文章编号：2095-2163(2022)11-0034-07 中图分类号：TP393.08 文献标志码：A

基于灰狼优化的 V-detector 检测器分布方法

郑德强

（佳木斯大学 信息电子技术学院，黑龙江 佳木斯 154000）

摘 要： 基于免疫否定选择机制的 V-detector 算法，其检测器分布不优仍导致检测黑洞或冗余问题。本文借助群体智能优化算法中灰狼优化算法的无导数寻优机制，提出了一种改进的 V-detector 算法。引入混沌机制对灰狼种群进行合理的初始化，通过莱维飞行增强算法的全局寻优能力，利用改进后的 V-detector 算法优化检测器的落点分布，使得检测器的分布更加合理，并获得更高的覆盖率。在二维数据集上的验证，表明改进算法提高了检测器覆盖率；在 NSL-KDD 数据集上的验证，表明优化后的检测器集合能在较短时间内生成质量较高的检测器。

关键词： 入侵检测；V-detector 算法；灰狼优化算法；莱维飞行

Detection distribution method of V-detector based on grey wolf optimization

ZHENG Deqaing

（College of Information and Electronic Technology , Jiamusi University, Jiamusi Heilongjiang 154000, China）

【Abstract】 The poor detector distribution of V-detector algorithm based on immune negative selection mechanism still leads to detection of black holes or redundancy problems. This paper proposes an improved V-detector algorithm with the help of the non-derivative optimization mechanism of the gray wolf optimization algorithm in the swarm intelligence optimization algorithm. The chaos mechanism is introduced to initialize the gray wolf population reasonably, the global optimization capability of the algorithm is enhanced by the Levy flight, and the improved V-detector algorithm is used to optimize the drop point distribution of the detector, which makes the distribution of the detector more reasonable and obtains high coverage. The verification on the two-dimensional data set shows that the improved algorithm improves the coverage of the detector, and the verification on the NSL-KDD data set demonstrates that the optimized detector set can generate high-quality detectors in a short time.

【Key words】 intrusion detection; V-detector algorithm; gray wolf optimization algorithm; Levy flight

0 引 言

5G 商用时代的全面到来意味着互联网信息数量将迎来指数型爆炸，互联网安全随即变得更加重要。入侵检测系统（Intrusion Detection System, IDS）作为一种主动防御的网络安全技术，长期以来一直将其视作防火墙之后的第二道防线。通常情况下，入侵检测是通过对网络流量以及计算机系统日志等进行扫描，发现危险信息及漏洞。常见的入侵检测技术有基于人工免疫、统计分析以及神经网络[1]等技术的检测方法。其中，基于人工免疫的入侵检测作为一种受生物免疫系统启发而提出的入侵检测技术，受到了国内外学者的广泛关注。

作为人工免疫入侵检测技术的核心算法，否定选择算法（Negative Selection Algorithm, NSA）具有无须先验知识，只用少量正常样本便能检测无限数据的特点[2]。由于现代网络信息数据流的特点，否定选择算法已经从二进制否定选择算法发展为现在的实值否定选择算法（Real-value Negative Selection Algorithm, RNSA）。由于否定选择机制的特殊性，算法生成的检测器会存在检测黑洞或者冗余现象。针对这一特点，Zhou 等人[3]提出了 V-detector 算法。

V-detector 算法应对检测冗余现象优先生成半径较大的超球体检测器，针对覆盖黑洞问题生成半径较小的检测器，这样有利于控制检测器的规模，提高了 RNS 的精确性和时效性[4]。但是由于检测器分布的随机性和半径的不确定性也导致了检测器的高重叠问题，同时 V-detector 算法不可避免地沿袭了否定选择算法存在的检测黑洞问题。

针对 V-detector 存在的问题，文献[4]利用异常检测中容易被忽略的非自体元素，将非自体元素

基金项目： 国家自然科学基金（61172168）。

作者简介： 郑德强（1996-），男，硕士研究生，主要研究方向：网络与信息安全。

通讯作者： 郑德强 Email：208033016@ stu.jmsu.edu.cn

收稿日期： 2022-05-28

进行非自体区域的检测以及用来生成检测器。这样保证了新生成的检测器落入检测黑洞的概率大大减小，提高了检测黑洞的覆盖率以及检测器的质量。文献[5]通过修改检测器的生成规则以及对检测器进行优化，同时采用记忆检测器和成熟检测器两种组合，减少了计算量和存储空间的大小，提高了检测率，改进了无线传感器网络入侵检测模型。文献[6]通过与克隆选择算法(Clonal Selection Algorithm, CSA)的结合，引入定距变异的思想。提升了检测率和覆盖率，但是由于自体集的庞大，导致算法较为耗时。文献[7]受容差粗糙集启发，提出了一种新的检测器构造方法，将其运用到 V-detector 算法中，提高了检测效率。

