基于智能蜂群算法的 DDoS 攻击检测系统

余学山[1] 韩德志[1] 杜振鑫[1,2]
(上海海事大学信息工程学院 上海 201306)[1]
(韩山师范学院计算机与信息工程学院 广东 潮州 521041)[2]

摘 要 随着大数据应用的普及,DDoS 攻击日益严重并已成为主要的网络安全问题。针对大数据环境下的 DDoS 攻击检测问题,设计了一种融合聚类和智能蜂群算法(DFSABC_elite)的 DDoS 攻击检测系统。该系统将聚类算法与智能蜂群算法相结合来进行数据流分类,用流量特征分布熵与广义似然比较判别因子来检测 DDoS 攻击数据流的特征,从而实现了 DDoS 攻击数据流的高效检测。实验结果显示,该系统在类内紧密度、类间分离度、聚类准确率、算法耗时和 DDoS 检测准确率方面明显优于基于并行化 K-means 的普通蜂群算法和基于并行化 K-means 算法的 DDoS 检测方法。

关键词 DDoS 攻击,智能蜂群算法,流量特征分布熵,聚类算法,广义似然比较

中图法分类号 TP309.2 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.12.019

DDoS Attack Detection System Based on Intelligent Bee Colony Algorithm

YU Xue-shan[1] HAN De-zhi[1] DU Zheng-xin[1,2]
(College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China)[1]
(School of Computer Information Engineering, Hanshan Normal University, Chaozhou, Guangdong 521041, China)[2]

Abstract With the popularity of the applications of big data, DDoS attacks become increasingly serious and have been the main network security issues. This paper designed a DDoS attack intrusion detection system based on clustering and intelligent bee colony algorithm (DFSABC_elite) for DDoS attack detection in environment of big data. The system combines the clustering algorithm and the intelligent bee colony algorithm to classify DDoS attack data flow, and uses the traffic feature distribution entropy and the generalized likelihood comparison distinguishing factor together to detect the characteristics of DDoS attack data stream, thus achieving the efficient detection of DDoS attack data flow. Experimental results show that this system is obviously superior to the ordinary bee colony algorithm based on parallelization K-means and the DDOS detection algorithm based on parallelization K-means in terms of intra-class compactness, inter-class separation, clustering accuracy, consumed time and DDoS detection accuracy.

Keywords DDoS attack, Intelligent bee colony algorithm, Traffic feature distribution entropy, Clustering algorithm, Generalized likelihood comparison

1 概述

分布式拒绝服务攻击(Distributed Denial of Service, DDoS)是目前网络上最常见且最难防御的一种网络攻击。2014 年 12 月,爆发了运营商 DNS 网络 DDoS 攻击事件。从 12 月 10 日凌晨开始,网络监控到攻击流量突增的情况,上午 11 点开始,攻击开始活跃,多地不断出现网页访问缓慢,甚至无法打开等故障。攻击者不仅在短时间内发起了峰值大于 6G bps 的查询请求(全国范围内大于 100G 的攻击),而且连续变换二级域名,造成各地 DNS 递归服务器的延迟增长,核心解析业务受到严重影响。现在的 DDoS 攻击范围遍布全球,攻击源难以追踪和定位,使得 DDoS 攻击检测的难度大大增加。此外,分布式攻击并无规律可寻,某些 DDoS 攻击发送的数据请求是合理的,使用了常见的协议和服务,这给 DDoS 攻击的检测带来了严峻的考验。

目前,一些 DDoS 攻击检测模型(如堆空间监测[1]、流量特征分辨[2]等)依然存在很多缺陷,例如在检测时存在较大的遗漏率、较多的漏检流量数据以及高误报率。本文研究大数据环境下的 DDoS 攻击检测方法,与传统 DDoS 攻击检测方法不同,本文方法将 DDoS 攻击检测算法嵌入到大数据处理

到稿日期:2017-11-15 返修日期:2018-01-16 本文受国家自然科学基金(61373028,61672338)资助。

余学山(1993-),男,硕士,主要研究方向为机器学习与云计算,E-mail:xueshan0529@163.com;**韩德志**(1966-),男,博士,教授,CCF 高级会员,主要研究方向为云计算、云存储与安全技术、大数据应用,E-mail:dezhihan88@sina.com(通信作者);**杜振鑫**(1976-),男,博士,讲师,主要研究方向为机器学习、数据挖掘与云计算。

框架中，针对超大规模数据量进行并行实时异常检测，而传统的 DDoS 攻击检测算法则不能与大数据处理框架相结合，对大数据流异常检测的实时性和效率都较差。本文设计了一种基于智能蜂群算法以及能够在 Spark 数据处理平台上运行的并行化 K-means 聚类算法[3]，结合蜂群觅食行为模型与聚类思想建立了新的数据流分类模型来分离异常数据流，同时结合流量特征分布熵与广义似然比较判别因子，以快速并精准地检测异常数据流中的 DDoS 攻击数据流。

