一种基于人工免疫和代码相关性的计算机病毒特征提取方法

王 维 张鹏涛 谭 营 何新贵

(北京大学信息科学技术学院智能科学系 北京 100871)

(机器感知与智能教育部重点实验室 北京 100871)

摘 要 现有的计算机病毒检测方法利用病毒特征码来检测病毒,已经不能适应病毒技术的发展,特别是其无法检测出病毒的新变种与未知病毒.受自然免疫系统的启发,该文提出了一种基于人工免疫的利用计算机病毒代码相关性的计算机病毒特征提取方法.这种特征提取方法在底层提取出与病毒相关的字节模式,在相对更高的层面上记录这些字节模式之间的共同作用信息,之后利用阴性选择算法提取出计算机病毒检测基因库,实现了对训练集上合法程序的完美记忆,从而保证了该文方法的误判率处于极低的水平.计算机病毒检测基因库在个体层上存储病毒样本,一个样本中储存了若干个不定长的基因,充分利用了同一个样本的不同基因代码之间的相关性.为了尽可能少地丢失有效信息,这种方法在基因层上对基因进行匹配,在个体层上对可疑程序进行分析,最终由整个计算机病毒检测基因库做出分类决策.实验表明:此方法对未知病毒的平均识别率达到 94%,同时对合法程序的误判率保持在 2%之内,具有较强的泛化能力,能够有效识别病毒伪装,检测出已知病毒的新变种,对未知病毒也具有较强的识别能力.

关键词 病毒检测;人工免疫;特征提取;代码相关性;连续一致匹配

中图法分类号 TP18 **DOI 号**: 10.3724/SP.J.1016.2011.00204

A Feature Extraction Method of Computer Viruses Based on Artificial Immune and Code Relevance

WANG Wei ZHANG Peng-Tao TAN Ying HE Xin-Gui

(*Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, Beijing* 100871)

(*Key Laboratory of Machine Perception of Ministry of Education, Beijing* 100871)

Abstract Existing anti-virus methods make use of signatures to detect malicious codes. They are inefficient to detect various forms of computer viruses, especially new variants and unknown viruses. Inspired by biologic immune system, a novel artificial immune based signature extraction method is proposed. This method automatically identifies bit patterns that correlate with viruses using instruction frequency and file frequency, and then identifies higher-level genes that are associated with viruses, generating a detecting virus gene library using the negative selection algorithm which leads to a fairly low false positive rate compared with the traditional signature-based methods. The advantages of our proposed method are described as follows. In the feature extraction phase, the detecting virus gene library stores virus samples with variable number of variable length genes at individual level, and uses multiple genes coexistence in one virus to avoid the pos-

收稿日期:2010-01-07;最终修改稿收到日期:2010-07-12.本课题得到国家自然科学基金(60673020,60875080)和国家"八六三"高技术研究发展计划项目基金(2007AA01Z453)资助.**王 维**,男,1982 年生,博士研究生,主要研究方向为计算智能、数据挖掘、人工免疫系统及其在信息安全中的应用等. E-mail: weiwang@cis.pku.edu.cn. **张鹏涛**,男,1986 年生,博士研究生,主要研究方向为人工免疫系统、智能信息处理算法、计算机信息安全、模式识别等. **谭 营**(通信作者),男,1964 年生,博士,教授,博士生导师,主要研究领域为计算智能、机器学习、智能信息处理及其在信息安全中的应用等. E-mail: ytan@pku.edu.cn. **何新贵**,男,1938 年生,教授,博士生导师,中国工程院院士,主要研究领域为模糊逻辑、神经网络、进化计算、数据库理论.

sible loss of information considerably, fully taking the advantages of relevance between viral instructions within a virus program; in the classification phase, suspicious programs are analyzed at individual level in contrast to the existing gene matching technique. Experimental results indicate that the proposed method yields high detection rates for obfuscated viruses with an averaged recognition rate of 94% in real-world conditions, the false positive rate can be maintained below 2%. The method has a good generalization ability, and is able to effectively and efficiently detect new variants of known virus and unknown viruses.

Keywords　virus detection; artificial immune; feature extraction; code relevance; successive consistency matching

1　引　言

传统的计算机反病毒方法是以特征检测为基础的，这些方法利用从病毒中提取的特定特征来检测出有相似行为的病毒程序.它们对于已知或者是出现过的病毒有着很高的识别率，但是对于没有出现过的未知病毒或者病毒的新变种缺乏快速而准确的识别能力[1].各种先进的病毒技术应用多态或是变型的方式企图逃避基于特征的检测，常见的有插入冗余代码、代码位置调换、寄存器的重新组合和同义指令代换等[2].病毒的制造者们，针对了传统病毒的扫描器，通过类似方法很轻易地改写自己的代码，躲避了包含这些病毒传统特征的扫描.启发式的扫描器试图通过利用病毒代码更一般化的特点，诸如结构化或者行为化模式，来弥补这一缺陷[3].不过这个过程需要介入很多专家知识，而且建立出的模型常常在对未知病毒的高识别率和较低的正常文件误判率之间顾此失彼.

以生物体为原型的计算机系统和自然生物系统有着天然的联系，而自然免疫系统又具有强大的区分“自体”和“异体”的能力，这种功能与计算机安全系统的反病毒功能极为类似[4-5].因此，借助自然免疫机理，如阴性选择机理、克隆选择机理等机理，采用人工免疫模型来识别计算机中的合法程序(称为“自体”)和病毒程序(称为“异体”)成为病毒检测的一个可行的发展方向[6-7].许多人已经提出了应用于计算机安全方面的人工免疫模型，并在实际应用中取得了较好的结果，它可以自动无监督地完成自体和异体的区分，阴性选择原理能保持一个较低的程序误判率.但其中仍有某些方面并不完善[8].例如，泛化能力较差，只能用于检测少量的病毒，无法在大数据集上取得良好的表现；生成检测器时，采用随机产生再筛选的方法，无向导性，致使检测器生成效率低下；没有充分利用多个相关检测器间的相关性；检测器的长度常常人为根据经验设定，不能推广到更广泛的适用环境；另外，各种模型对病毒的识别率也不高，不能满足实际应用的要求.

在病毒的实际工作机理中，一个病毒的多个指令都是相关的，病毒多个关键代码的有机结合才产生了病毒作用.之前的研究成果往往不是特别关注于此，特征储存的方式通常是独立的.基于此思想，本文提出了一种新的特征提取方法，充分利用了组成病毒的相关指令的相关性，使得病毒特征的提取在个体层上完成，将每个病毒样本的多个指令存放在此病毒样本对应的数据库空间中，采用与其特征生成、储存对应的匹配检测模式，并由此建立了模型.

