

改进的检测器大小可变的免疫异常检测算法^①

舒才良 严宣辉 曾庆盛 (福建师范大学 数学与计算机科学学院 福建 福州 350007)

摘 要: 在传统的免疫异常检测算法中, 通常存在检测器对非我空间的覆盖漏洞, 以及检测器数量过大且相互覆盖等问题, 这是导致免疫异常检测算法效率较低的主要原因。提出了一种能够处理混合型数据的免疫异常检测算法, 它能够生成不同大小的检测器, 提高对非我空间覆盖的效率。通过模拟实验和对比实验表明, 该算法能够较好地完成对混合型数据的处理, 并有效提高生成检测器的效率。

关键词: 入侵检测; 人工免疫; 阴性选择; 混合属性

Improved Immune-Based Anomaly Algorithm with Variable-Size Detector

SHU Cai-Liang, YAN Xuan-Hui, ZENG Qing-Sheng

(School of Mathematics and Computer Science, Fujian Normal University, Fuzhou 350007, China)

Abstract: Previous research has indicated there are some defects in the traditional immune-based algorithm of detecting abnormality. Two common defects of this are as follows: the detectors created by the algorithm cannot cover all Nonself-space, and the number of the detectors in the algorithm is too big to keep the detectors from covering each other. These defects are the main reasons why the algorithm is not efficient. In this paper, an algorithm, which can deal with mixed-type data and produce detectors in different sizes to improve the capacity of covering Nonself-space is proposed. The simulated and compared experiments show that this algorithm can preferably process mixed-type data and effectively improve the efficiency of generating detectors.

Keywords: intrusion detection; artificial immunity; negative selection; mixed-attribute

人类的免疫系统^[1]是一种高度进化、复杂的系统, 它具有学习、记忆和自适应的调节能力。该系统是由免疫分子、免疫细胞、免疫组织和免疫器组成, 其最基本的能力就是识别“自我与非我”(Self or Nonself), 也就是说它能够识别哪些组织属于正常机体, 哪些是不正常组织或外来的入侵体。对于“自我”, 免疫系统不作响应; 而对于“非我”, 免疫系统将对其进行清除。

受人类免疫系统的启发, 1994 年, Forrest 率先将人工免疫的思想引入计算机异常检测领域, 提出了阴性选择算法^[2]。2000 年, De Castro 和 Von Zuben 提出了克隆选择算法 (Clonal Selection Algorithm,

CSA)^[3]。之后许多学者围绕免疫原理在计算机安全领域中的应用问题做了大量的研究, 取得了一些令人振奋的成果^[4,5]。然而, 这些研究在取得一定成果的同时也存在一些不足:

(1)传统的免疫算法生成的检测器大小通常是固定不变的, 以致普遍存在检测器对非我空间的覆盖漏洞;

(2)许多算法生成的检测器有相互覆盖的现象, 使得检测器数量过大, 算法效率较低;

(3)大多数算法采用实数编码时, 只能处理数值性属性的数据, 而对混合属性数据缺乏处理能力。

文献中提出了一种“r 可变阴性选择算法”^[6], 该

① 基金项目:福建省自然科学基金(2007J0315);福建省教育厅科研专项重点项目(JK2009006)

收稿时间:2009-12-20;收到修改稿时间:2010-01-22

算法通过调整亲和力和阈值来解决检测器大小固定不变的问题,这对解决检测“黑洞(hole)”^[7]的问题提供了很好的思路。我们经过研究认为,可以对该算法加以改进,解决检测器之间相互覆盖和只能处理数值型数据等问题,以提高其效率和扩大其应用范围。

1 问题定义

从人工免疫学的角度看,异常检测问题就是区分自我和非我的问题。在基于人工免疫理论的异常检测算法中,“自体(Self)”是指合法的数据和信息,即正常的网络行为,“非自体(Nonself)”是指不合法的、恶意的攻击代码,即非法行为。为了描述本文的算法,首先定义以下概念和符号。