2 DDoS 攻击的原理及特点

DDoS 攻击[3]是当前网络中最为常见且最难防御的一种网络攻击，具有流量高、攻击范围广、攻击源难以确定等特点。DDoS 是一种分布式网络攻击，攻击源遍布全球网络。DDoS 攻击者可以使用网络上的任意一台主机或者个人电脑作为攻击端，通过非法扫描或入侵控制网络上一些闲置的主机或服务器来控制大量代理机，即傀儡机。攻击者以这些被控制的主机或服务器作为主控端，通过植入一些攻击命令来使其进一步向所控制的代理机发送攻击命令。攻击者躲在庞大的网络背后进行各种操作，实现了大范围的分布式拒绝服务攻击，这些攻击往往难以追踪和察觉。DDoS 攻击的原理如图 1 所示。

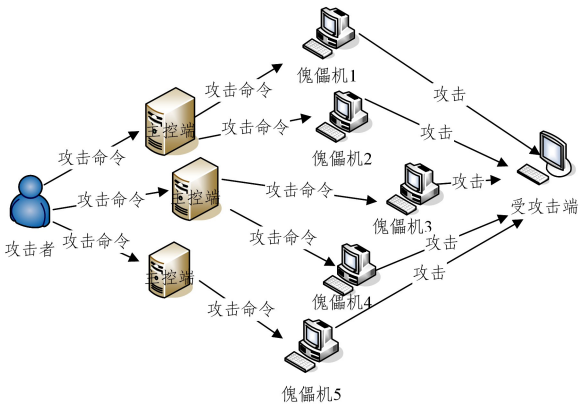

图 1 DDoS 攻击的原理图

Fig. 1 Schematic of DDoS attack

3 基于智能蜂群算法的 DDoS 攻击检测

3.1 聚类问题及原理

聚类属于无监督学习方法[4]，其主要原理是：在未知样本的情况下，计算样本数据间的距离（欧氏距离、汉明距离、马氏距离、余弦距离等），再根据样本距离划分数据所属类别。聚类问题[5]是指在数域空间，存在具有 n 个数据对象的数据 $X=\{X_1,X_2,X_3,\cdots,X_i,\cdots,X_n\}$，第 i 个数据对象 $X_i=\{x_{i1},x_{i2},\cdots,x_{im}\}$，$x_{im}$ 表示数据对象 X_i 的第 m 个特征。聚类的过程就是为数据集 X 找到一个划分 $G=\{G_1,G_2,\cdots,G_k\}$，即 G 满足 $\forall g\neq h, C_g\neq\varnothing, C_g\cap C_h=\varnothing, \forall g, \bigcup_{g=1}^{k} C_g=X$，同时 C_g 内包含的数据对象尽量相似，且 $\forall g\neq h, C_g$ 所包含的数据与 C_h 内的差异度尽量大。K-means[6]是最为经典的一种聚类划分算法，其原理是首先随机选取 k 个对象，每个初始对象表示一个聚类的中心或平均值，依次遍历剩余的每个对象，计算其到各个聚类中心的距离，比较这些距离的值，并将它们分配到距离最小的聚类中心，然后重新计算每个聚类的中心。重复该过程，直到聚类准则函数收敛。

3.2 智能蜂群算法的基本原理及优点

在 ABC（Artificial Bee Colony）算法[7]中，人工蜂群包括引领蜂、跟随蜂和侦查蜂 3 类。假设在 D 维空间中，种群规模为 $2\times SN$（引领蜂个数＝跟随蜂个数＝SN），蜜源与引领蜂逐一对应，即蜜源数目为 N，第 i 个蜜源的位置为 $X=\{X_1,X_2,X_3,\cdots,X_N\}$。每个蜜源的位置代表优化问题的一个候选解，花蜜的数量反映解的质量。人工蜂群搜索最优蜜源的过程如下：

1）引领蜂对当前蜜源进行邻域搜索，产生新蜜源，贪婪选择较优蜜源。

2）跟随蜂根据引领蜂分享的信息选择一个蜜源并进行邻域搜索，贪婪选择较优蜜源。

3）引领蜂放弃蜜源，转变成侦查蜂，并随机搜索新的蜜源。在搜索过程中，跟随蜂根据引领蜂分享的信息，以轮盘赌的方式按式（1）选择一个蜜源：

$$p_i=\frac{fit_i}{\sum_{i=1}^{N} fit_i} \tag{1}$$

其中，p_i 表示第 i 个解的适应度，fit 是食物源适应度，其可通过式（2）进行计算：

$$fit_i=\begin{cases}\dfrac{1}{1+f_i}, & f_i\geqslant 0\\ 1+abs(f_i), & \text{其他}\end{cases} \tag{2}$$

引领蜂根据记忆中食物源的位置进行邻居搜索，当找到更好的食物源时会评估其适应度，引领蜂根据式（3）进行搜索：

$$v_{ij}=X_{ij}+R_{ij}(X_{ij}-X_{hj}) \tag{3}$$

其中，$i\in\{1,2,\cdots,N\}$，$j\in\{1,2,\cdots,m\}$，并且 h 是随机选取的，R_{ij} 是$[-1,1]$之间的一个随机数。V_{ij} 表示邻居食物源，X_{ij} 表示当前食物源，X_{hj} 表示随机选取的食物源，且 X_{hj} 中的 h 是随机选取的。每个解都会经历数次迭代[8]，如果其没有被改善则将被舍弃。如果某个解 i 经过 $limit$[9] 次迭代后未更新成功，则按照式（4）初始化：