2　相关工作

病毒特征的提取并非一个新问题.Kephart 等人[9]通过运用少量已知病毒对大量文件的感染，之后提取 12～36 字节的不同常数定长区域，从中挑选出可以得到最低误判率的作为病毒的特征.虽然这种方法不需要专家的帮助就可能快速提取出病毒的特征，但是作者也都承认其无法适用于检测病毒在一定程度的多态化.其它有些检测方法还尝试使用诸如 win32 dll 调用、ASCII 字符串或者字节序列作为特征.Henchiri 和 Japkowicz 针对这些由于特定训练集产生的过拟合特征现象而提出了一种基于数据挖掘的病毒检测特征提取和评价模式[10]，这项工作关注于不同家族病毒的分种属特征，采用数据挖掘家族内部和外部区分开对待的方法来管理一个建立在病毒集上的详细特征搜索.

受免疫系统阴性选择机制启发，Forrest 等[11-15]提出了一种检测异常变化的阴性选择算法.该算法

识别自己和非己时,不需要参考非己的信息,特别适用于未知时变环境下的故障诊断和计算机安全监控等情况,所以应用其进行病毒检测是很合理的,有助于研究者用有限的知识来认识和解释未知情形.但是在这种阴性选择算法下,特征集合中的特征与"自体"的信息数目成指数关系.在很多情况下,如计算机系统中,"自体"的信息数目可以被看做是足够大的,此时覆盖整个"异体"空间所需的特征数目就会太大,因此难以在大规模数据中进行应用.此外,这种无向导地随机产生特征会进行大量的无用运算,消耗大量的时间.进一步,利用固定长度的字符串来标识个体,也存在一定的不合理性.文献[16]改进了阴性选择算法,给出了一种复杂度较低的随机选择的检测器生成算法,使特征数目和"自体"信息数目成正比例关系.这种方法使特征的数目大大减少,但是没有从根本上解决特征过多的问题.

在此基础上 Lee 等人提出了一个可以检测出未知病毒的基于人工免疫的病毒检测系统[17](VDS),使用基于病毒行为的检测方法从这些可疑程序中选出病毒程序.该模型利用已有知识来有向导地生成病毒特征(即病毒基因),是对阴性选择算法的一个改进,克服了阴性选择随机生成特征和只适用于稳定系统的缺点[18-19].从特征提取的角度来看,该模型的有效性受到特征提取区、比较单元大小、位置等因素的严重影响,并且一个病毒只有一个特征,因此特征长度通常会比较长,而特征长度太长时,在实际中是不可用的,往往特征的一个小平移都会致使特征失效.

Deng 等人简要分析了前人的工作,提出了特征提取需要着重研究的几个方面[2],具体为:使用变长特征来代替固定长度的特征;采用多个特征并存的方式来标识一个病毒,而不是只采用一个病毒特征;充分利用病毒的多个特征间的相关信息.Karnik 等人提出了一个基于余弦相似度度量程序特征的方法,用来识别文件的多态形式,并将其用于病毒检测[20-21].他们的工作对象是已知一个病毒代码的变种,期望以较高的概率检测出这个代码的任何伪装形式.这种针对代码调换伪装技术的特征度量方法对本文的特征匹配采用方式很有启发.

3 特征的提取方法

3.1 特征的有向导生成

阴性选择算法中,检测器集合中的检测器与"自体"的信息数目基本是成指数关系的.在很多情况下,如计算机系统中,"自体"的信息数目可以被看做是足够大的,此时覆盖整个"异体"空间所需的检测器数目就会太大,因此难以在大规模数据中进行应用.此外,这种无向导地随机产生检测器会进行大量的无用运算,消耗大量的时间.进一步,利用固定长度的字符串来标识个体,也存在一定的不合理性.即使有些方法对其特征提取进行优化,使检测器数目和"自体"信息数目成线性关系[22],可以使检测器的数目大大减少,但是仍然没有从根本上解决检测器过多的问题,并且仍然是无向导生成检测器.本文试图根据病毒特征在各个不同大小训练集中呈现出的显著性,通过阈值疏导或者截流,利用先验的信息最大程度地挖掘可用信息和计算代价的平衡.

生物体的遗传信息主要存放在 DNA 上,但并不是 DNA 上的所有片段都能表达遗传信息.基因才是具有遗传效应的 DNA 片段,而基因又是由若干个脱氧核苷酸(ODN)组成的.三者之间的关系如图 1 所示.

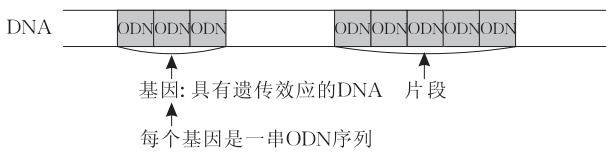

图 1 DNA、基因以及 ODN 关系

下面先简要解释一下本文中用到的名词.

(1) DNA. 整个程序的 bit 串称为程序的 DNA;

(2) 基因. 病毒的检测器,DNA 的片断,病毒检测的比较单元;

(3) 脱氧核苷酸. 每两个字节看作是一个脱氧核苷酸,记作 ODN,若干个脱氧核苷酸组成了基因.

病毒程序的代码对应着生物体中的 DNA.少量起着病毒作用的关键代码被认为是病毒的基因,这些基因由病毒的 ODNs 组成.多个 ODN 的有序连接表示程序的一个指令或多个指令的有序集合.本文采用滑动窗口的方法来计数 ODN.

某程序的一段 DNA 为

CD21 C307 1FCD 218C C0B8,

其中包含的 ODN:

CD21 21C3 C307 071F 1FCD CD21

218C 8CC0 C0B8.

病毒特征的初始选择采用了有向导的方式,利用已知浓度的信息(式(1))来统计每个 ODN 趋向于代表病毒的程度.

I_n是训练集中所有病毒的 ODN 总数;

I_s是训练集中所有合法程序的 ODN 总数；

I_n^i是训练集中 ODN i 在病毒中出现的次数；

I_s^i是训练集中 ODN i 在合法程序中出现的次数；

F_n是训练集中病毒文件数目；

F_s是训练集中合法程序数目；

F_n^i是训练集中含有 ODN i 的病毒文件数；

F_s^i是训练集中含有 ODN i 的合法程序数；

其中 $i\in[0,65535]$.