定义 1. 将所有的网络行为定义为抗原集合 Ag , $Ag = \{ \langle x, r \rangle | x \in R^n, x = (x_1, x_2, \dots, x_n), r \in R^+ \}$, 正常的网络行为定义为自体集合 $SelfSet$, 异常的网络行为定义为非自体集合 $NonselfSet$, $SelfSet \cup NonselfSet = Ag$, $SelfSet \cap NonselfSet = \emptyset$ 。

定义 2. 将所有用来检测抗原的检测器定义为抗体集合 Ab , $Ab = \{ \langle y, r \rangle | y \in R^n, y = (y_1, y_2, \dots, y_n), r \in R^+ \}$, $Ab = I_m \cup T_b$, I_m 为未成熟抗体集合, 为成熟抗体集合 $I_m = \{ x | x \in Ab, x.r = 0 \}$, $T_b = \{ x | x \in Ab, x.r \in R^+ \}$ 。

定义 3. “匹配”是指在一定的度量规则下,若抗原 ag 与抗体 ab 的相似程度(亲和力)小于匹配阈值 b , 则称抗原 ag 和抗体 ab 匹配, 记为 $Match(ag, ab)$ 。

定义 4. 将抗体与抗原, 抗体与抗体之间的匹配程度定义为亲和力, 亲和力计算公式:

$$affinity(ab, ag) = dist(ab, ag) + diff(ab, ag) \quad (1)$$

公式中 $dist(ab, ag)$, $diff(ab, ag)$ 分别用来处理分类属性性和数值型属性相似度。

$$dist(ab, ag) = \sum_{i=0}^k f(x_i, y_i) \quad (2)$$

其中 $f(x_i, y_i) = 0$, if $(ab.x_i = ag.y_i)$; else $f(x_i, y_i) = 1$ 。

$$diff(ab, ag) = \sqrt{\sum_{i=k+1}^n (x_i - y_i)^2} \quad (3)$$

例如: 有混合属性数据 $ab = (a, c, b, 0.3, 0.4)$, $ag = (c, a, b, 0.2, 0.3)$, $b = 1.15$, 由上面公式可得 $dist(ab, ag) = 1$, $diff(ab, ag) = 0.1414$, $affinity(ab, ag) = 0.1414$, $affinity(ab, ag) < b$, 所以 ab 与

ag 相匹配, 记为 $Match(ag, ab)$ 。

2 传统的阴性选择算法分析

传统的基于阴性选择的免疫学习算法步骤如下:

①初始化: 输入自体集合 $SelfSet$ 和匹配阈值 b ;

②以随机方式产生未成熟抗体 ab ;

③将未成熟抗体 ab 依次与自体集合 $SelfSet$ 中的抗原进行阴性选择;

a) 根据匹配规则, 如果 ab 遇到与之匹配的自体细胞, 则死亡;

b) 如果 ab 不与 $SelfSet$ 中任何自体细胞匹配, 则 ab 成熟, 将 ab 加入到成熟抗体集合 T_b 中;

④重复②、③步骤直到满足结束条件;

⑤返回生成的成熟抗体集合 T_b ;

图 1 表示了这个过程。

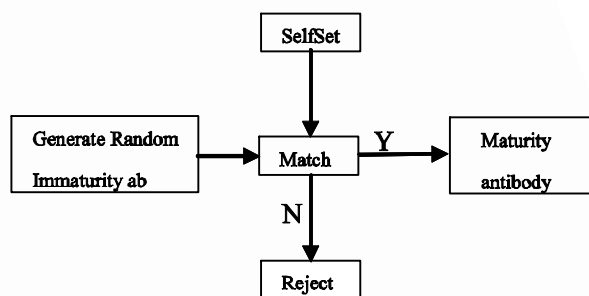


图 1 传统阴性选择算法流程

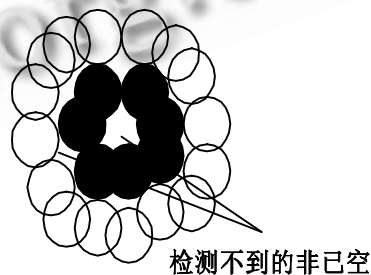


图 2 固定大小检测器覆盖空间 (图中深色的代表自体抗原, 周边无色的代表抗体)