鉴于此，本文引入灰狼优化(Grey Wolf Optimizer, GWO)，将其运用到检测器的生成中，并对检测器的分布进行优化，提高检测器的覆盖率。基本思想为：通过 GWO 算法对随机生成的检测器分布位置进行优化，优先生成较大半径检测器，同时标记出新的检测器，保证覆盖率。达到理想覆盖率后合理调整适应值，进行检测黑洞的覆盖。实验结果表明，该算法能显著提高检测器的分布覆盖率和减少检测黑洞，同时运行时间也较为理想。

1　相关工作

1.1　V-detector 算法

由于检测器的半径就是匹配的阈值，所以检测器的半径可以作为人工控制的因素来进行调配。V-detector算法针对实值否定选择算法存在的检测器数量庞大的问题，优先使用半径较大的检测器先覆盖非自体区域；针对检测漏洞问题，提出使用半径小的检测器去覆盖检测黑洞。这样大大提高了检测器的质量，但是检测器的分布位置仍是随机产生的，依然有优化的空间。因此，可以使用灰狼优化算法对检测器的落点位置进行优化，使检测器分布更加合理。

1.2　灰狼优化算法

研究可知，灰狼优化算法就是从灰狼群体捕食行为启发得到的[8]。灰狼群体有一个非常严格的社会统治阶层。灰狼种群的等级如图 1 所示。

由图 1 可知，领导层通常是一雄一雌，叫做 α，对应到算法，α 即为算法中的最优解。处于第二层的是 β 狼，β 狼属于从属狼，也是算法中的次优解。第三层的 δ 狼、也即算法中的第三优解。处于最底层的是 ω 狼，ω 为剩余的所有解。

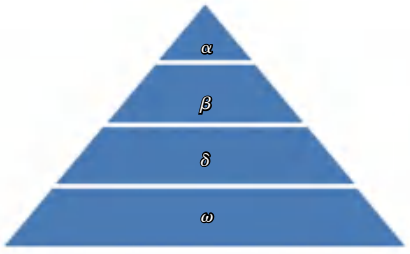

图 1　灰狼种群的等级制度

Fig. 1　Gray wolf population hierarchy

灰狼算法的数学模型可描述为：

$$\vec{D} = |\vec{C} \cdot \vec{X}_P(t) - \vec{X}(t)| \tag{1}$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \tag{2}$$

其中，t 为当前迭代次数；$\vec{A}$ 和 $\vec{C}$ 都是系数向量；$\vec{X}_p$ 表示猎物的位置向量；$\vec{X}$ 表示灰狼的位置向量。式(1)表示个体与目标的距离，式(2)表示灰狼个体的位置更新公式。在此基础上进一步推得的系数向量 $\vec{A}$ 和 $\vec{C}$ 的计算公式分别如下：

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \tag{3}$$

$$\vec{C} = 2 \cdot \vec{r}_2 \tag{4}$$

其中，$\vec{a}$ 为收敛因子，因子大小在迭代过程中从 2 线性减少到 0，这也是为了模仿灰狼靠近猎物的过程。$|\vec{r}_1|$ 和 $|\vec{r}_2|$ 为[0,1]之间的随机数。

灰狼群体的狩猎过程通常由 α 狼指挥，β 狼和 δ 狼偶尔也会参与狩猎指挥。由于在实际函数优化过程中，问题最优解(猎物位置)往往是不可知的，为了模拟灰狼的捕猎过程，规定 α、β 和 δ 对猎物的潜在位置有更好的了解。每次迭代都会得到当前最优的 3 个解的同时，强制其他灰狼个体(包括 ω)根据最优位置来更新自己的位置。此处需要用到的数学公式可写为：

$$\begin{aligned}
\vec{D}_\alpha &= |\vec{C_1} \cdot \vec{X}_\alpha - \vec{X}| \\
\vec{D}_\beta &= |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}| \\
\vec{D}_\delta &= |\vec{C_3} \cdot \vec{X}_\delta - \vec{X}|
\end{aligned} \tag{5}$$

$$\begin{aligned}
\vec{X}_1 &= \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha) \\
\vec{X}_2 &= \vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta) \\
\vec{X}_3 &= \vec{X}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta)
\end{aligned} \tag{6}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{7}$$