统计算法基本步骤如下：

1. 初始化 I_n、I_s、I_n^i、I_s^i、F_n^i、F_s^i为 0；

2. 选择一个合法程序，初始化标志数组 $flag[i]=0$；

3. 采用长度为 2 个字节的滑动窗口，读取窗口中的 ODN，计算其值 i，将此作为其索引；

4. I_s^i++，I_s++；

5. 如果 $flag[i]=0$，则 F_s^i++，并将 $flag[i]$置为 1，标记 ODN i 已经在该程序中出现过一次；

6. 滑动窗口向前滑动一个字节，返回步 3，直到该合法程序文件结束；

7. 返回步 2，直到训练集中所有合法程序统计完成；

8. 选择一个病毒程序，初始化标志数组 $flag[i]=0$；

9. 采用长度为 2 个字节的滑动窗口，读取窗口中的 ODN，计算其值 i，将此作为其索引；

10. I_n^i++，I_n++；

11. 如果 $flag[i]=0$，则 F_n^i++，并将 $flag[i]$置为 1，标记 ODN i 已经在该程序中出现过一次；

12. 滑动窗口向前滑动一个字节，返回步 3，直到该病毒程序文件结束；

13. 返回步 2，直到训练集中所有病毒程序统计完成.

由上面的程序，模型能够统计出 ODN 在合法程序和病毒程序中出现的频率信息. 模型要根据其频率信息，计算出每个 ODN 趋向于代表病毒的程度：ODN i 被挑选进入病毒 ODN 库的概率与其在病毒程序所有 ODN 中出现的频率成正比，与其在合法程序所有 ODN 中出现的频率成反比；与训练集中包含 ODN i 的病毒文件数与所有病毒文件数的比例成正比，与训练集中包含 ODN i 的合法程序数与所有合法程序数的比例成反比.

基于上述条件，本文提出了一个 ODN 选择公式，如下：

$$\text{浓度函数}\quad S^i=\begin{cases}\dfrac{W_n^i}{W_n^i+W_s^i}, & W_n^i\neq 0\\ 0, & W_n^i=0\end{cases}\qquad(1)$$

其中，$W_n^i=\dfrac{I_n^i\times F_n^i}{I_n\times F_n}$表示 ODN i 在病毒程序中的浓度；$W_s^i=\dfrac{I_s^i\times F_s^i}{I_s\times F_s}$表示 ODN i 在合法程序中的浓度.

下列表 1 是在本文采用的病毒库中部分病毒 ODN 浓度信息.

表 1　部分 ODN 的浓度信息

ODN	I_n^i	F_n^i	I_s^i	F_s^i	S^i
0685	18	10	1614	162	0.033897
0686	90	32	326	47	0.90548
0687	16	16	268	42	0.536853
0688	57	20	1582	159	0.187642
0689	175	27	2414	193	0.340751
068A	148	33	926	129	0.675725
068B	444	65	10980	195	0.407222
068C	79	39	481	70	0.823437

表中 S^i表示 ODN i 趋向于代表病毒程序的程度，当 S^i超过某一选择阈值之后，其就被加入到病毒 ODN 库中. 本文将此阈值记作 S_1，称作 ODN 选择阈值. 显然，ODN 选择阈值 S_1是一个与训练集有关的常数，即当训练集固定后，最优的 S_1也就固定成为一个常数选择. 选择一个合适的阈值使得病毒 ODN 库中个体数目最少且又最具病毒特征的代表性是非常重要的，这个参数的选择会在之后的实验分析部分给出方法讨论.

3.2　特征的存储结构

Deng 等人总结的病毒特点清晰地表明[2]，实际的病毒运行机理是：(1) 特征不应该为了计算的简洁方便而采取不符合实际的固定长度；(2) 多个特征并存才可以用来标识一个病毒，而非只采用一个病毒特征；(3) 病毒的多个特征间是有极大的相关联系的. 这些方面的研究为本文采用的特征存储结构指明了理论方向，虽然前人的工作没有包括任何建立起来的模型和试验.

本文首次提出了在个体层上检测病毒的概念，以充分利用多个相关基因的相关性. 尝试将每个病毒样本的多个基因存放在此病毒样本对应的一个数据库空间中，最后通过空间中的所有病毒进行两两匹配，得出病毒个体之间定义出的相似度值，为充分利用多个基因的相关性提供基础. 这种存储方式被称为个体层上的存储. 基于有向导的特征生成方法，可以很好控制住 ODN 的个数，特征存储的空间即可被控制住，从而控制了最终匹配检测时的计算代价，避免了出现训练的时间过长而致模型失去实用性的问题.

本文方法的训练流程和检测流程分别如图 2、图 3 所示，可以简单地概括为首先有向导地生成病毒特征的 ODN 库，这是组成病毒特征的最基本单

元,在此基础上与任一程序的字符串进行匹配,形成一系列不定数目的不定长 ODN 串,属于某个程序的储存在一起,不同程序的分开储存,从而得到了病毒基因库和类病毒基因库,在这过程中,需要运用人工免疫方法中的阴性选择算法,对初始得到的这种病毒候选基因进行免疫,去除其特征表示的模糊状态,进而得到用来标示文件可以应用于特征检测的检测基因库.

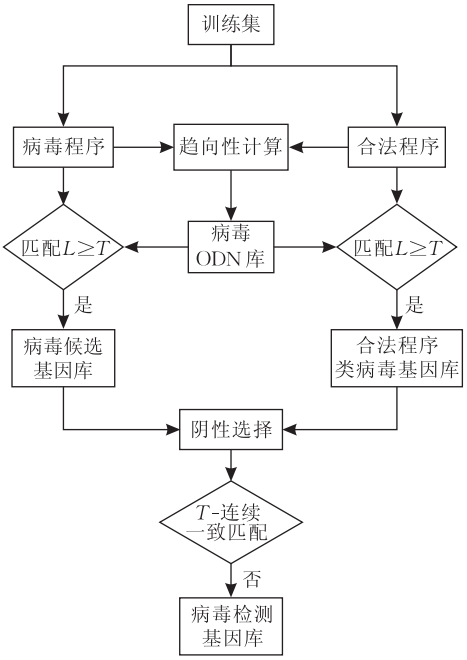

图 2 模型的训练流程

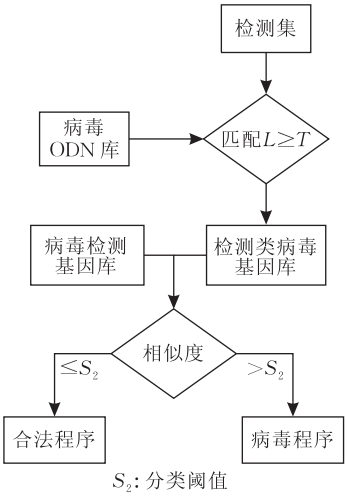

图 3 模型的检测流程

病毒基因库的基本存储单位是病毒样本个体.在每个病毒样本个体中,保存了该样本的所有基因,这样就使得同一病毒的不同基因存放在一起,不同病毒的基因分离保存.每个基因是不定长的,每个样本储存的基因数目也不同,如图 4 所示.

