 0.9614 | 0.9694 | 0.0306 | 0.0386 | 53 | 0.9654 | 0.9269 |
| 8 | 1295 | 395 | 30 | 52 | 0.9614 | 0.9294 | 0.0706 | 0.0386 | 48 | 0.9453 | 0.9060 |
| 9 | 1294 | 415 | 10 | 53 | 0.9607 | 0.9765 | 0.0235 | 0.0393 | 55 | 0.9686 | 0.9295 |
| 10 | 1294 | 403 | 22 | 53 | 0.9607 | 0.9482 | 0.0518 | 0.0393 | 51 | 0.9544 | 0.9149 |
| 综合 | 12910 | 4040 | 210 | 560 | 0.9584 | 0.9506 | 0.0494 | 0.0416 | 47.1 | 0.9545 | 0.9130 |

**表 2 融合算法 10 次交叉实验结果**

| 序号 | TP | TN | FP | FN | $\mathrm{TP}_{\mathrm{rate}}$ | $\mathrm{TN}_{\mathrm{rate}}$ | $\mathrm{FP}_{\mathrm{rate}}$ | $\mathrm{FN}_{\mathrm{rate}}$ | 节点数 | $G_{-\mathrm{mean}}$ | $F_{1-\mathrm{measure}}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1301 | 408 | 17 | 46 | 0.9659 | 0.9600 | 0.0400 | 0.0341 | 12 | 0.9629 | 0.9283 |
| 2 | 1299 | 409 | 16 | 48 | 0.9644 | 0.9624 | 0.0376 | 0.0356 | 11 | 0.9634 | 0.9274 |
| 3 | 1298 | 411 | 14 | 49 | 0.9636 | 0.9671 | 0.0329 | 0.0364 | 14 | 0.9653 | 0.9288 |
| 4 | 1299 | 413 | 12 | 48 | 0.9644 | 0.9718 | 0.0282 | 0.0356 | 12 | 0.9681 | 0.9323 |
| 5 | 1297 | 414 | 11 | 50 | 0.9629 | 0.9741 | 0.0259 | 0.0371 | 10 | 0.9685 | 0.9314 |
| 6 | 1296 | 413 | 12 | 51 | 0.9621 | 0.9718 | 0.0282 | 0.0379 | 15 | 0.9669 | 0.9291 |
| 7 | 1298 | 411 | 14 | 49 | 0.9636 | 0.9671 | 0.0329 | 0.0364 | 11 | 0.9653 | 0.9288 |
| 8 | 1300 | 413 | 12 | 47 | 0.9651 | 0.9718 | 0.0282 | 0.0349 | 13 | 0.9684 | 0.9333 |
| 9 | 1297 | 414 | 11 | 50 | 0.9629 | 0.9741 | 0.0259 | 0.0371 | 16 | 0.9685 | 0.9314 |
| 10 | 1297 | 408 | 17 | 50 | 0.9629 | 0.9600 | 0.0400 | 0.0371 | 12 | 0.9614 | 0.9241 |
| 综合 | 12982 | 4114 | 136 | 488 | 0.9638 | 0.9680 | 0.0320 | 0.0362 | 12.6 | 0.9659 | 0.9295 |

决策树算法的 0.9506 和模糊模式算法的 0.8226，模糊模式算法尤其偏低，倾向于判断为正常文件的设计使其牺牲了更为重要的病毒检测率。三者检测率的总体标准差依次为 0.00527, 0.01779, 0.01047，融合算法的稳定性比决策树算法有明显提升，其标准差不到决策树算法的 1/3，而病毒文件的检测率、稳定性等指标在实际应用中尤为重要。

图 2 可见，对于非平衡样本集中体现分类器病毒类分类性能的 $F_{1-\text{measure}}$ 值，融合算法为 0.93，分类效果好于决策树算法和模糊模式算法，其中模糊模式算法的 $F_{1-\text{measure}}$ 均值仅为 0.88。融合、决策树、模糊三者 $F_{1-\text{measure}}$ 值的总体标准差依次为 0.0025, 0.0100, 0.0067，融合算法 $F_{1-\text{measure}}$ 值稳定性大幅提升，曲线平稳，其标准差仅为决策树方法的 1/4。