其中，$\vec{D}_\alpha$，$\vec{D}_\beta$ 和 $\vec{D}_\delta$ 分别表示 α，β 和 δ 到其它个

体之间的距离；$\vec{X}_\alpha$,$\vec{X}_\beta$ 和 $\vec{X}_\delta$ 分别表示 α, β 和 δ 的当前位置；$\vec{C}_1$,$\vec{C}_2$,$\vec{C}_3$ 是随机向量，$\vec{X}$ 为当前处理的灰狼个体位置。

式(6)中，$\vec{X}_1$,$\vec{X}_2$ 和 $\vec{X}_3$ 分别表示灰狼群体中 ω 个体向 α, β 和 δ 个体前进的步长和方向；式(7)则表示 ω 个体的最终更新位置。

2 本文算法

2.1 改进灰狼优化算法

针对灰狼算法的改进有很多，例如种群初始化方式、修改位置更新方程、重新设定距离控制参数及收敛因子等，为了应对形态空间的复杂性，本文引入混沌初始化和莱维飞行来增强灰狼优化算法的寻优能力。

2.1.1 混沌初始化

灰狼种群的初始化十分重要，初始灰狼优化算法的迭代种群都是随机产生，随机产生的种群容易陷入局部最优[9]。对应到检测器生成过程，局部最优会导致达到理想检测率所需检测器数量增加，增加运算资源的消耗。因此本次研究中引入混沌机制，用于群体的初始化。

混沌现象是一种非常普遍的非线性行为，表现出运动随机的同时也存在着一定的内在规律性，因此也被称为貌似随机的不规则运动。混沌机制具有非线性、非周期性、遍历性等特点，这些特点正有利于克服群体智能算法初始解的盲目性。实验证明[10]，使用混沌机制进行种群初始化会有利于优化算法进行更有效的全局搜索。

混沌机制有很多映射模型，常见的有 Logistic 映射、PWLCM 映射、Singer 映射等，本文选取 Logistic 映射模型[11]进行种群初始化。其数学模型如下：

$$z_{k+1} = \mu z_k(1 - z_k) \tag{8}$$

其中，μ 可称为分支参数，通常取值为 4；z_k 为混沌变量。

2.1.2 莱维飞行

由于自体形态空间的复杂性，实际搜索中难免出现寻优漏洞，导致检测黑洞的产生。为了更好地进行全局寻优，本文算法引入莱维飞行[12]来改进灰狼算法的寻优过程。

莱维飞行得名于其搜索的运动轨迹，即在随机行走的过程中会有一定的概率实现大跨步，将其与步长分布没有重尾的随机行走相比，该运动轨迹像飞行一样。

简而言之，莱维飞行是一种以短距离随机搜索为主，一定概率长距离搜索为辅的搜索行走方式。这种搜索方式可以使得算法在复杂的自体空间搜索变得更加全面，减少检测黑洞的产生和避免寻优算法陷入局部最优。

莱维飞行位置更新公式[13]为：

$$x_i^{(t+1)} = x_i^{(t)} + s \oplus Levy(\lambda) \qquad i = 1,2,\cdots,n \tag{9}$$

其中，$x_i^{(t)}$ 表示 x_i 迭代到第 t 代的位置；“$\oplus$”表示点对点乘法；s 为步长控制变量；$Levy(\lambda)$ 为随机搜索路径，满足：

$$Levy \sim u = t^{-\lambda} \quad 1 < \lambda \leqslant 3 \tag{10}$$

由于莱维飞行的实现至今未有一个统一的形式，所以本文采用 Mantegna 算法模拟，Mantegna 算法数学表述如下。

步长 s 计算公式为：

$$s = \frac{\mu}{|\nu|^{\frac{1}{\alpha}}} \tag{11}$$

其中，μ、ν 为正态分布，即：

$$\mu \sim N(0,\sigma_\mu^2)$$

$$\nu \sim N(0,1)$$

具体地：

$$\sigma_\mu = \left\{ \frac{\Gamma(1+\alpha)\sin\left(\frac{\pi\alpha}{2}\right)}{\Gamma\left(\frac{(1+\alpha)}{2}\right)\alpha^{2\frac{(\alpha-1)}{2}}} \right\}^{\frac{1}{\beta}} \tag{12}$$