$$X_i=X_{\min}+rand(0,1)(X_{\max}-X_{\min}) \tag{4}$$

其中，$X_{\max}$ 和 $X_{\min}$ 分别表示定义域的上边界和下边界。

ABC 算法是一种基于群体的随机优化方法，已被证明用于解决优化问题时非常有效。该方案的搜索方程虽然在勘探方面较好，但在开发方面仍然存在不足，因此 ABC 算法的开采性能并不突出。为了更好地平衡勘探和开发，文献[10]提出了结合深度优先搜索（DFS）框架的新的蜂群算法 DFS-ABC。DFS 框架可以将更多的计算资源优先分配给更好的解，以增强算法的开采能力。这里将 DFS 框架应用于 ABC，DFS 框架在大多数情况下可以加快收敛速度。为了更好地平衡 ABC 的勘探与开采能力，借鉴了文献[10]中的两个新颖的解决方案，其搜索方程分别如式(5)和式(6)所示。这两个方案能取得更好的性能，且明显优于多种粒子群算法和差分算法。

$$V_{i,j}^G = X_{e,j}^G + \phi_{i,j} \cdot (X_{e,j}^G - X_{k,j}^G) \tag{5}$$

$$V_{e,j}^G = \frac{1}{2}(X_{e,j}^G + X_{\text{best},j}^G) + \phi_{e,j} \cdot (X_{\text{best},j}^G - X_{k,j}^G) \tag{6}$$

在式(5)和式(6)中，X_e 是精英解，X_k 是随机选择的个体，X_{best}是全局最优解。式(5)包含了精英解 X_e 的信息，可以应用于各个蜜蜂阶段；而式(6)不仅利用了精英解的信息，而且采用了当前最好的解 X_{best}并将其运用于跟随蜂阶段。式(5)、式(6)与 DFSABC 相结合形成一个新的 ABC 变体，即 DFSABC_elite，将其命名为智能蜂群算法(DFSABC_elite)。

3.3　基于智能蜂群算法的聚类模型的设计

基于智能蜂群算法的聚类是将 DFSABC_elite 和并行化的 K-means 算法相结合来实现聚类的过程。DFSABC_elite 由于较好地平衡了算法的勘探与开采能力，因此具有较强的跳出局部最优解的能力。而大量的研究成果表明：并行化的 K-means 算法对初始聚类中心敏感，虽然已经有较多篇文献采用粒子群[11]、差分[12]等算法优化并行化 K-means 的聚类中心，但由于 DFSABC_elite 具有较强的跳出局部最优解的能力，因此采用 DFSABC_elite 优化并行化 K-means 算法能够进一步提高并行化 K-means 的性能。其算法思想为：在每次迭代过程中，利用 DFSABC_elite 对聚类中心进行优化，然后再重新计算聚类中心，K-means 算法和 DFSABC_elite 算法交替进行，直到聚类结束。一方面，并行化 K-means 算法迭代加快了算法的收敛速度；另一方面，DFSABC_elite 的结合使算法降低了对初始聚类中心的依赖，增强了算法的鲁棒性。在 DFSABC_elite 中，将聚类中心定义为食物源，并定义两个指标(紧密度指标和分离度指标)作为食物源优劣度；同时，将适应度高的聚类中心定义为优劣度(食物源的好坏)高的食物源，这样可以将聚类问题和蜜蜂觅食行为定义成如表 1 所列的对应关系。

表 1　聚类与蜜蜂觅食行为的对应情况

Table 1　Corresponding situations between cluster and foraging behavior of bees

蜂群觅食行为	聚类问题
食物源位置	聚类中心
食物源优劣度 1	紧密度指标
食物源优劣度 2	分离度指标
高优劣度食物源	高适应度聚类中心

本文采用 CH index[13]作为聚类评价指标。CH index 是基于类内紧密度和类间分离度定义的聚类评价指标，其相关定义如下。

令类内紧密度为类内离差矩阵 W 的迹，如式(7)所示：

$$\text{tr}(W) = \sum_{i=1}^{k} \sum_{s \in c_i} d^2(s, Z_i) \tag{7}$$

令类间分离度指标为类间离差矩阵 B 的迹，如式(8)所示：

$$\text{tr}(B) = \sum_{i=1}^{k} n_i d^2(Z_i, z) \tag{8}$$

由此，可以得出 CH index 的方程，如式(9)所示：

$$CH(k) = \frac{\text{tr}(B)/(k-1)}{\text{tr}(W)/(n-k)} \tag{9}$$

其中，Z_i 是第 i 个类的类中心，z 为所有样本的中心，n_i 为第 i 个类的样本数，k 为聚类数，n 为样本总数。对于聚类问题，类间分离度越大，类内紧密度越小，$CH(k)$的值就越大，说明这样的划分越好。

本文提出的基于智能蜂群算法的聚类模型的具体实现步骤如下：

1)初始化数据集和相关参数，设定参数 N、聚类数 k、控制参数 $limit$ 以及最大迭代次数 MCN。

2)根据样本数据集和聚类数 k 来确定食物源向量维数 $L_i = k * d$，初始化蜂群产生 N 个食物源，其中 d 为样本维数。根据式(2)计算食物源适应度的值。