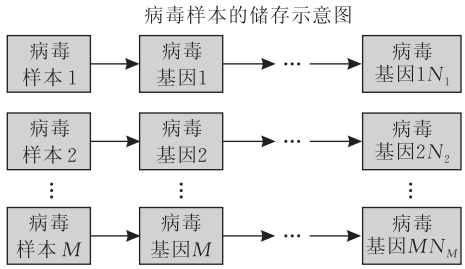

图 4 病毒基因的储存方式

3.2.1 病毒候选基因库

模型利用上一步生成的病毒 ODN 库中的 ODN,采用连续匹配的方式匹配病毒 DNA,从而生成病毒的候选基因.所谓连续匹配方式是指从第一个发生匹配的位置开始,采用滑动窗口的方式向后进行匹配比较,一直匹配前进,直到发生间断为止,此时检查从开始匹配到结束匹配共有多少个病毒 ODN 库中的 ODN 参与了匹配,如果 ODN 数目超过某个阈值 T,则将病毒 DNA 的这个片段作为病毒基因,否则认为该片段不包含足够多的信息,不是病毒的关键代码,即不是病毒的基因.显然,这种连续匹配方式具有一位 ODN 的容错能力.

这里,如果 T 太小,则生成的基因不能包含足够的信息,而造成基因的数量庞大,包含过多的无效基因,从而使系统性能和效果大大下降;而如果 T 太大,又会因为匹配长度要求较高,漏选一些重要信息.综合考虑了最常用的计算机指令为 1 或 2 个字节后,本文规定 T 取 3,这时,最小的基因长度为 4 个字节,包含了 1～4 个指令,具有较丰富的信息,可以成为一个基因了,在信息的多选和漏选之间做了一个平衡.

利用连续匹配方式,生成一个不定长的病毒候选基因的过程如图 5 所示.

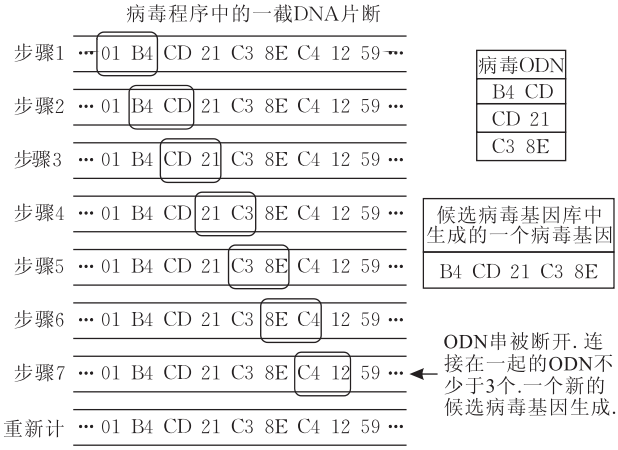

图 5 候选基因库的生成

采用这种方式，模型将所有生成的病毒基因都保存到与其对应的病毒样本的数据库空间中，形成了病毒候选基因库. 因为该基因库中的基因有可能与合法程序的基因发生匹配，故称作病毒候选基因库.

3.2.2　病毒检测基因库

利用病毒 ODN 库中的 ODN，采用生成病毒候选基因库的方法，通过 ODN 与训练集中所有合法程序 DNA 的匹配，模型能够很容易地生成合法程序的类病毒基因库.

模型将合法程序的类病毒基因看作“自体”，将病毒的候选基因看作“异体”，采用 T-连续一致匹配规则，进行阴性选择，即一旦病毒的某个基因与合法程序的任何一个基因匹配成功，则删除病毒的该候选基因. 重复这个过程，直到病毒候选基因库中所有和合法程序类病毒基因发生匹配的基因都被删除为止. 至此，病毒候选基因库升级成为病毒的检测基因库. 显然，此基因库中的基因都不和任何合法程序的类病毒基因匹配，能够完美地识别训练集中的合法程序. 在图 6 中，显示了整个免疫的进行过程，基因 1 是一个病毒候选基因，而基因 2 则是一个类病毒基因，它们通过以上定义的一致连续匹配进行了阴性选择，对传统的 R 位连续匹配规则进行了修改. 新规则如下：两个基因串进行匹配，采用滑动窗口的方式进行分析，如果二者连续不小于 T 个属于病毒 ODN 库中的 ODN 相同，则认为这两个基因匹配成功. 需要注意的是，这里的 T 指的是发生匹配的并且属于病毒 ODN 库中的 ODN 数目. 以后本文中提到的匹配长度，均为发生匹配的并且属于病毒 ODN 库中的 ODN 的数目.

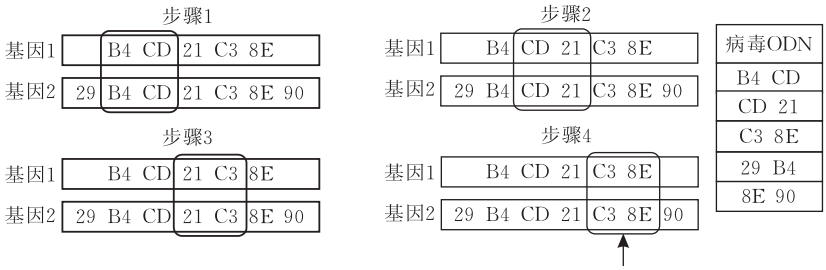

图 6　T-连续一致匹配规则

由于在生成病毒候选基因时，阈值 T 为 3，即只有大于等于 3 个 ODN 的连续连接才包含了足够的信息，形成一个基因，那么也只有大于等于 3 个 ODN 的连续连接都匹配，才可以认为这两个基因具有很强的相似性，此时才认为两个基因匹配成功，故 T 取值跟上文需要保持一致.

3.2.3　可疑程序类病毒基因库

利用病毒 ODN 库中的 ODN，采用生成病毒候选基因库的方法，通过 ODN 再与检测集中所有待检测程序 DNA 的匹配，模型能够很容易地生成待检测程序的类病毒基因库，整个过程相似于产生检测基因库的自体部分合法程序类病毒基因库的产生.

这里采用类病毒基因库能够减少无效基因的数目，从而提高模型的检测效率和检测时间. 原因在于非病毒 ODN 组成的基因与经过阴性选择的病毒检测基因是不能发生匹配的，这样的基因对整个系统是冗余的，对检测没有任何帮助. 之后的检测集待检测程序和病毒程序的匹配是由待检测程序的类病毒基因库与病毒检测基因库来完成的，它们的存储方式是以程序为基本单元的.

3.3　特征的多层次匹配

在特征的匹配问题之上，为了提高模型的准确度，本文希望利用尽可能多的有效信息，根据之前提出的特征提取方法和特征储存方式的特点，在 3 个逻辑层面上进行逐一匹配. 在对可疑程序进行检测时，在底层，即基因层，本文采用了 T-连续一致匹配规则，采用模糊匹配的方式来进行容错匹配，在最大限度上挖掘两个基因的相似性；在中层，即个体层，本文采用个体匹配的方式，得到可疑程序和病毒样本的相似度，最大限度上识别病毒的伪装，以发现已知病毒的新变种和未知病毒；在高层，即决策层，本文计算出可疑程序和病毒检测基因库的相似度，由整体对可疑程序进行分类，符合生物免疫学原理，即对抗原的识别是一种整体行为，能够最大程度上提高模型的准确性. 如图 7 所示.