其中，α 通常取值为常数 1.5。

2.2 IGV-detector 算法

人工免疫入侵检测算法的目标是生成的检测器拥有尽可能大的覆盖率和尽可能小的重叠率，很明显优先使用较大半径的检测器有利于得到更大的覆盖率。

假设有 N 个自体元素，设其中心点为 $P_i(i = 1,2,\cdots,N)$，此处的 $P_i = [x_1^i, x_2^i, \cdots, x_L^i]$ $(i = 1,2,\cdots,N)$，自体集的半径为 $R_i(i = 1,2,\cdots,N)$。生成 M 个检测器，规定检测器中心点为 $p_i(i = 1,2,\cdots,M)$，其中 $p_i = [w_1^i, w_2^i, \cdots, w_L^i]$，半径为 $r_i(i = 1,2,\cdots,M)$[14]。改进算法的目标是找到最优的 o_i 和 r_i，生成质量高的检测器保证较大的覆盖率和较小的重叠率。

对于检测器 j，假设 p_k 是离其最近的检测器的中心点，于是就有：

$$|P_k - p_j| = \min_i^N |P_i - p_j| \tag{13}$$

本文自体和检测器以及检测器和检测器之间的亲和力计算仍然采用欧氏距离，即 $|P_k - p_j|$ 和 $|P_i - p_j|$ 的计算公式见如下：

$$|P_k - p_j| = \sqrt{\sum_{l=1}^{L} (x_l^k - w_l^j)^2} \tag{14}$$

$$|P_i - p_j| = \sqrt{\sum_{l=1}^{L} (x_l^i - w_l^j)^2} \tag{15}$$

如果此时存在检测器 i 使得 $|P_i - p_j| \leqslant R_i$，即该检测器落在了自体区域，则该检测器视为无效检测器。此时得到检测器 j 的半径：

$$r_j = |P_k - p_j| - R_k \tag{16}$$

式(16)保证了每次生成的检测器都有尽可能大的半径，同时和自体不会有重叠。通过公式(15)和公式(16)得到检测器的半径 r_j 依赖于检测器的落点 p_j。由于落点位置的无导数性质，可以使用灰狼算法来优化检测器的 p_j，同时适应值被定义为检测器的 r_j 大小。对应于灰狼算法也就是半径较大的候选检测器个体等级更高，每次迭代完成后头狼 α 的位置即为相对最优检测器的位置，α 值即为该检测器的半径大小。

至此，研究给出了 IGV-detector 算法流程如图 2 所示。改进灰狼算法优化后的 IGV-detector 算法步骤详见如下。

Step 1　设定生成成熟检测器阈值以及自体集自体半径、灰狼算法迭代次数等参数。

Step 2　混沌初始化灰狼种群和设定算法参数 a，A 和 C。

Step 3　初始化前 3 只狼的位置并计算每只灰狼个体的适应值。

Step 4　记适应值最好的个体为 X_α，适应值次之个体为 X_β，适应值再次之个体为 X_δ。

Step 5　在不与自体元素和已有成熟检测器发生免疫亲和的前提下结合莱维飞行策略搜索自体空间，通过公式(7)更新每个个体的位置。

Step 6　更新灰狼算法参数 a，A 和 C。

Step 7　计算每只灰狼个体的适应值，并更新 α，β，δ。

Step 8　灰狼算法迭代次数加 1，并判断算法迭代次数是否到达阈值。如果未到达、转到 Step5。反之，继续执行算法。

Step 9　得到 X_α 即为合格检测器的中心，α 为合格检测器的半径。将合格检测器纳入成熟检测器集。

Step 10　判断成熟检测器个数是否到达阈值要求，没有即转 Step 2，达到要求则输出成熟检测器集并终止算法。

上文所述的算法保证了每次生成的成熟检测器尽可能拥有最大的半径以及最优的位置，但是由于迭代过程中迭代方向的不确定，使得落点在每次更换位置时要进行自体和检测器的免疫耐受，这使得在达到较高的覆盖率 C_1 时迭代生成一个成熟检测器所需时间大大增加，延迟到达理想的覆盖率 C_2 的时间。此时将优化算法的适应值由候选检测器半径改为检测器覆盖率，同时候选落点的亲和力计算只与自体元素进行。尽管这样会导致检测器重叠率增加，但是这样的重叠会更好地弥补检测黑洞的产生。