3)为食物源分配一个引领蜂并根据贪婪原则选择食物源，根据式(3)进行搜索并产生一个新食物源。

4)根据食物源适应度，由式(1)计算选择各个食物源的概率。跟随蜂再次利用式(5)进行邻域搜索，若发现适应度更高的食物源，则替换原引领蜂的旧食物源并转变成引领蜂。

5)若连续 $limit$ 次迭代后，引领蜂的适应度未得到进化，则对应的引领蜂转变成侦查蜂，并根据式(4)更新食物源。

6)对表示聚类中心的食物源进行一次并行化 K-means 迭代，按最近邻原则聚类划分，再重新计算每一个聚类的聚类中心，并根据贪婪原则更新蜂群。

7)记录当前找到的最优食物源，若当前的迭代次数小于 MCN，则转向步骤 3)进行下一次迭代；否则输出最优解作为聚类结果。

将 DFSABC_elite 和并行化 K-means 算法相结合实现的聚类算法称作改进的智能蜂群算法，其伪代码如算法 1 所示。

算法 1　改进的智能蜂群算法

```
1. 初始化实物源，使用式(2)计算实物源的适应度，引领蜂根据式(5)搜索相应的食物源
2. Flag=1
3. While 不符合结束条件 do
4.    for 每个引领蜂 do
5.       经过 limit 次迭代，若适应度未能成功更新，则使用式(4)更新食物源
6.       If flag then
7.          利用式(5)与式(6)分别完成 DFSABC_elite 的引领蜂与跟随蜂阶段
8.          Flag=0
9.       End if
10.      使用式(2)计算食物源适应度
11.      使用并行化 K-means 算法对新产生的食物源和其原始食物源进行迭代，并重新计算聚类中心，同时利用贪婪选择策略判定是否需要更新实物源，若食物源有改进，则将 flag 置为 1
12.   End for
13.   根据式(1)计算实物源被选择的概率 Pi
14.   For 每个跟随蜂 do
15.      If rand()<Pi then
16.         对比全局最优食物源，使用式(4)获取新实物源并产生候选解
17.         使用式(2)为每个食物源计算适应度 fiti
18.         使用并行化 K-means 算法对新产生的食物源和其原始食物
```

源进行迭代并重新计算聚类中心，同时利用贪婪选择策略判定是否需要更新实物源，若食物源有改进，则将 flag 置为 1

19. End if
20. End for
21. If 任何引领蜂变成侦查蜂 then
22. 侦查蜂随机产生新食物源
23. End if
24. 记录目前的最优食物源
25. End while

4 检测系统的原理与模型

4.1 检测原理及流程

基于改进的智能蜂群算法的 DDoS 攻击检测思想是基于异常检测的方法，首先通过聚类判断待检测数据流是否偏离正常数据流[14]，从而判定数据流是否异常；对于异常的数据流，再通过流量特征熵[15]与广义似然比较相结合的方法识别其是否是 DDoS 攻击数据流。图 2 给出了简单的聚类模型。

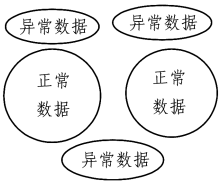

图 2 聚类模型

Fig. 2 Clustering model

根据 DDoS 攻击的特征，首先选取流量特征分布熵作为检测 DDoS 攻击数据流的特征，其流量特征即为报文的某个字段。一个时间段内的所有报文在其不同取值上呈现的一个分布，被称为流特征分布。Lakhina 等[16]提出了源 IP 地址(SrcIP)、目的 IP 地址(DstIP)、源端口号(SrcPort)、目的端口号(DstPort) 4 个流特征分布，其在不同的网络中表现出了不同的分散和集中特性，可以用于划分网络异常。例如，DDoS 攻击发生时会产生大量数据包，其中源 IP 地址比较分散而目的 IP 地址则比较集中。

设 X 表示流量特征，集合为 $\{n_i=1,2,\cdots,N\}$，n_i 为报文 X 取值为 X_i 的个数，N 为不同取值的个数。熵值 $H(X)$ 的定义如下：

$$H(X)=-\sum_{i=1}^{N}\frac{n_i}{S}lb(\frac{n_i}{S}) \tag{10}$$

其中，$S=\sum_{i=1}^{N}n_i$ 表示报文总数；lb 表示 $\log_2$；$H(X)$ 的取值范围为 $(0,\log_2 N)$，分布越分散则 $H(X)$ 越大。分布最集中时，即 X 只有一个取值，此时 $H(X)$ 取最小值 0；分布最分散，即 $n_1=n_2=\cdots=n_N$，$H(X)$ 取最大值 $\log_2 N$。分别计算 4 个流特征的熵，从而对异常数据流进行分类。

与 Peng 等[17]提出的通过监控流量中新 IP 地址数目变化识别 DDoS 攻击和 Sun 等[18]提出的流连接密度(FCD)的概念识别 DDoS 攻击的方法相比，流特征分布熵能充分刻画 DDoS 攻击的分布特性，即攻击报文从分散的源 IP 地址流向集中的目的 IP 地址，因此其可以作为 DDoS 攻击的一个很好的特征检测方法。