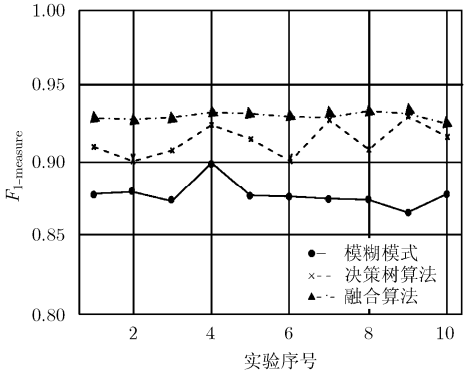


图 2 3 个模型 10 次交叉验证 $F_{1-\text{measure}}$ 值曲线图

**4.2.3 总体分类性能与 $G_{-\text{mean}}$ 值** 评价非平衡数据集中总体分类性能的 $G_{-\text{mean}}$ 值如图 3 所示，融合算法、决策树算法、模糊模式算法的 10 次交叉 $G_{-\text{mean}}$ 平均值分别为 0.966, 0.954, 0.901，其总体标准差依次为 0.0024, 0.0090, 0.0057，融合算法仍然最稳定，其标准差约为决策树方法的 1/4。可见融合算法的稳定性明显优于决策树算法，总体分类性能较模糊模式算法有大幅提升。由于融合算法结合了模糊模式信息，总体分类性能较决策树算法有了提高，稳定性也得到较大提升。

**4.2.4 决策树的节点个数** 两种决策树模型生成的结点数见表 1，表 2，决策树模型节点数在 47 个左右，节点数总体标准差为 4.41。融合算法检测方法需要节点数较少，仅为 13 个左右，融合模型节点数总体标准差为 1.8，生成节点数的稳定性较好，其生成的节点数仅为决策树模型的 25%左右。

### 4.3 融合算法的理论分析

决策树的显著优点是所给分类问题的计算时间复杂度与决策树的节点数成线性增长关系。决策树

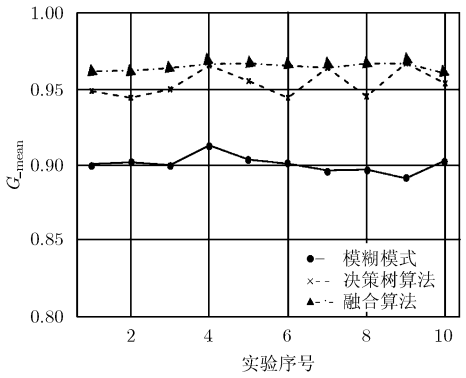


图 3 3 个模型 10 次交叉验证 $G_{-\text{mean}}$ 值曲线图

中某些重要模式由于噪声及数据波动可导致建树过程复杂，而由模糊模式提炼的较为本质的属性具有一定鲁棒性，一定程度上可抗噪声干扰，结合建树算法互为补充，在保证正确率的前提下，既可由模糊模式提高稳定性，也能在两者的融合下大幅度节省节点数。当样本足够多，种类全面时融合算法的优势会更加明显。

## 5 结束语

本文在决策树检测实验的基础上提出了一种基于模糊模式和决策树算法相融合的脚本病毒检测算法，通过实验结果分析，融合算法在脚本病毒检测中具有很好的检测效果与可行性，总体分类性能与病毒分类性能较模糊模型有明显提高，比单纯的决策树模型也有一定提高。这种模糊模式同决策树融合的具体方法可显著减少节点生成个数，增加了属性的区分度。它克服了模糊模式对病毒类的低检测率，优化了决策树的构建过程，较好地同时利用了模糊信息与自身的特征信息，且性能稳定，生成节点数较少。

本文提出的脚本病毒检测算法针对 VBScript，对于其它的脚本病毒的研究需提取有效特征再应用。寻求模糊模式和改进决策树的更有效的融合办法，是将来的工作方向之一。

## 参 考 文 献

[1] Willem D G, Dominique D, and Frank P. Better security and privacy for web browsers: a survey of techniques, and a new implementation[C]. 8th International Workshop on Formal Aspects of Security and Trust, Leuven, Belgium, 2011: 21–38.