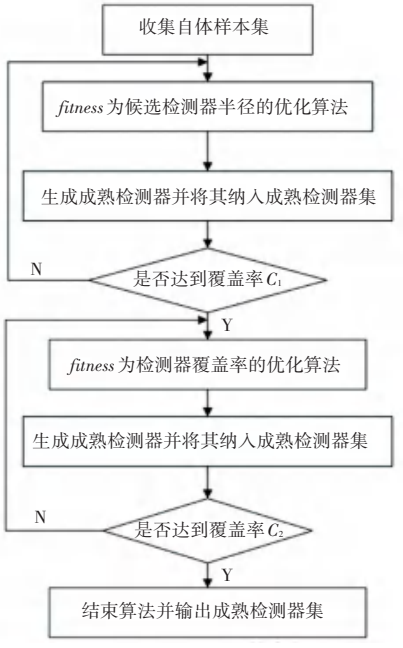

图 2　IGV-detector 算法流程

Fig. 2　IGV-detector algorithm flow chart

2.3　IGV-detector 算法收敛性分析

全局搜索算法的判敛准则及相关定义[15-16]如下。

定义 1　给定一个目标函数 f，函数的解空间是从 $\Re^n$ 到 $\Re$，S 是 $\Re^n$ 的一个子集。在 S 中寻找一个点 z，能够使得函数 f 的值最小化或者至少能够生成一个函数 f 在 S 上的可接受的下确界。

假设 1　$f(H(z,\xi)) \leqslant f(z)$，如果 $\xi \in S$，则 $f(H(z,\xi)) \leqslant f(\xi)$。其中，$H$ 是指可以在待求解问题空间产生解的函数。假设 1 要求 H 函数产生的新解优于当前解。z 是存在于解的子集 S 中的某个最小值，ξ 是根据相应算法在子集 S 中所得到的一系列的可行解。

定义 2　在 Lebesgue 测度空间中，定义搜索的下界为：

$$\phi = \inf(m; v[z \in S \mid f(z) < m] > 0) \quad (17)$$

其中，$v[A]$ 是在集合 A 上的 Lebesgue 测度。

假设 2 对于 S 的任意 Borel 子集 A，若其测度 $v(A) > 0$，则有 $\prod_{i=0}^{\infty}(1-\mu_i[A]) = 0$，这里 $\mu_i[A]$ 是由测度 μ_i 所得到 A 的概率。

定理 1 全局收敛的充要条件 假设目标函数 f 为可测函数，区域 S 是 $\Re^n$ 的可测子集，能够满足假设 1 和假设 2，设 $\{z_t\}_{t=0}^{\infty}$ 为算法生成的解序列，可得 $\lim_{t\to+\infty} B[z_t \in R_\varepsilon] = 1$，这里 $B[z_t \in R_\varepsilon]$ 是指在第 t 步由算法生成的解 $z_t \in R_\varepsilon$ 的概率。

引理 1 IGV-detector 算法满足假设 1。

证明 由于算法的迭代方向是单调的，即检测器的半径或者整体覆盖率都是逐渐变大的，所以本文算法明显满足假设 1。

引理 2 IGV-detector 算法满足 Condition 2。

证明 假设 2 是指对于位置测度为 v 的任意一个 A 的子集，如果采用随机抽样的方法，那么重复错过集合 A 的概率必定为零。由于算法的 ε 可接受区域 $R_\varepsilon \subset S(R_\varepsilon = \{z \in S \mid f(z) < \phi + \varepsilon\}, \varepsilon > 0)$，所有在可接受区域取得点的概率肯定是非零值。文献[17]已经证明了原始灰狼算法的灰狼群状态空间的一般状态转移至最优状态的转移概率为 1，即：

$$\lim_{t\to\infty} P^{(t)}(\zeta_i \to \zeta_j) = 1 \quad (18)$$

研究中证明了原始灰狼算法是全局收敛的，即满足上述条件。IGV-detector 算法是在 GWO 算法的基础上运用混沌初始化和莱维飞行策略更新灰狼种群的位置。因此，对于原始灰狼算法种群，设其支撑集的并集为 α；对运用混沌初始化以及莱维飞行作用的灰狼种群，设其支撑集的并集为 β。由于 2 种改进策略的随机性，必然存在整数 t_1，使得当 $t > t_1$ 时，$\beta \supseteq S$。因此，对于 IGV-detector 算法，存在整数 t_2，使得当 $t > t_2$ 时，$\alpha \cup \beta \supseteq S$。定义 S 的任意 Borel 子集 $A = M_{i,t}$，则有 $v(A) > 0$，$\mu_t[A] = \sum_{i=1}^{N}\mu_{i,t}[A] = 1$，即 $\prod_{i=0}^{\infty}(1-\mu_i[A]) = 0$。所以，IGV-detector 算法满足假设 2。

定理 2 GWO 算法收敛到全局最优，即 $\lim_{t\to\infty} P\{X(t) \in G \mid X(0) = \Phi_0\} = 1$。

证明 由于 IGV-detector 算法满足 Condition 1 和 Condition 2，算法满足定理 1 的条件，所以 IGV-detector 算法是一个全局收敛算法，也即算法以概率 1 全局收敛。