此外，特征匹配在网络流量监测中也是常用的手段之一。因此，我们采用特征匹配方法中的广义似然比较[18]和流特征分布熵相结合的方法来检测 DDoS 攻击。

首先，计算初步聚类的网络流量数据的均值 μ_z 和协方差矩阵的最大似然估计值 C_z。

$$\mu_z=\frac{1}{q}\sum_{i=1}^{n}m_i \tag{11}$$

其中，m_i 表示初步聚类流量的特征，q 表示流量特征个数。

$$C_z=\frac{1}{q}\sum_{i=1}^{n}(m_i-\mu_i)(m_i-\mu_i)^{\mathrm{T}} \tag{12}$$

对于未检测数据流量 x，计算其是否异常的判别因子 RM(广义似然)的方程式为：

$$RM(x)=(x-\mu_z)^{\mathrm{T}}C_z^{-1}(x-\mu_z) \tag{13}$$

最后，将判别因子作为阈值与流量特征熵进行比较来判别是否发生 DDoS 攻击：若 $RM(x)\geqslant H(x)$，则未发生 DDoS 攻击；否则发生 DDoS 攻击。

DDoS 攻击的检测流程如图 3 所示。

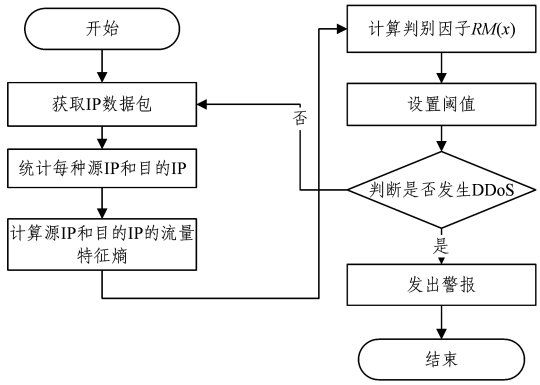

图 3 基于流量特征熵的 DDoS 攻击预警流程

Fig. 3 DDoS attackpre-warning process based on traffic feature entropy

4.2 DDoS 攻击检测模型的设计

本文提出的基于改进的智能蜂群算法聚类的 DDoS 攻击检测模型采用异常检测[19-20]的方法，异常检测是指通过监视系统使用的异常情况可以检测出违反网络安全的事件。该方法通过建立数据聚类模型，从待检测数据流中分离出异常数据流。对于异常的数据流，通过流量源 IP 地址和目的 IP 地址特征熵与广义似然比较相结合的方法来检测是 DDoS 攻击数据流，还是其他异常数据流。基于改进的智能蜂群算法聚类的 DDoS 攻击检测模型的结构如图 4 所示。

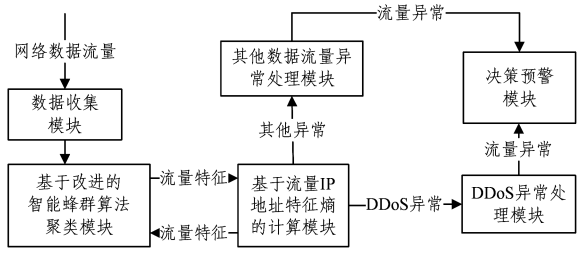

图 4 攻击检测模型

Fig. 4 Attack detection model

图 4 中的检测模型包含：数据收集模块、基于改进的智能

蜂群算法聚类模块、流量 IP 地址特征熵的计算模块、其他数据异常流量处理模块、DDos 异常处理模块和决策预警模块。

1）数据收集模块：收集网络中的数据流量并提取 IP 地址。

2）基于改进的智能蜂群算法聚类模块：根据改进的智能蜂群算法的原理对收集的数据流量进行聚类，从而区分正常流量和异常流量。

3）基于流量 IP 地址特征熵的计算模块：对于聚类后的异常流量，利用流量源 IP 地址和目的 IP 地址的特征熵和广义似然比较相结合的方法来检测是否发生 DDoS 攻击。

4）其他数据异常流量处理模块：对其他非 DDoS 的异常流量进行处理[21-22]。

5）DDoS 异常处理模块：对 DDoS 异常进行相应的处理[23]，本文主要讨论 DDoS 异常检测。

6）决策预警模块：对各种异常数据流量报警。

本系统的具体工作步骤如下：

1）运用数据收集模块收集当前网络数据流量并提取源 IP 地址和目的 IP 地址的特征。

2）将收集到的网络流量传递给基于改进的智能蜂群算法聚类模块，并进行数据聚类，继续监测正常类网络流量，对于异常类网络流量则进一步进行 DDoS 攻击检测。

3）根据基于流量源 IP 地址和目的 IP 地址的特征熵与广义似然比较相结合的检测方法，分离 DDoS 攻击数据流和其他异常数据流，并分别进行处理。

4）对所有异常数据流处理完毕后报警。

本文的检测系统利用智能蜂群算法与 K-means 聚类算法相结合的方式对网络数据流量进行聚类，并利用流量源 IP 和目的 IP 的特征熵与广义似然比较判别因子共同检测 DDoS 攻击，可以有效检测和及时处理网络中的 DDoS 攻击数据流。

5　实验与分析

为了验证本文提出的改进的智能蜂群算法的性能以及其在 DDoS 攻击检测中的效果，本文从两个方面进行实验：1）聚类的有效性与准确性，将基于改进的 DFSABC_elite 的聚类、基于结合并行化 K-means 的 ABC 和单独的并行化 K-means 的聚类进行仿真实验比较。2）DDoS 攻击检测效果，即将本文方法与其他 DDoS 攻击检测方法进行效果比较。实验中，选择 KDDCUP99 数据集[24]作为测试数据集，KDDCUP99 数据集是网络安全研究人员公认的数据集，作为研究入侵网络数据及应用算法优劣的评价标准，其为本文 DDoS 攻击入侵检测的研究奠定了坚实的基础。数据集的相关信息如表 2 所列。