上述算法中 b 为亲和力阈值, 检测系统利用算法生成的检测器集合 T_b 进行检测, 如果 T_b 中存在抗体 ab 与抗原 ag 之间的亲和力小于 b , 则此抗原 ab 为异常的, 否则为正常的。从抗体的生成及对抗原的检测方法中我们可以看到, 亲和力阈值 b 实质上相当于自体半径的大小。通过进一步分析, 我们发现传统的

阴性选择算法存在一些缺陷：非自体空间中，会存在生成的成熟抗体覆盖不到非我的空间，即“黑洞”（如图 2），从而使得落在此空间中的异常抗原不能被检测到，导致系统的检测率较低。“黑洞”是指没有产生成熟抗体来覆盖到的异常抗原的空间。

3 改进的检测器大小可变免疫异常检测算法

传统阴性选择算法中产生这一缺陷的原因是算法产生的抗体过于单一，没有考虑到抗体的多样性，对生成的相似抗体没有进行抑制。为减少传统阴性选择算法中“黑洞”的数量、提高算法检测效率和扩大其应用范围，我们提出了一种改进的检测器大小可变的异常检测算法。

3.1 检测器半径可变策略

根据免疫系统的多样性原理可知，人体的抗体并不是单一的，而是多样的，这样人体才能对抗千变万化的抗原[8]。抗体的存在会受到人体内同种抗体浓度的抑制。抗体的半径由下面公式确定，如下：

$$ab.r = \text{affinity}(ab, ag) - ag.r \quad (4)$$

其中， $\text{affinity}(ab, ag)$ 是当前抗体与自体细胞的最小亲和力，是自体细胞的半径。因为每个抗体与自体抗原的最小亲和力是不同的，所有自体抗原的半径是固定不变的，即不同的抗体具有不同的半径，所以抗体所覆盖的非我空间的大小不一样。抗体半径示意图如图 3。

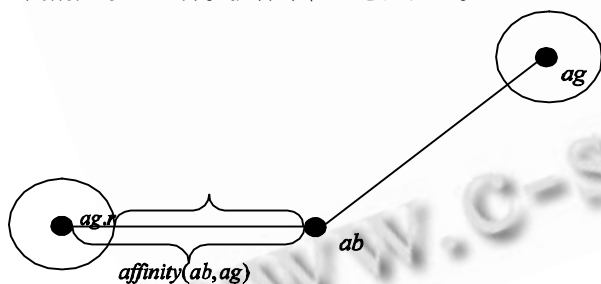


图 3 半径的确定

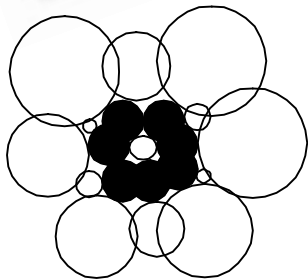


图 4 可变检测器覆盖的空间
(图中深色的代表自体抗原，周边无色的代表抗体)

例如， $ab=(c,a,b,0.3,0.4)$, $ag=(c,b,a,0.6,0.2)$, $ag.r=0.05$ ，由公式可以算得 $\text{affinity}(ab, ag) = 1.13$ $ab.r = 1.08$ 。半径可变检测器覆盖的空间如图 4。

3.2 抗体的抑制策略

为了减少成熟抗体的数量，我们在抗体的生成过程中采用了抑制策略。设当前未成熟抗体为，如果在成熟抗体集合中存在抗体，使得，则说明当前未成熟抗体已被抗体所覆盖到，是冗余抗体，抑制其成熟。

