# NLOF:一种新的基于密度的局部离群点检测算法

王敬华 赵新想 张国燕 刘建银
(华中师范大学计算机学院 武汉 430079)

**摘 要** 基于密度的局部离群点检测算法(LOF)的时间复杂度较高且不适用于大规模数据集和高维数据集的离群点检测。通过对LOF算法的分析,提出了一种新的局部离群点检测算法NLOF,该算法的主要思想如下:在数据对象邻域查询过程中,尽可能地利用已知信息优化邻近对象的邻域查询操作,有关邻域的计算查找都采用这种思想。首先通过聚类算法DBSCAN对数据集进行预处理,得到初步的异常数据集。然后利用LOF算法中计算局部异常因子的方法计算初步异常数据集中对象的局部异常程度。在计算数据对象的局部异常因子的过程中,引入去一划分信息熵增量,用去一划分信息熵差确定属性的权重,对属性的权值做具体的量化,在计算各对象之间的距离时采用加权距离。在真实数据集上对NLOF算法进行了充分的验证。结果显示,该算法能够提高离群点检测的精度,降低时间复杂度,实现有效的局部离群点的检测。

**关键词** 数据挖掘,离群点检测,信息熵,聚类
**中图法分类号** TP311.13 **文献标识码** A

## NLOF:A New Density-based Local Outlier Detecting Algorithm

WANG Jing-hua ZHAO Xin-xiang ZHANG Guo-yan LIU Jian-yin
(Academy of Computer Science, Central China Normal University, Wuhan 430079, China)

**Abstract** The time complexity of the density-based outlier detecting algorithm (LOF algorithm) is not ideal, which effects its applications in large scale datasets and high dimensional datasets. Under such circumstances, a new density-based outlier detecting algorithm (NLOF algorithm) was introduced. The main idea of the NLOF algorithm is as follows: the known information is used as much as possible to optimize the neighborhood query operation of adjacent objects in the process of neighborhood searching of a data object. This method is adopted in neighborhood computing and searching in this paper. Firstly, clustering algorithm is taken as a preprocessing, and local outlier factors are calculated only for the data objects out of clusters. Secondly, the local outlier factors for the data objects out of clusters are calculated in the same way as calculating the local outlier factors in LOF algorithm. Leave-one partition information gain is introduced in the process of calculating the local outlier factor. The weight of attribute is determined by leave-one partition information gain. The weighted distance is used in calculating distances between objects. Extensive experimental results show the advantages of the proposed method. NLOF algorithm can improve the outlier detection accuracy, reduce the time complexity and realize the effective local outlier detection.

**Keywords** Data mining, Outlier detection, Information entropy, Clustering

## 1 引言

离群点检测的目的是在大量的、复杂的数据集合中消除噪音数据或发现潜在的、有意义的知识。Hawkins对离群点做了如下定义,"一个离群点是一个观察点,它偏离其他观察点如此之大以致引起怀疑是由不同机制生成的"[1]。离群点检测可以广泛地应用在电子商务犯罪和信用卡欺诈的侦查、网络入侵检测、生态系统失调检测、公共卫生、医疗和天文学上稀有的未知种类的天体发现等领域中。罕见事件通常比常规事件更有吸引力,因为其往往蕴含着真实、有价值却又出乎人们意料的知识。因此这些领域都十分重视离群点的研究。

离群点检测算法有很多,大致上可以分为:基于分布/深度、距离、密度、聚类和偏离等。随着机器学习、人工智能、模式识别等领域的发展和进步,越来越多新颖、有效的离群点检测技术和方法不断被提出,例如:利用人工神经网络进行离群点检测[2]、基于分区的离群点检测方法[3]、基于模糊粗糙集的离群点检测方法[4]、利用自组织映射技术进行离群点检测[5]等。

基于统计学(Distribution-Based)(基于分布的和基于深度的)的离群点检测把离群点看作是一种二元性质,依赖于给

到稿日期:2012-12-30 返修日期:2013-03-11 本文受国家自然科学基金项目(61170017)资助。
王敬华(1965—),男,硕士,副教授,主要研究方向为数据挖掘、现代信息系统,E-mail:jhuawang@126.com;赵新想(1988—),女,硕士生,主要研究方向为数据库与数据挖掘;张国燕(1986—),女,硕士生,主要研究方向为数据库与数据挖掘;刘建银(1986—),男,硕士生,主要研究方向为数据库与数据挖掘。