3 实验及结果分析

为了更直观地表示检测器分布情况，本文算法在二维数据集和公共数据集 NSL-KDD 下进行了验证。实验在 Matlab R2018b 软件下进行，其中主机配备 16 GB 内存、Intel(R) Core(TM) i5-1035G1 CPU、Windows 10 操作系统。实验统计工具采用 Matlab 自带的探查器工具。

3.1 算法在二维数据集下的验证

二维数据集选取的是 Zhou[3] 二维人造数据集。该数据集有环形及五角星等自体分布形态，每种形态有 1 000 条自体数据，皆被归一化到[0,1]之间。本实验对比了 V-detector 算法、粒子群优化[18] 的 PV-detector 算法、灰狼算法优化的 GV-detector 算法以及改进灰狼算法优化的 IGV-detector 算法，对比达到相同覆盖率所需检测器个数及时间、以及生成固定检测器个数所达到的覆盖率。

覆盖率的计算采取文献[3]中的方法，即在检测空间内随机采样 m 个点，如果有 n 个点未被覆盖到，则估计的覆盖范围为 $1-n/m$。当期望的覆盖率为 $\alpha = 1-n/m$ 时，至少迭代的次数应为：

$$m = 1/(1-\alpha) \quad (19)$$

实验选取五角星二维空间和环形二维空间进行测试。设置 PV-detector 算法、GV-detector 算法、IGV-detector 算法的寻优迭代次数为 100，当本文算法生成检测器覆盖率在五角星二维空间下达到 95%时，各算法生成检测器的分布覆盖情况如图 3 所示。

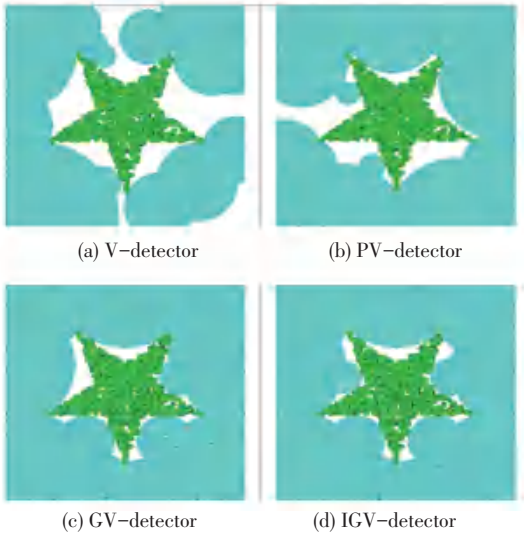

图 3 4 种算法生成相同检测器的覆盖情况

Fig. 3 The coverage for the same detector generated by four algorithms

由图 3 可知，由于 V-detector 算法检测落点都是未加干预地随机产生，所以其重叠率较高，覆盖率较低；由于检测器的分布有了指导方向，所以其余 3 种算法的检测器分布更加合理，优先在目标空间边缘生成大半径检测器，有效避免了边缘检测黑洞的产生，覆盖率更高。同时由于粒子群算法陷入“早熟”现象，灰狼优化算法拥有比粒子群算法更好的寻优和避免陷入局部最优的能力[8]，所以 GV-detector 算法、IGV-detector 算法的覆盖率比 PV-detector 算法的覆盖率更高。同样地，由于引入了莱维飞行，使得算法在寻优过程中实现了对自体区域的跨越，覆盖效果更好。

为了验证改进算法的鲁棒性和稳定性，本文在相同条件下重复试验了 20 次，测试在五角星和环形二维数据集下进行，当达到相同覆盖率 4 种算法所用时间和检测器个数详见表 1、表 2。

表 1　达到 95%覆盖率所用时间对比

Tab. 1　Comparison of the time to reach 95% coverage

算法	时间/s	
	五角星	环形
V-detector	37.723	40.308
PV-detector	20.234	35.184
GV-detector	18.945	16.454
IGV-detector	16.883	16.987

表 2　达到 95%覆盖率所用检测器个数对比

Tab. 2　Comparison of the number of detectors to achieve 95% coverage

算法	检测器数目(个)	
	五角星	环形
V-detector	21	28
PV-detector	18	16
GV-detector	14	15
IGV-detector	13	13

3.2　算法在公共数据集下的验证

公共数据集采取 NSL-KDD 数据集，该数据集是 KDD CUP99 数据集的改进版本，相较于后者，NSL-KDD 去除了大量的冗余信息，训练集和测试集的数据规模缩小后使数据集变得更加合理，同时使得实验有了一致性和可对比性。NSL-KDD 里面训练集包含 125 973 条记录，测试集包含 22 544 条记录，其中数据维数为 42 维。本文实验选取了其中 20%的数据。