3.3 检测器可变的阴性选择算法

算法：改进的检测器大小可变的异常检测学习算法

输入：抗原集合，自体半径

输出：抗体集合

- ① Initialization: input SelfSet ;
- ② While(generationNum < maxGenNum)
- ③ { Generate immature antibody by random way;
- ④ For each $ab_j \in T_b$
- ⑤ { compute $\text{affinity}(ab_i, ab_j)$; // 计算抗体与抗体之间的亲和力
- ⑥ If ($\text{affinity}(ab_i, ab_j) < ab_j.r$);
- ⑦ { delete ab_i ; generationNum++; goto 3; } // 抗体抑制
- ⑧ For each $ag_j \in \text{self}$ // 阴性选择
- ⑨ { Compute $\text{affinity}(ab_i, ag_j)$; // 计算抗体与抗原之间的亲和力
- ⑩ If ($\text{affinity}(ab_i, ag_j) < ag_j.r$)
- ⑪ { delete ag_j ; generationNum++; goto 3; }
- ⑫ Retain the minimal $\text{affinity}(ab_i, ag_j)$; } // 保存当前最少的亲和力
- ⑬ $ab_i.r = \text{affinity}(ab_i, ag_j) - ag_j.r$; // 确定成熟抗体半径
- ⑭ $T_b = T_b \cup ab_i$;
- ⑮ generationNum++;
- ⑯ Return T_b ;

3.4 算法讨论

算法检测阶段用成熟抗体集合来检测异常，能被成熟抗体识别的抗原为异常抗原，否则为正常抗原。以下对算法的新特性进行讨论：

① 抑制策略：在抗体生成的过程中，引入抗体抑制策略(在算法 5、6 两步)，通过删除相互覆盖的抗体，

减少了冗余抗体的生成。

②处理混合型数据：从定义 4 可以看出，算法能够处理分类属性和数值属性的混合型属性数据，而传统的阴性选择算法一般采用二进制编码，用实数编码时，也只能处理数值型数据。

③改进了检测器半径的确定方法：算法产生的每个成熟抗体都带了一个大小不同的半径，以确定该成熟抗体的覆盖范围(如图 4)，检测器半径计算的方法是通过算法迭代循环中保存当前最少的亲和力抗原，并用公式(5)计算得到的。而原有算法是采用从小到大逐渐增加半径，进行多次尝试的方法来确定半径，其效率较低。

④检测方法的改进：算法在检测异常阶段是通过成熟抗体各自的半径来判断被测抗原是否在其覆盖范围内，而传统的阴性选择算法是通过阈值来确定的；文献[6]中的 r 可变阴性选择算法是通过预设多个阈值来实现的，而改进的检测器大小可变的异常检测算法则是为每个检测器确定了一个半径。

通过分析可以知道，本文所提出的算法使检测器集合尽可能多的覆盖非我空间，以降低黑洞的数量，提高检测效率。

4 仿真实验

实验在 Windows XP 系统下进行，采用 Visual C++ 6.0 作为编程工具，实验数据从 KDDCUP99 中随机选取。该数据集是 1998 年美国麻省理工学院林肯

实验室为入侵检测模型评估而建立的测试数据集，共提供了 4,900,000 条数据。包含 38 种不同的攻击类型，这些攻击类型可以归为 4 大类：Dos,R2L,Ur2 和 Probe。实验选用了数据 41 维属性中较重要的 10 维属性值，分别为 duration、protocol_type、service、flag、src_bytes、num_failed_logins、root_shell、num_access_files、srv_count、serror_rate、same_srv_rate。其中 protocol_type、service、flag 这 3 维为分类型属性，其它 7 维是数值型属性^[9]。

为了评价实验的效果，定义以下指标：

$$\text{检测率} = \frac{\text{正确检测到的正常和异常抗原的数量}}{\text{测试抗原总数}}$$

$$\text{漏报率} = \frac{\text{异常抗原被检测为正常抗原的数量}}{\text{异常抗原的数量}}$$

$$\text{误报率} = \frac{\text{正常抗原被检测为异常抗原的数量}}{\text{正常抗原的数量}}$$

实验时，数值型属性采用实数编码，并标准化到 [0, 1] 区间。对于分类型属性，将每个取值用一个整数来编码。我们把从 KDDCUP99 中选取的数据分成 5 个数据集，第一个数据集包含所有的攻击类型，其它四个数据集分别只包含 Dos,Probe,Ur2,R2L 这四种攻击类型。实验过程中设置算法的最大迭代次数 maxGenNum=10000，分别对每个数据集各进行了 50 次试验，统计对各种类型攻击的平均检测效果并与传统的阴性选择算法进行了比较，比较结果如统计数据表 1 所示：