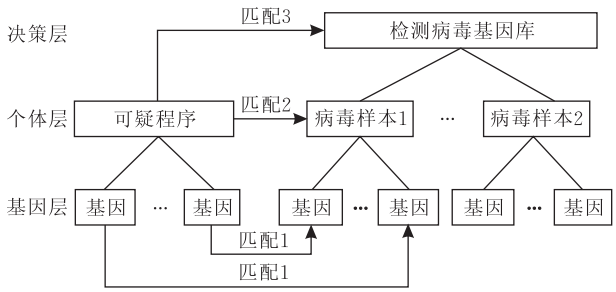

图 7　基于人工免疫系统的病毒特征分层匹配

3.3.1 基因库的优化

本文的所有基因库都是在个体层上对样本程序基因进行存储.由于同一病毒有可能具有多个相同基因,如果这些相同基因分别存储一次,则会浪费许多存储空间,增加系统中的基因数目,在基因库升级和可疑程序检测时增加许多不必要的运算,降低系统性能,所以模型为每个基因定义了一个权值,该权值等于该基因在其样本程序中的出现次数.这样,模型就对基因进行了一次优化,大大降低了其中的基因数目,使无效运算降至最低.另外,不同病毒也有可能具有相同的基因,因为模型是在个体层上对样本程序基因进行存储,以充分利用每个病毒不同基因间的相关性,因此这些相同基因还是需要分别存储.实际上,基因库的优化过程是随着基因的生成一起完成的.

3.3.2 基因匹配的相似度

两个基因进行匹配时,仍然需要采用 T-连续一致匹配规则.两个基因间的匹配程度定义为相似度值.

当两个基因不匹配的时候,本文定义这两个基因的相似度值为 0;当两个基因能够匹配的情况下,应满足以下的一个条件,即如果 $T_1=T_2+T_3$,其中 $T_i\geq T(i=1,2,\cdots)$,则两个基因 T_1 位 ODN 匹配时,这两个基因的相似度值要大于两个基因 T_2 位匹配时的相似度值与两个基因 T_3 位匹配时的相似度值的加和,定义 S_i 为 i 位 ODN 匹配的相似值,有 $S_n>S_{n-i}+S_i$.通过数学归纳法,有

$$S_n=2n-1 \tag{2}$$

综上所述,得到两个基因匹配程度的度量公式——相似度值计算公式如下:

$$S=\begin{cases}0, & n<R\\ \\ (2\times n-1)\times w_1\times w_2, & n\geq R\end{cases} \tag{3}$$

其中,S 为基因 1 和基因 2 的相似度值,n 为两个基因的匹配长度,w_1、w_2 分别代表基因 1 和基因 2 的权值.

3.3.3 个体间的匹配方式

将检测程序的一个基因和病毒样本的每个基因进行匹配比较,采用 T-连续一致匹配规则,利用式(3)计算出该基因和病毒样本每个基因的相似度值,并将最大的相似度值作为该基因和病毒样本的相似度值;重复这个过程,直到得到可疑程序每个基因和病毒样本的相似度值;最后将这些相似度值加和,作为可疑程序和病毒样本个体间的相似度值,算法如下:

```
//初始化 similarity[M],用于保存可疑程序每个基因
//和病毒样本的相似度值
initial similarity[M]=0;
//初始化 simiarity_indi,用来保存可疑程序和病毒样
//本的相似度值
initial similarity_indi=0;
    for (i=0; i<M; i++)
//遍历可疑程序的所有基因
    {
      for (j=0; j<N; j++)
//遍历病毒样本的所有基因
      {
//得到可疑程序第 i 个基因和病毒样本第 j 个基因的
//相似度值
        temp=get_match_value();
//保存可疑程序第 i 个基因和病毒样本前 j 个基因的
//最大的相似度值
        if (similarity[i]<temp)
          similarity[i]=temp;
      }
//计算得到可疑程序和病毒样本的相似度值
      similarity_indi+=similarity[i];
}
```

为了统一地对训练集以及检测集进行分析和决策,标准化处理待检测程序和病毒检测特征库的匹配值计算,可以得出可用阈值进行划分的检测程序和病毒检测基因库中每个病毒样本的相似度值,计算公式定义如下:

$$S=\frac{\sum_i S_i}{\sum_i \max((2n_i-1)\times w_i)\times\sum w} \tag{4}$$

其中,S_i 是病毒检测特征库中存储在一个病毒和待检测程序的相似度值,$\sum_i S_i$ 是整个检测基因库在个体层匹配之后的综合决策值,$\max((2n_i-1)\times w_i)$ 表示病毒样本 i 里包含的长度为 n_i 的检测基因能够提供最长的可能基因匹配相似度值,$\sum w$ 表示待检测程序所有类病毒基因的权值加和.

3.3.4 基因匹配过程中的参数选择

上文提到的选择阈值 S_1 和对于相似度度量的程序分类阈值 S_2 具有较高的耦合度,但如果同时优化选择两个参数在训练集上找到的使系统对训练集

识别率最高的参数，其推广能力会变得很差而无法在检测集中取得较好的结果. 分析有如下原因：病毒检测基因库经过了合法程序类病毒基因库的阴性选择过程，在训练集上合法程序和病毒检测基因库的相似度值为 0，绝大多数病毒程序和病毒检测基因库的相似度值不为 0，那么此时随着程序分类阈值 S_2 的提高，模型对合法程序的识别率没有提高，而模型对病毒程序的识别率将会降低. 由此，选择阈值 S_1，程序分类阈值 S_2 选择为 0，此时才能使模型以最高的识别率识别训练集. 往往实际情形是训练集中的合法程序数目有限，不能完全覆盖自体空间，所以经过阴性选择后产生的病毒检测基因库中仍然可能含有非病毒基因，这样就使得检测集上的合法程序和病毒检测基因库的相似度值不一定为 0. 如果优化选择程序分类阈值 S_2 为 0，就会使系统将检测集上所有和病毒检测基因库产生很小匹配的合法程序误判为病毒程序，从而使系统具有较高的误警率，降低系统的推广泛化能力. 从这里可以看出，在实际应用中，训练集上的最优参数是没有意义的.

鉴于此，本模型将选择阈值和程序分类阈值割裂开来考虑，采用一维搜索的方法，来对两个阈值参数分别进行优化选择. 具体的优化选择方式如下.