表 2　KDDCUP99 数据集

Table 2　KDDCUP99 dataset

数据集	数据个数/万	属性个数	聚类数 K
KDD CUP99	约 50	39	4

首先，通过 MATLAB R2014a 对改进的 DFSABC_elite 和其他算法进行仿真实验，算法运行 10 次获得改进的 DFSABC_elite 的聚类准确率。将实验获得的结果与普通蜂群算法（ABC）结合并行化 K-means 改进的聚类和单独的并行化 K-means 算法聚类的准确率进行比较，结果如图 5 所示。

从图 5 中可看出，本文提出的基于改进的智能蜂群算法（DFSABC_elite）的准确率稳定在 87% 以上，最高可达到约 97%，在 3 种算法中是最优的。此外，普通蜂群算法（ABC）结合并行化 K-means 聚类的准确率总体稍低于改进的 DFSABC_elite，在 84% ～ 95% 之间浮动；而单独的并行化 K-means 算法的表现最差，准确率的浮动范围大概在 82% ～ 92%之间。由此可知，将聚类思想融入蜂群算法可以提高算法的聚类准确率，从而提高了检测 DDoS 攻击的效率。

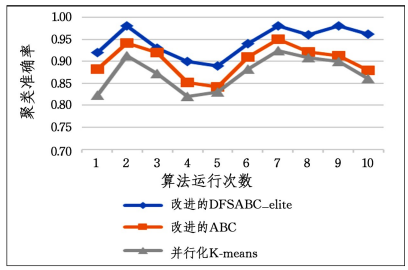

图 5　3 种算法的聚类准确率

Fig. 5　Clustering accuracy of three algorithms

为验证本文算法最佳聚类数的有效性，通过实验对分离度和紧密度进行跟踪记录，实验结果如图 6 和图 7 所示。

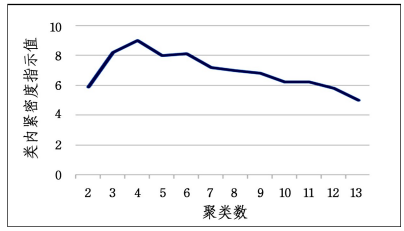

图 6　类内紧密度

Fig. 6　Intra-class compactness

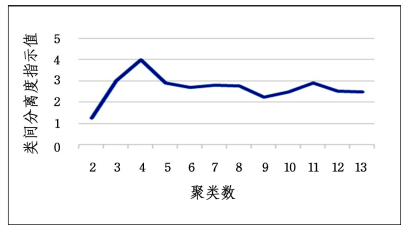

图 7　类间分离度

Fig. 7　Inter-class separation

紧密度是衡量类内相似度的重要指标，紧密度越大则类内的数据越多，各个数据之间的相似度越大；分离度是不同类之间的重要指标，分离度越大则类间的差异越大。由图 6 和图 7 可以看出，当聚类数为 4 时紧密度指标值和分离度指示值均为最高。实验结果显示两个指标能很好地用于确定最优聚类数。

为检测基于改进的智能蜂群算法的 DDoS 攻击检测模型的性能，在实验中搭建了实验平台，目标主机为 Linux 操作系统，利用实验室服务器上搭建的云平台，部署 4 台虚拟机，组成一个小型 Spark 集群[25-26]。在平台集群上对获取的网络流

量数据进行分析并将其作为训练样本，生成流量检测模型和基于智能蜂群算法的检测模型。由实验统计得到本文提出的检测模型和其他 DDoS 攻击检测模型进行检测的准确率和耗时结果，如表 3 和图 8 所示。

表 3 本文的 DDoS 检测模型与其他检测模型的结果对比

Table 3 Comparison of results of proposed DDoS detection model and other models

（单位：%）

DDoS 检测	改进的 ABC	K-means	本文算法
准确率	97.43	95.26	99.30
遗漏率	0.40	1.33	0.27
误报率	2.17	3.41	0.43

从表 3 可以看出，本文所提出的 DDoS 检测方法的准确率最高，检测准确率高达 99%；基于改进的 ABC 算法的 DDoS 检测方法的准确率则要低一些；而基于并行化 K-means 算法的 DDoS 检测方法的准确率最差。

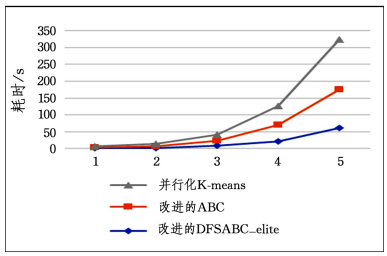

图 8 算法耗时对比

Fig. 8 Comparison of consumed time of different algorithms

为得到本文所提算法、结合并行化 K-means 的普通蜂群算法和单独的并行化 K-means 算法运行速率的对比，采用 KDD CUP99 数据集的训练集作为实验数据，选择数据集中的 10 种属性作为本文实验的测试数据的属性。采取 5 组数据量，分别为 1 万、5 万、10 万、20 万和 50 万，虽然每组数据的数量不同但数据分布是相似的。从图 8 可以看出，在数据量较小时，改进的 DFSABC_elite 的检测速率优势不明显，但是随着数据量的增大，本文所提方法的检测速率要远快于结合并行化 K-means 的改进的 ABC 算法。另外，改进的 ABC 算法的检测速率要稍快于并行化 K-means 算法的检测速度，说明 DFSABC_elite 对并行化 K-means 算法依赖聚类中心的缺陷有一定的改进。