表 1 与传统阴性选择算法对比实验结果表

异常类型	传统阴性选择算法			改进的阴性选择算法		
	平均检测率	平均漏报率	平均时间	平均检测率	平均漏报率	平均时间
Dos attack	0.978884	0.031652	4.910379	0.982842	0.023999	3.776476
Probe attack	0.987565	0.085868	5.611793	0.999203	0.003937	4.654300
Ur2 attack	0.997219	0.107438	4.908818	0.997749	0.086981	5.097074
R2L attack	0.944579	0.108754	5.173682	0.962113	0.091862	5.177444

从表 1 中可以看出，改进后的算法在检测效率上有了一定的提高，降低了漏报率和算法运行时间。算法在对 Ur2 攻击数据集测试时，时间比传统的算法运行的时间长。我们认为是因为算法在生成检测器的时候还是随机生成的，实验收敛时间会存在偏差。算法

在对 Ur2, R2L 两数据集进行测试时，虽然检测率很高，但漏报率同样也很高。这是因为在 KDDCUP99 数据集中绝大部分的入侵数据为 Dos 和 Probe 类型，只有很少一部分为 Ur2 和 R2L 类型的入侵数据。我们在实验中通过随机抽取的方式从整个数据集中选取部

分数据进行实验,因此 Ur2 和 R2L 类型的入侵数据数量较稀少,所以算法很难通过自体细胞学习到它们的特征导致漏报率相对较高,但对总的检测率影响不大。

自体半径是算法中一个需要预先设置的值,为分析这个预设值对算法性能的影响,我们对检测率、成熟抗体数量随自体半径 r 的变化做了实验分析,实验采用了第一个数据集,其中包括了所有攻击类型数据,我们将所有的攻击数据都当作异常进行了处理(设置为值 -1)。实验结果如图 5 和 6 所示。

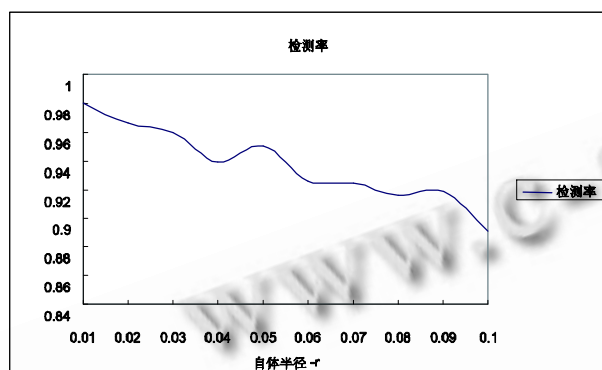


图 5 检测率随自体半径的变化

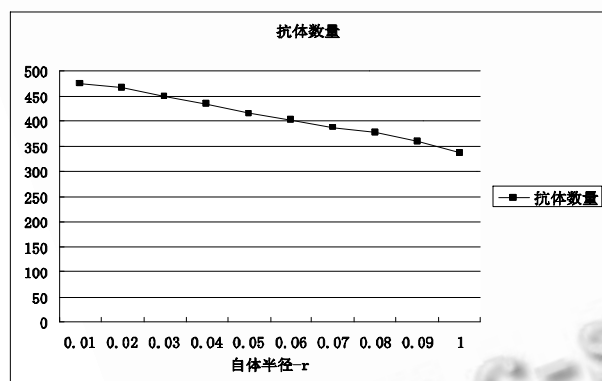


图 6 成熟抗体数量随自体半径的变化

从图 5 中可以看出,检测率随自体半径 r 的增大,呈现出下降的趋势。这是因为自体半径的增大,将部分异常抗原的空间划分到自体空间来,从而在抗体集中生成不了检测这部分异常抗原的成熟抗体,从而导致检测率下降。算法本身带有一定的随机性,在实验中会出现一些小的波动。同样从图 6,我们可以看到随着自体半径 r 的增大,算法生成的成熟抗体数量也呈下降趋势。自体半径增大了,自体细胞在空间中所占的区域就越大,相对的异常空间变小,所需用来覆盖异常空间的成熟抗体数量也相应减少了。实验在自