首先模型固定程序分类阈值，采用一维搜索的方法，根据选择阈值的取值范围及其意义，从 0.5 开始，以 0.005 为步长，前进 101 步，到 1 终止. 然后模型采用上一步得到的训练集上最优的 S_1，采用一维搜索的方法，根据程序分类阈值 S_2 的取值范围和意义，从 0 开始，以 0.0001 为步长，一直前进，直到模型对训练集中病毒程序的识别率低于 50%. 搜索过程中，随着程序分类阈值的增大，模型对病毒的识别率会单调降低，对合法程序将达到完美识别而不能再提高. 这样的特点导致此方法可以巧妙地利用检测集的一些规律性结论来为本文分类阈值参数选择提供帮助. 通过观察总结可以看到，在检测集上程序分类阈值和模型的识别率同样有上述关系，即程序分类阈值越大，则模型对检测集中合法程序的识别率越高，对检测集中病毒程序的识别率越低. 由此可以得到两条单调曲线的叠加：单调增加的合法程序的识别率和单调减小的病毒程序识别率，得到的模型对检测集的总体识别率（即正确识别的程序总数/检测集程序总数）是一个单峰曲线. 出于对容错能力和高识别率的一种权衡，将模型对训练集的平均识别率曲线第二个阶梯中点所对应的程序分类阈值 S_2 为最优的程序分类阈值. 具体的过程如图 8、图 9 所示.

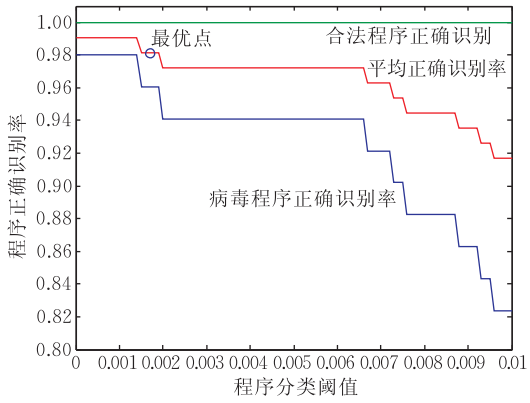

图 8　训练集分类阈值选择优化

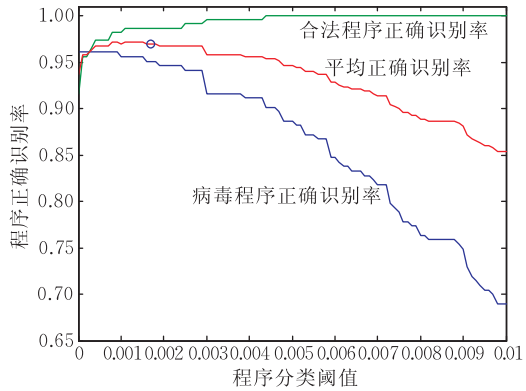

图 9　检测集分类阈值选择优化

4　实验及分析

本文的实验使用了两组数据集：(1) 文献[10,23]中使用的含有 1512 个经典病毒程序的数据集；(2) Windows 平台下实际运行的合法和非法程序中采集的数据集 cilpku08，这个数据集包含了最新的 3547 个病毒，根据它们各自的属性，可以划分归属于 685 个不同的病毒家族. 数据集中的具体内容可以在下列网址中下载：http://www.cil.pku.edu.cn/resources.

表 2 和表 3 分别是两组实验数据集的详细信息，表 4 是本文的实验环境.

表 2　Henchiri-Dataset 数据集

	文件类型	文件数目	平均体积	最小体积	最大体积
合法程序	EXE	1414	107	16	501
恶意程序	病毒	2880	6.2	22	93.4
	木马	88	9.4	49	72.5
	构造器	6	10	528	33.6
	其它	20	11.6	456	88.5

注：表 2 中，平均体积和最大体积的单位是 KB，最小体积的单位是 Byte，下同.

表 3 CILPKU08 数据集

	文件类型	文件数目	平均体积	最小体积	最大体积
合法程序	EXE	915	138.5	817	997
恶意程序	病毒	3465	4.8	23	59.5
	木马	39	4.4	49	5.93
	其它	43	6.8	48	31.2

表 4 实验环境

环境	配置
操作系统	Windows XP
计算机硬件	CPU:Pentium4 1.5GHz, RAM:512MB
编程语言	C 语言
编译平台	Microsoft Visual C++6.0

在第 1 个数据集上本文通过实验可以比较出本文方法的效果,在第 2 个数据集上通过三组随机抽取的病毒程序,以不同的集合划分,来验证实验结果的稳定性以及泛化能力.

本文将文献[10]作者 Henchiri 提供的第一组数据集中的病毒以家族为单位的方法分成 5 个部分.同时从 Windows XP 平台上收集到 1414 个合法程序,采用同样的办法也将其划分成 5 个部分.用本文提出的方法进行 5 倍的交叉验证实验,此时训练集中的病毒和检测集中的病毒在家族级别上相互独立,实验结果可信度较高.实验的结果如表 5 所示.

表 5 数据集 1 对比组实验结果

分类器	虚警率/%	病毒识别率/%	识别率/%
ID3	4.16	90.56	93.29
J48	5.24	92.56	93.65
NaïveBayes	0.13	37.17	69.51
SMO	5.71	92.26	93.39
Fold1	1.77	84.69	91.32
Fold2	1.77	90.99	95.05
Fold3	0.71	91.28	95.03
Fold4	2.47	95.53	96.36
Fold5	1.77	91.91	95.14
OURS	1.70	91.20	94.63

本文的方法在保持较低虚警率的情况下,取得了较高的病毒识别率,总体识别率较 Henchiri 等在文献[10]中的最优识别率高出 1%(见图 10).

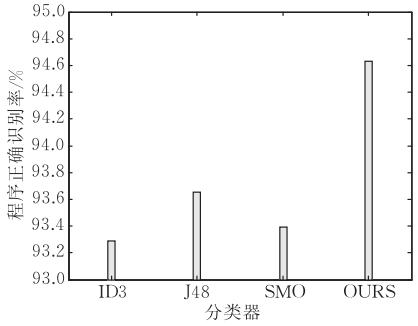

图 10 数据集 1 对比组实验结果

在第 2 组数据集上,第 1 组实验是在 538 个程序的数据集上进行的,其中共有284个合法程序和 254 个病毒程序.按照训练集和检测集 4/1、1/1、1/4 的比例,进行了 3 次不同的划分.实验结果如表 6 所示.