[2] Kim J Y and Moon B R. New malware detection system using metric-based method and hybrid genetic algorithm[C]. Proceedings of the 14th International Conference on Genetic and Evolutionary Computation Companion, Philadelphia, PA, United States, 2012: 1527–1528.

[3] Choi J H, Choi C, Kim H Y, *et al.*. Efficient malicious code detection using N-gram analysis and SVM[C]. International Conference on Network-Based Information Systems, Tirana, Albania, 2011: 618–621.

[4] Asaf S, Robert M, Yuval E, *et al.*. Detection of malicious code by applying machine learning classifiers on static features: a state-of-the-art survey[J]. *Journal of Information Security and Applications*, 2009, 14(1): 16–29.

[5] Robert M and Yuval E. Malicious code detection using active learning[C]. Second ACM SIGKDD International Workshop, PinKDD 2008, Las Vegas, NV, USA, 2008: 74–91.

[6] Zadeh L A. The concept of a linguistic variable and its applications to approximate reasoning[J]. *Information Sciences*, 1975, 8(3): 199–249.

[7] 张涛, 付垒朋, 张瀚, 等. 一种基于模糊模式的脚本病毒检测方法[OL]. 中国科技论文在线, http://www.paper.edu.cn. 2011.9.

Zhang Tao, Fu Lei-peng, Zhang Han, *et al.*. Research on method of classification based on fuzzy pattern model in script virus detection[OL]. http://www.paper.edu.cn. 2011.9.

[8] Quinlan J R. C4.5 Programs for Machine Learning[M]. San Mateo, CA, Morgan Kaufmann Publishers, 1993, Chapter 1–Chapter 5.

[9] Thakur D, Markandaiah N, and Raj D S. Re optimization of ID3 and C4.5 decision tree[C]. International Conference on Computer and Communication Technology, Allahabad, India, 2010: 448–450.

[10] Rad B B, Masrom M, Suhaimi I, *et al.*. Morphed virus family classification based on opcodes statistical feature using decision tree [C]. The International Conference on Informatics Engineering & Information Science, University Technology Malaysia, Kuala Lumpur, Malaysia, 2013: 123–131.

[11] Fan Jie and Wen Pu. Application of C4.5 algorithm in web-based learning assessment system[C]. The Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, China, 2007: 4139–4143.

[12] VX Heavens. Virus collection[OL]. http://vx.netlux.org. 2009.10.

[13] Zou Meng-song, Han Lan-sheng, Liu Qi-wen, *et al.*. Behavior-based malicious executables detection by multi-class SVM[C]. Proceedings of IEEE Youth Conference on Information, Computing and Telecommunication, Piscataway, NJ, United States, 2009: 331–334.

[14] Nakamura E and Kehtarnavaz N. Optimization of fuzzy membership function parameters[C]. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Piscataway, NJ, United States, 1995: 1–6.

[15] Rajeev S and Kyuseok S. A decision tree that integrates building and pruning[C]. Proceedings of 24th International Conference on very Large Database, New York, USA, 1998: 404–415.

[16] Duda R O, Hart P E, and Stork D G. Pattern Classification [M]. 2nd Edition, New York: USA, Wiley, 2001, Chapter 8.

张　涛：　女，1983年生，硕士生，研究方向为信息安全.

张　瀚：　男，1978年生，男，博士，教授，博士生导师，研究方向为计算智能、信息安全、生物信息学.

付垒朋：　男，1983年生，硕士生，研究方向为信息安全.