在数据预处理部分，由于原始数据的离散性，需要进行连续化处理。比如协议类型部分，变换规则为：TCP →1；UDP → 2；ICMP → 3 等。同时由于高维数据对检测率的影响[19]，需要对数据进行降维处理。数据的降维方式有很多，本文选取的是比较成熟的 PCA 方法，选择保留 90%的特征降维后数据维度为 20 维。自体集数据的半径对最终的实验检测指标的影响是巨大的，本文的自体集数据半径设为 0.01。归一化方法选择最大最小归一化方法，方法公式具体如下：

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{20}$$

其中，$x_{\min}$ 为样本中数据最小值，$x_{\max}$ 为样本中数据最大值。

入侵检测对应于实际应用的评价指标主要有检测率、误报率等，本实验测试了 4 种算法不同的检测器数量对应的检测率。结果如图 4 所示。由图 4 中分析可知，V-detector 由于其生成检测器分布的不确定性，检测率呈现缓慢递增的趋势。其余 3 种优化算法在检测器数目较少时便能达到较理想的检测率，这是因为在检测器生成早期优先生成高质量的检测器，在算法后期生成的检测器多用于弥补检测黑洞，因此检测率增长缓慢，但总地来看优化后的算法仍能以相对较少的检测器数量达到较高的检测率。

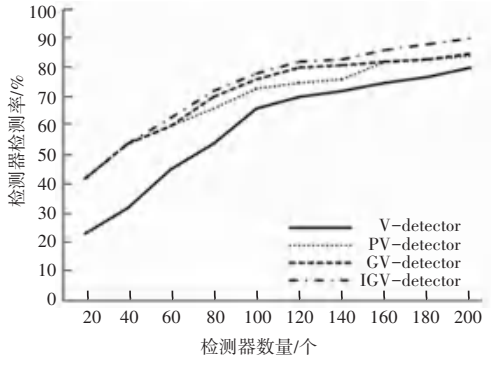

图 4　4 种算法生成相同检测器数量与检测率关系

Fig. 4　The relationship between the number of detectors and the detection rate generated by the four algorithms

生成 200 个检测器所需时间对比见表 3。由表 3 可知，达到相同检测率优化后的算法所需时间更短。

表 3　生成 200 个检测器所需时间对比

Tab. 3　Comparison of the time required to generate 200 detectors

算法	V-detector	PV-detector	GV-detector	IGV-detector
时间/s	580.56	520.12	510.23	486.72

总地来看，IGV-detector 算法能在较短的时间达到更高的检测率，符合入侵检测实时性的要求。

4 结束语

本文通过引入灰狼优化算法对经典算法 V-detector 检测器的分布进行了优化。灰狼优化算法等元启发式方法由于其无需计算导数、无需过多先验知识等优点在近些年受到了大量关注。V-detector 继承了否定选择算法的“基因”,即检测器都是随机产生的。在自体空间未知的情况下,随机似乎更能满足检测器分布广泛的需求。在引入灰狼优化算法之后,检测器的耐受分布便有了指导的方向,同时由于群体寻优的机制存在可以更好地应对复杂的自体空间。结果表明,改进后的 V-detector 算法提高了检测器的质量,减少了不必要的检测器重叠以及对检测黑洞进行了更好的覆盖。但是由于迭代过程中寻优方向的不确定性,导致每次产生的新的候选落点都要与自体和已有成熟检测器进行免疫耐受,这种不确定性导致了在检测器生成后期生成一个成熟检测器的时间有所增加。以后的改进方向应是对迭代落点的方向进行引导,避免或者减少这种大量的亲和力计算。

参考文献

[1] 刘月峰, 王成, 张亚斌, 等. 用于网络入侵检测的多尺度卷积 CNN 模型[J]. 计算机工程与应用, 2019, 55(03): 90-95,153.

[2] 金章赞, 廖明宏, 肖刚. 否定选择算法综述[J]. 通信学报, 2013, 34(01): 159-170.

[3] ZHOU Ji, DASGUPTA D. Real - valued negative selection algorithm with variable - sized detectors [C]//NCS 3102: Proceedings of Gecco. Berlin:Springer, 2005: 287-298.

[4] LI Zhiyong, LI Tao. Using known nonself samples to improve negative selection algorithm [J]. Applied Intelligence, 2021 (prepublish):

[5] SUN Ziwei, XU Yimin, LIANG Guangwei, et al. An intrusion detection model for wireless sensor networks with An improved V-detector algorithm[J]. IEEE Sensors Journal, 2018,18(5): 1971-1984.