从表 6 可以看出,模型对训练集和检测集都有较高的识别率.其中,模型能够完美记忆训练集中的合法程序,对病毒程序的记忆率也在 96%左右;训练好的模型能够以 98%以上的识别率识别未知的合法程序,对完全未知的病毒程序也能达到 93%左右的识别率.值得重点说明的是,模型对训练集和检测集的识别率并没有随着训练集的减小而降低,这是传统病毒检测模型所难以实现的.尤其是在实验 Test3 上,训练集中程序数目远小于检测集程序数目,此时模型对训练集中的病毒程序的识别率为 96.1%,低于 Test2,却高于 Test1.Test3 上训练好的模型在检测集上具有更加出色的表现,其对合法程序的识别率高达 98.7%,对病毒程序的识别率也高达 95.1%,均高于 Test1 和 Test2 的模型在检测集上的表现.这就说明,本文提出的模型能够在小数据集上学习到足够的知识,并利用这些有限的知识,在检测无限大的空间时,取得足够好的成绩,具有较强的泛化能力.

表 6 数据集 2 第 1 组实验结果

实验编号	训练集							检测集						
	合法程序			病毒程序			APR/%	合法程序			病毒程序			APR/%
	A	P	PR/%	A	P	PR/%		A	P	PR/%	A	P	PR/%	
Test1	227	227	100	203	194	95.6	97.9	57	56	98.2	51	47	92.2	95.4
Test2	142	142	100	127	124	97.6	98.9	142	140	98.6	127	119	93.7	96.3
Test3	57	57	100	51	49	96.1	98.1	227	224	98.7	203	193	95.1	97.0

注:A 表示数据集中相应程序的程序总数;P 表示模型对程序的正确识别数目;PR 表示模型对程序的正确识别率,其中 PR=P/A;APR 表示模型对程序的平均识别率,其值等于(合法程序正确识别数+病毒程序正确识别数)/(合法程序总数+病毒程序总数).

第 2 组实验是在 1815 个程序的数据集上进行的,其中共有 915 个合法程序和 900 个病毒程序.按照训练集和检测集 2/1、1/1、1/2 的比例,对 1815 个程序组成的数据集进行了 3 次不同的划分.实验结果如表 7 所示.

表 7　数据集 2 第 2 组实验结果

实验编号	训练集							检测集						
	合法程序			病毒程序			*APR*/%	合法程序			病毒程序			*APR*/%
	A	*P*	*PR*/%	*A*	*P*	*PR*/%		*A*	*P*	*PR*/%	*A*	*P*	*PR*/%	
Test4	610	610	100	600	538	89.7	94.9	305	303	99.3	300	270	90.0	94.7
Test5	457	457	100	450	399	88.7	94.4	458	450	98.3	450	400	88.9	93.6
Test6	305	305	100	300	279	93.0	96.5	610	601	98.5	600	543	90.5	94.5

从上述两个表可以看出特征选择对于实验结果的稳定性.

接下来,实验被推广到更大的数据集上,用训练好的模型对整个数据集进行检测,验证对已知和未知程序的综合识别能力,在最大程度上反应模型的泛化能力,为模型在实际中应用提供一定的根据. 表 8 是上述训练好的模型直接应用在更大的检测空间的检测结果,用以验证对未知程序的综合识别能力.

表 8　数据集 2 第 3 组实验结果

实验编号	检测集						*APR*/%
	合法程序			病毒程序			
	A	*P*	*PR*/%	*A*	*P*	*PR*/%	
Test1	915	875	95.6	3547	3263	92.0	92.7
Test2	915	871	95.2	3547	3248	91.6	92.3
Test3	915	875	95.6	3547	3298	93.0	93.5
Test4	915	913	99.8	3547	3107	87.6	90.1
Test5	915	907	99.1	3547	3084	86.9	89.4
Test6	915	906	99.0	3547	3235	91.2	92.8

6 个在不同训练集上训练好的模型对检测集中合法程序的识别率基本都在 95% 以上. 特别地,Test4、Test5 和 Test6 的模型对检测集中合法程序的识别率高达 99%以上,这主要是因为这 3 个实验的训练集中合法程序数目较多,并且本模型能够完美识别训练集中的合法程序. 具有代表意义的一个实验是 Test6,其中的合法程序数目只有 305 个,但其对检测集合法程序的识别率竟高达 99.0%,这样,本文有理由认为,适当地提高训练集中合法程序数目能够较大地提高模型对检测集中合法程序的识别率. 6 个模型对病毒的识别率稳定在 92%左右. 特别地,Test3 的病毒程序训练数目较少,但对于包含 3547 个病毒程序的检测集,其识别病毒率高达 93.0%.

由于病毒检测系统是一个高度的实时系统,因此模型检测可疑程序的速度成为了模型可用性的重要指标. 上述 3 组实验中检测合法程序平均耗时约为 0.013s,检测病毒程序平均耗时约为 0.09s. 如图 11 所示.

通过以上几个方面的分析可以看出,本文提出的模型不仅能够有效地检测出病毒程序,同时能够在一个可以接受的训练时间内,产生一个稳定的而且基因规模不随训练集大小显著增长的病毒检测基因库,从而快速完成对可疑程序的检测,可以达到实时系统的要求,是一个有效的病毒检测模型.

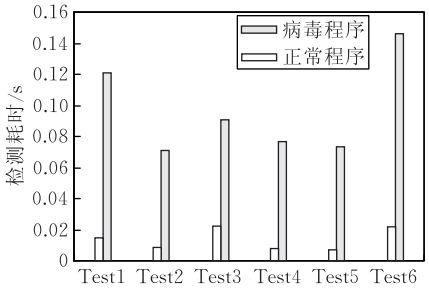

图 11　特征匹配检测耗时

5　结　论

本文的主要贡献在于结合了免疫思想和病毒本身两者特点,提出了一种新的更符合实际的病毒特征提取的方法,使得病毒被提取的特征生成时针对性更强;储存时突破了原先定长的限制;在匹配方面,比对在一个样本与另一个样本之间进行,样本内部的基因需要共同作用才能产生病毒性的特点被挖掘出来,病毒关键代码之间的相关性被充分利用,将对病毒的检测识别上升成为一种多层次分阶段的整体行为.