结束语 本文在智能蜂群算法的基础上，提出了一种将聚类算法与智能蜂群算法(DFSABC_elite)相结合的新型聚类方法，可以将智能蜂群算法觅食行为模型与聚类算法的迭代相结合。在聚类的过程中，根据蜂群在食物源上的贪心选择的原理进行优化聚类，并使用并行化 K-means 迭代重新计算聚类，通过两种算法相互交替来获得最优的聚类，从而在 DDoS 检测过程中提高流量的区别分类的效率和准确率。实验表明，该方法在聚类的准确率上优于其他聚类算法，并且能有效获得最优聚类数；流量特征分布熵与广义似然比较判别因子相结合来检测 DDoS 攻击数据流，在 DDoS 攻击检测中不仅误报率低于其他 DDoS 攻击检测技术，而且 DDoS 攻击检测速度明显得到了提高。

参 考 文 献

[1] SOPHIA G A, GANDHI M. Stealthy DDoS detecting mechanism for cloud resilience system[C]// 2017 International Conference on Information Communication and Embedded Systems (ICICES). IEEE, 2017: 1-5.

[2] MODI C, PATEL D, BORISANIYA B, et al. A survey on security issues and solutions at different layers of Cloud computing [J]. The Journal of Supercomputing, 2013, 63(2): 561-592.

[3] HAN D Z, BI K, JIN J, et al. A DDoS Attack Detection System Based on Spark Framework[J]. Computer Science & Information Systems, 2017, 14: 28.

[4] GUELLIL I, AZOUAOU F. Arabic Dialect Identification with an Unsupervised Learning (Based on a Lexicon). Application Case: ALGERIAN Dialect[C]// Computational Science and Engineering. IEEE, 2017: 724-731.

[5] SAIDA I B, KAMEL N, OMAR B. A New Hybrid Algorithm for Document Clustering Based on Cuckoo Search and K-means [M]// Recent Advances on Soft Computing and Data Mining. Springer International Publishing, 2014: 59-68.

[6] NIU B, DUAN Q, LIU J, et al. A population-based clustering technique using particle swarm optimization and k-means[J]. Natural Computing, 2016, 16(1): 1-15.

[7] TEODOROVIĆD, ORCO M D. Advanced OR and AI Methods in Transportation BEE Colony Optimization — A Cooperative Learning Approach to Complex Transportation PRoblems[C]// Proceedings of the 16th Mini — EURO Conference and 10th Meeting of EWGT. 2008.

[8] KARABOGA D, BASTURK B. On the performance of artificial bee colony (ABC) algorithm[J]. Applied Soft Computing, 2008, 8(1): 687-697.

[9] ZHAO W, MA H, HE Q. Parallel k-means clustering based on mapreduce[C]// IEEE International Conference on Cloud Computing. Springer Berlin Heidelberg, 2009: 674-679.

[10] CUI L, LI G, LIN Q, et al. A novel artificial bee colony algorithm with depth-first search framework and elite-guided search equation[J]. Information Sciences, 2016, 367-368: 1012-1044.

[11] PRÍNCIPE J C, MIIKKULAINEN R. Advances in self-organizing maps[C]// Advances in Self-Organizing Maps: 7th International Workshop, WSOM 2009. Springer, 2009.

[12] DU B, SUN Y, CAI S, et al. Object Tracking in Satellite Videos by Fusing the Kernel Correlation Filter and the Three-Frame-Difference Algorithm[J]. IEEE Geoscience & Remote Sensing Letters, 2017, PP(99): 1-5.

[13] CAO Y C, CAI Z Q, SHAO Y B. An improved artificial bee colony clustering algorithm based on K-means [J]. Journal of Computer Applications, 2014, 34(1): 204-207. (in Chinese)
曹永春，蔡正琦，邵亚斌. 基于 K-means 的改进人工蜂群聚类算法[J]. 计算机应用，2014，34(1)：204-207.

[14] YANG X R, HAN B, SUN Z G, et al. SDN-based DDoS Attack Detection with Cross-Plane Collaboration and Lightweight Flow

Monitoring[C]// Global Communications Conference. 2017.

[15] YAN Y, ZHANG S, TANG J, et al. Understanding characteristics in multivariate traffic flow time series from complex network structure[J]. Physica A: Statistical Mechanics & Its Applications, 2017, 477.

[16] LAKHINA A, CROVELLA M, DIOT C. Mininganomalies using traffic feature distributions[C]// Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. ACM, 2005: 217-228.

[17] PENG T, LECKIE C, RAMAMOHANARAO K. Proactively detecting distributed denial of service attacks using source IP address monitoring[C]// International Conference on Research in Networking. Springer Berlin Heidelberg, 2004: 771-782.