[6] 何泾沙, 韩松, 朱娜斐, 等. 基于改进 V-detector 算法的入侵检测研究与优化[J]. 信息网络安全, 2020, 20(12): 19-27.

[7] CHMIELEWSKI A. Tolerant V -detector algorithm[J]. Journal of Physics: Conference Series, 2018, 1061(1).

[8] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey Wolf optimizer [J]. Advances in Engineering Software, 2014, 69: 46-61.

[9] 张琳,汪廷华,周慧颖. 基于群智能算法的 SVR 参数优化研究进展[J/OL]. 计算机工程与应用: 1-16[2021-06-23]. http://kns.cnki.net/kcms/detail/11.2127.TP.20210521.1125.014.html.

[10] ANDERSON-COOK C M. Practical genetic algorithms [J]. Publications of the American Statistical Association, 2004, 100 (471): 1099.

[11] RAGULSKIS M, VAINORAS A, SMIDTAITE R, et al. The logistic map of matrices[J]. Discrete and Continuous Dynamical Systems-Series B, 2013, 16(3): 927-944.

[12] VISWANATHAN G M, AFANASYEV V, BULDYREV S V, et al. Lévy flights search patterns of biological organisms[J]. Physica A: Statistical Mechanics and its Applications, 2012, 295(1): 85-88.

[13] 王庆喜, 郭晓波. 基于莱维飞行的粒子群优化算法[J]. 计算机应用研究, 2016, 33(09): 2588-2591.

[14] GAO X Z, OVASKA S J, WANG X. Genetic algorithms-based detector generation in negative selection algorithm [C]// IEEE Mountain Workshop on Adaptive & Learning Systems. Logan, UT, USA:IEEE, 2006.

[15] 方伟,孙俊,谢振平,等. 量子粒子群优化算法的收敛性分析及控制参数研究[J]. 物理学报, 2010,59(06):3686-3694.

[16] 谢铮桂,钟少丹,韦玉科. 改进的粒子群算法及收敛性分析[J]. 计算机工程与应用, 2011,47(01):46-49.

[17] 张孟健,龙道银,王霄,等. 基于马尔科夫链的灰狼优化算法收敛性研究[J]. 电子学报, 2020,48(08):1587-1595.

[18] KENNEDY J, EBERHART R C. Particle swarm optimization [C]//Proceedings of IEEE International Conference on Neural Network. Piscataway, NJ:IEEE,1995: 1942-1948.

[19] 张凤斌, 杨泽, 葛海洋. 基于聚类的邻域检测器生成算法[J]. 计算机工程, 2016, 42(02): 131-136.

(上接第 33 页)

[10] WU Mingjie, CHEN Qingkui. Low-cost andhigh-speed communication scheme in edge cluster based on DPDK[J]. Journal of Chinese Computer Systems, 2020,41(12): 2641-2648.

[11] RIZZO L. Netmap:a novel framework for fast packet I/O[C]// USENIX ATC'12: Proceedings of the 2012 USENIX conference on Annual Technical Conference.Berkeley, CA:ACM,2012:101-112.

[12] 吴明杰, 陈庆奎. 基于 DPDK 的边缘集群内低成本高速通信方案[J]. 小型微型计算机系统,2020,41(12): 2641-2648.

[13] ZHU Wenjun, LI Peng, LUO Bao-zhou, et al. Research and implementation of high performance traffic processing based on intel DPDK [C]//2018 9th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP). Taiwan: IEEE,2018: 62-68.

[14] KOURTIS M A, XILOURIS G, RICCOBENE V, et al. Enhancing VNF performance by exploiting SR-IOV and DPDK packet processing acceleration [C]/ /2015 IEEE CONFERENCE ON Network Function Virtualization and Software Defined Network (NFV-SDN) .San Francisco, CA: IEEE,2015: 74-78.

[15] WANG Haipeng, HE Dazhong, WANG Huan. Comparison of high-performance packet processing frameworks on NUMA[C]// IEEE International Conference on Software Engineering & Service Science (ICSES). Beijing:IEEE ,2016: 54-58.

[16] XUE Zhenghua, DONG Xiaoshe, HU Leijun, et al. Research on transmission model of high performance server cluster deployment system [J]. Chinese Journal of Computers,2008,31 (11) :1956-1964.

[17] CERRATO I, ANNARUMMA M, RISSO F. Supporting fine-grained network functions through intel DPDK [C]//2014 Third European Workshop on Software Defined Networks(EWSDN) . Torino, Italy :IEEE,2014: 1-6.