体半径为 0.01 时取得了最好的平均值,此时检测率为 98.02%,漏报率为 2.76%,成熟抗体数量为 475.3。

从上述对实验结果的分析可以看出,虽然减少自体半径 r 的值可以提高检测率,但也会使得成熟抗体的数量增加,这会使检测效率降低。这是因为一个训练好的检测器(或分类器)的分类规则越少,进行分类时的效率就越高,特别是对数据量大、实时性要求较高的环境是而言(例如高速网络环境中),这是一个重要的性能。在平衡检测率和检测效率的情况下,我们认为对于上述的实验数据,自体半径 r 设置为 0.01 是一个合理的值。以上实验数据的分析对于我们选择算法参数提供了参考依据。

5 总结

作为人工免疫系统中的核心算法之一的阴性选择算法,其性能对整个异常检测系统具有重要影响。检测器半径可变的阴性选择方法,是通过生成大小各不相同的抗体,自适应调整检测范围,增加抗体的多样性,减少“黑洞”的数量并以此提高检测率。

本文所提出的一种改进的检测器大小可变的异常检测算法,通过改进检测器半径生成方法,提高了算法效率,并采用了抑制策略以解决检测器相互覆盖的问题。此外算法能处理混合型数据等问题,从而提高了其检测效率并扩大了其应用范围。

对于所有需要预先设定阈值的检测方法而言,最大的困难是如何预先给定一个阈值。传统的方法是根据经验或多次实验尝试的结果来预先给定一个阈值,因此难免带有盲目性。而“检测器大小可变”的算法是通过自适应产生每个检测器的“半径”,实际上避免了预先设定阈值这个困难和不确定的问题。因此这种算法所具有的核心思想方法——“可变”,有相当大的可借鉴性,可以在许多领域得到推广和应用。

参考文献

- 1 李涛.计算机免疫学.北京:电子工业出版社,2004.
 - 2 Forrest S, Parelson A, Allen L, et al. Self-nonspecific Discrimination in A Compute. Proc. of the 1994 IEEE Symposium on Research in security and Privacy. Los Alamos,CA,IEEE Compture Society Press,1994.
 - 3 De Castro LN, Von Zuben FJ. The clonal selection algorithm with engineering applications. Proc. of GECCO. Workshop on Artificial Immune Systems and
- (下转第 43 页)

(上接第 59 页)

- Their Application, Nevada, USA, 2000:36 – 37.
- 4 Kim J, Bentley PJ. Towards an artificial immune system for network intrusion detection:an investigation of dynamic clonal selection. Proc. of the Congress on Evolutionary Computation 2002, Honolulu, 2002:1015 – 1020.
- 5 Ji Z, Dasgupta D. Real-Valued Negative Selection Alogrith with Variable-Sized Detectors. Genetic and Evolutionary Computation. Seattle: IEEE Press, 2004:287 – 298.
- 6 张衡,吴礼发,张毓森,等.一种 r 可变阴性选择算法及其仿真分析. 计算机学报, 2005,28(10):1614 – 1619.
- 7 Zhang LH, Zhang GH, Yu L, et al. Intrusion detection using rough set classification. Journal of Zhejiang University SCIENCE 2004,5(9):1076 – 1086.
- 8 Ji Z, Dasgupta D. Estimating the detector coverage in a negative selection algorithm. Genetic and Evolutionary Computation .Washington: IEEE Press,2005:88 – 97.
- 9 Fries TP. A Fuzzy-Genetic Approach to Network Intrusion Detection. Proc. of the 2008 GECCO Conference Companion on Genetic and Evolutionary Computation, 2008:2141 – 2146.