[18] SUN Q D, ZHANG D Y, GAO P. Distributed Denial of Service Attack Detection Based on Time Series Analysis[J]. Chinese Journal of Computers, 2005, 28(5): 767-773. (in Chinese)
孙钦东，张德运，高鹏. 基于时间序列分析的分布式拒绝服务攻击检测[J]. 计算机学报，2005，28(5)：767-773.

[19] XU Z, ZHU S, FU B, et al. Motion coherence based abnormal behavior detection [C] // Control and Decision Conference. IEEE, 2017: 214-218.

[20] HAN D Z, BI K, XIE B L, et al. An Anomaly Detection on the Application-Layer -Based QoS in the Cloud Storage System[J]. Computer Science and Information Systems, 2016, 13(2): 659-676.

[21] YUAN Y, WANG D, WANG Q. Anomaly Detection in Traffic Scenes via Spatial-Aware Motion Reconstruction [J]. IEEE Transactions on Intelligent Transportation Systems, 2017, 18(5): 1198-1209.

[22] CHANG R K C. Defending against flooding-based distributed denial-of-service attacks: a tutorial[J]. IEEE Communications Magazine, 2002, 40(10): 42-51.

[23] LEMON J. Resisting SYN Flood DoS Attacks with a SYN Cache[C]// Bsdcon Conference. 2002.

[24] WU J S, ZHANG W P, MA Y. The Data Analysis of KDD-CUP99 Data Set [J]. Computer Applications and Software, 2014 (11): 321-325. (in Chinese)
吴建胜，张文鹏，马垣. KDDCUP99 数据集的数据分析研究[J]. 计算机应用与软件，2014(11)：321-325.

[25] AHMED H, ISMAIL M A, HYDER M F, et al. Performance Comparison of Spark Clusters Configured Conventionally and a Cloud Service[J]. Procedia Computer Science, 2016, 82: 99-106.

[26] ZAHARIA M, DAS T, LI H, et al. Discretized Streams: An Efficient and Fault-Tolerant Model for Stream Processing on Large Clusters[C]// Usenix Conference on Hot Topics in Cloud Computing. USENIX Association, 2012.

(上接第 122 页)

参 考 文 献

[1] SANDHU R S, COYNE E J, FEINSTEINH L, et al. Role-based access control models[J]. Computer, 1996, 29(2): 38-47.

[2] COYNE E J. Role engineering[C]// Proceedings of the first ACM Workshop on Role-based access control. ACM, 1996.

[3] MITRA B, SURAL S, VAIDYA J, et al. A survey of role mining [J]. ACM Computing Surveys (CSUR), 2016, 48(4): 1-37.

[4] SCHLEGELMILCH J, STEFFENS U. Role mining with ORCA [C]// Proceedings of the tenth ACM symposium on Access control Models and Technologies. ACM, 2005: 168-176.

[5] ZHANG D N, RAMAMOHANARAO K, EBRINGER T. Role engineering using graph optimization[C]// Proceedings of the 12th ACM Symposium on Access Control Models and Technologies. 2007: 139-144.

[6] GUO Q, VAIDYA J, ATLURI V. The role hierarchy mining problem: discovery of optimal role hierarchies[C]// Computer Security Applications Conference. IEEE, 2008: 237-246.

[7] SARMAH A K, HAZARIKA S M, SINHA S K. Formal concept analysis: current trends and directions[J]. Artificial Intelligence Review, 2015, 44(1): 47-86.

[8] MOLLOY I, CHEN H, LI T, et al. Mining roles with multiple objectives[J]. ACM Transactions on Information and System Security (TISSEC), 2010, 13(4): 1-35.

[9] SOBIESKI S, ZIELINSKI B. Modelling role hierarchy structure using the Formal Concept Analysis[J]. Annales Umcs Informatica, 2010, 10(2): 143-159.

[10] KUMAR C. Designing role-based access control using formal concept analysis[J]. Security and Communication Networks, 2013, 6(3): 373-383.

[11] KUMAR C A, MOULISWARAN S C, LI J, et al. Role based access control design using triadic concept analysis[J]. Journal of Central South University, 2016, 23(12): 3183-3191.

[12] ZHANG L, ZHANG H L, HAN D J, et al. Theory and Algorithm for Roles Minimization Problem in RBAC Based on Concept Lattice[J]. Acta Electronica Sinica, 2014, 42(12): 2371-2378. (in Chinese)
张磊，张宏莉，韩道军，等. 基于概念格的 RBAC 模型中角色最小化问题的理论与算法[J]. 电子学报，2014，42(12)：2371-2378.

[13] GANTER B, WILLE R. Formal concept analysis: mathematical foundations[M]. New York: Springer Science & Business Media, 2012.

[14] ZHI H L. Extended Model of Formal Concept Analysis Oriented for Heterogeneous Data Analysis[J]. Acta Electronica Sinica, 2013, 41(12): 2451-2455. (in Chinese)
智慧来. 面向异构数据分析的形式概念分析扩展模型[J]. 电子学报，2013，41(12)：2451-2455.

[15] GODIN R, MINEAU G, MISSAOUI R, et al. Méthodes de classification conceptuelle basées sur les treillis de Galois et applications[J]. Revued'Intelligence Artificielle, 1995, 9: 105-137.

[16] GANTER B. Two Basic Algorithms in Concept Analysis[C]// International Conference on Formal Concept Analysis. Springer-Verlag, 2010: 312-340.