参　考　文　献

[1] White S R. Open problems in computer virus research//Proceedings of the Virus Bulletin Conference. Munich, Germany, 1998

[2] Deng P S, Wang J, Shieh W et al. Intelligent automatic malicious code signatures extraction//Proceedings of the IEEE 37th Annual International Carnahan Conference on Security Technology. Taipei, Taiwan, China, 2003: 600-603

[3] Gryaznov D. Scanners of the year 2000: Heuristics//Proceedings of the 5th International Virus Bulletin. Boston, Massachusetts, USA, 1999: 225-234

[4] Mo Hong-Wei. The Principles and Applications of Artificial Immune System. Harbin: Harbin Institute of Technology Press, 2002(in Chinese)

(莫宏伟. 人工免疫系统原理与应用. 哈尔滨：哈尔滨工业出版社，2002)

[5] Li Tao. Computer Immunology. Beijing: Publishing House of Electronics Industry, 2004(in Chinese)
(李涛. 计算机免疫学. 北京：电子工业出版社，2004)

[6] Jiao Li-Cheng. Immune Optimization: Calculation, Learning and Recognition. Beijing: Science Press, 2006(in Chinese)
(焦李成. 免疫优化——计算、学习与识别. 北京：科学出版社，2006)

[7] Jiao L, Du H. Development and prospect of the artificial immune system. Acta Electronica Sinica, 2003, 31(19): 1540-1548

[8] Mo Hong-Wei, Jin Hong-Zhang. Application of artificial immune system to computer security. Journal of Harbin Engineering University, 2003, 24(3): 278-286(in Chinese)
(莫宏伟，金鸿章. 人工免疫系统在计算机安全中的应用. 哈尔滨工程大学学报，2003，24(3)：278-286)

[9] Kephart J O, Arnold W C. Automatic extraction of computer virus signatures//Proceedings of the 4th Virus Bulletin International Conference. Abingdon, UK, 1994: 178-184

[10] Henchiri O, Japkowicz N. A feature selection and evaluation scheme for computer virus detection//Proceedings of the 6th International Conference on Data Mining (ICDM'06). Hong Kong, China, 2006: 891-895

[11] Forrest S, Perelson A S, Allen L et al. Self-nonself discrimination in a computer//Proceedings of the Security and Privacy. Oakland, CA, 1994: 202-212

[12] Forrest S, Hofmeyr S A, Somayaji A et al. A sense of self for Unix processes//Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, USA, 1996: 120-128

[13] Chao D L, Forrest S. Information immune systems//Proceedings of the 1st International Conference on Artificial Immune Systems. England: University of Kent at Canterbury Printing Unit, 2002: 132-140

[14] Dasgupta D, Forrest S. Tool breakage detection in milling operations using a negative-selection algorithm. Department of Computer Science, University of New Mexico: Technical Report Technical Report No. CS95-5, 1995

[15] Dasgupta D, Forrest S. Novelty detection in time series data using ideas from immunology//Proceedings of the ISCA 5th International Conference on Intelligent Systems. Reno, Nevada, 1996: 82-87

[16] Forrest S, Hofmeyr S A. Immunology as information processing//Segel L A, Cohen I R eds. Design Principles for the Immune System and Other Distributed Autonomous Systems. USA: Oxford University Press, 2000: 361-387

[17] Lee H, Kim W, Hong M. Artificial immune system against viral attack//Proceedings of the ICCS 2004. Lecture Notes in Computer Science 3037. Krakow, Poland, 2004: 499-506

[18] Dasgupta D, Attoh-Okine N. Immunity-based systems: A survey//Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation. Orlando, Florida, USA, 1997: 369-374

[19] Kephart J O. A biologically inspired immune system for computers//Proceedings of the 4th International Workshop on the Synthesis and Simulation of Living Systems. Cambridge, Massachusetts, USA, 1994: 130-139

[20] Karnik A, Goswami S, Guha R. Detecting obfuscated viruses using cosine similarity analysis//Proceedings of the 1st Asia International Conference on Modeling and Simulation. Phuket Thailand, 2007: 165-170

[21] Preda M D, Christodorescu M, Jha S et al. A semantics-based approach to malware detection//Proceedings of the 34th Annual Symposium on Principles of Programming Languages. Munich, Germany, 2007, 42(1): 377-388

[22] Kleiboeker S B. Applications of competitor RNA in diagnostic reverse transcription-PCR. Journal of Clinical Microbiology, 2003, 41(5): 2055-2061

[23] Schultz M G, Eskin E, Zadok E, Stolfo S J. Data mining methods for detection of new malicious executables//Proceedings of the IEEE Symposium on Security and Privacy. Oakland, CA, USA, 2001: 38-49

[24] Anchor K P, Williams P D, Gunsch G H et al. The computer defense immune system: Current and future research in intrusion detection//Proceedings of the 2002 Congress on Evolutionary Computation (CEC'02). Honolulu, HI, USA, 2002: 1027-1032

[25] Edge K S, Lamont G B, Raines R A. A retrovirus inspired algorithm for virus detection & optimization//Proceedings of the 8th Annual Genetic and Evolutionary Computation Conference. Seattle WA, 2006: 103-110

WANG Wei, born in 1982, Ph. D. candidate. His research interests include computational intelligence, data mining, artificial immune system and their applications to computer information security.

ZHANG Peng-Tao, born in 1986, Ph. D. candidate. His research interests include artificial immune system, intelligent information processing algorithm and computer information security, pattern recognition.

TAN Ying, born in 1964, Ph. D., professor, Ph. D. supervisor. His research interests include computational intelligence, machine learning, intelligent information processing and their applications to information security.

HE Xin-Gui, born in 1938, professor, Ph. D. supervisor, member of Chinese Academy of Engineering. His main research interests include fuzzy logic, artificial neural network, evolutionary computation, and database theory.

Background

This work outcomes of artificial immune theory research and its application to computer information security supported by the National Natural Science Foundation of China under grant Nos. 60673020 and 60875080, and partially supported by the National High Technology Research and Development Program (863 Program) of China, with grant No. 2007AA01Z453. The researches deal mainly with the facts that existing signature-based antivirus methods are inefficient to detect various forms of viruses, especially new variants and unknown viruses. These current methods often lead to a poor detection rate or a high false positive rate. Artificial immune system, inspired by human immune system, which is dynamic, adaptive and distributed, has an advantage by nature to protect benign files against virus invasion by distinguishing "nonself" from "self'. It is one direction of heuristic methods which is more sophisticated and has the potential to detect previously unknown viruses. Aimed at better performance of both detection rate and false positive rate, an AIS-based method with a novel feature selection and an improved negative selection mechanism is proposed in this paper to overcome three specific shortcomings in traditional AIS models. These shortcomings include using single fixed length virus signature to identity a virus, randomly generating the detectors leading to the bad efficiency, ignoring the relevance between different extracted signatures in one virus. Different from previous approaches, this paper try to use variable length instead of fixed length virus signatures and multiple signatures coexistence instead of single signature to identify a virus. The main contribution of this work is to fully utilize the sequencing relation of the virus signatures. This method generates a detecting virus gene library using the improved negative selection algorithm to ensure a fairly low false positive rate compared with the traditional signature-based methods. The advantages of our proposed method include: in the feature extraction phase, the detecting virus gene library stores virus samples with variable number of variable length genes at individual level, which helps raise the detection rate. The library also uses multiple genes coexistence in one virus to avoid the possible loss of information considerably, fully taking the advantages of relevance between viral instructions within a virus program; in the classification phase, suspicious programs is analyzed at individual level in contrast to the existing gene matching technique, the process neither memorizes specific byte sequences appearing in the actual file content nor monitors suspicious program behavior.