定数据集的全局分布，但是当给定的数据分布不均匀时，这些方法就不适用了；基于距离(Distance-Based)的离群点检测方法不需要事先了解数据的分布模式，同时可以适用于任意维度的数据集，但是需要用户选取合理的距离与比例参数以保证算法的效果，同时算法在高维数据集上存在一定的效率问题；基于聚类(Clustering)的方法中，对象往往是因为不属于任何类或者属于对象数量很少的类而被理解为离群点，离群点是作为聚类算法的副产物而被发现的；基于深度(Depth-Based)的方法在二维或三维数据集上比较有效，但由于算法的效率问题，其不适合高维数据集上的离群点检测；基于密度(Density-Based)的局部离群点检测方法不将离群点看作一种二元性质，而是转向量化地描述数据对象的离群程度，其能在数据分布不均匀的情况下准确地发现离群点。Breuing 等人提出了局部离群因子的概念 LOF[6]，这是一种基于密度的方法。自从 LOF 算法出现后，出现了很多离群度的度量方法，比较典型的有 COF 算法[7]、基于局部信息熵的加权子空间离群点检测算法[8]、多粒度偏差因子[9]、MDEF 算法、SLOM 算法和局部空间离群度测度等方法。总的来说，现有的基于离群度的局部离群点挖掘算法主要区别在于邻域的确定方法和离群度的计算方法不同[10]。但是上述算法都存在以下问题：计算复杂度高，检测结果的精度和重复计算的次数依赖于用户给定的参数。

由于上述基于离群度度量的离群点检测算法的计算量较大，本文提出了一种新的基于密度的局部离群点检测算法：NLOF 算法。本文的贡献主要在以下几个方面：

(1)局部异常因子 LOF 用于表征数据集中每个数据对象的异常程度，并且这种异常是局部的，与所求数据对象一定范围内的邻居分布有关。NLOF 算法充分将这一思想应用到计算局部异常因子的实际操作中，即尽可能地利用已知信息优化邻近对象的邻域查询操作，以提高运行效率。

(2)利用聚类算法 DBSCAN，通过检查数据集中每个对象的 $\varepsilon$-邻域来寻找聚类(在邻域查询优化后得到的 *k-distance* 邻域中进行查询)，得到初步的异常数据集。然后用 LOF 算法中计算局部异常因子的方法(在邻域查询优化后得到的对象的 *k-distance* 邻域中重新计算该对象到邻域内其他点的加权距离 $d(p,q,w)$ 和该点的第 $k$ 加权距离，两个对象之间的距离的度量用的是加权明考斯基距离，其中对象属性的权值由去一划分信息熵增量确定)计算初步异常数据集中对象的局部异常程度。

(3)在计算数据对象的局部异常因子的过程中，引入去一划分信息熵增量，用去一划分信息熵差确定属性的权重，对属性的权值做具体的量化，在计算各对象之间的距离时采用加权距离，能够有效提高离群点检测的精度。

(4)针对本文提出的方法在真实数据集上进行了充分的验证，证实了该算法的优越性。

## 2 预备知识

### 2.1 局部离群点检测算法 LOF

LOF 算法[11]为每一个数据对象赋予一个表征该数据对象离群程度的因子，这样离群数据不再是一个二性数值，而是一种趋向性的描述。LOF 算法的一些基本概念如下。

**定义 1**(对象 $p$ 的第 $k$ 距离，*k-distance*) 对任意的自然数 $k$，定义 $p$ 的第 $k$ 距离为 $p$ 与某个对象 $o$ 之间的距离，记为 *k-distance*($p$)，这里的 $o$ 满足以下条件：

(1)至少存在 $k$ 个对象 $o'\in D\backslash\{p\}$ 满足 $d(p,o')\leqslant d(p,o)$；

(2)至多存在 $k-1$ 个对象 $o'\in D\backslash\{p\}$ 满足 $d(p,o')<d(p,o)$。

其中，对象 $p$ 和对象 $o$ 之间的距离记作 $d(p,o)$。

**定义 2**(对象 $p$ 的第 $k$ 距离邻域，$N_{k\text{-}distance}(p)$) 已知数据对象 $p$ 的 *k-distance*($p$)，对象 $p$ 的第 $k$ 距离邻域是所有与 $p$ 的距离不超过 *k-distance*($p$)的数据对象的集合，即：

$$N_{k\text{-}distance}(p)=\{q\mid d(p,q)\leqslant k\text{-}distance(p)\} \tag{1}$$

式中，$N_{k\text{-}distance}(p)$在本文中简记为 $N_k(p)$。

**定义 3**(对象 $p$ 相对于对象 $o$ 的可达距离，*reach-dist*($p$,$o$)) 设 $k$ 为一自然数，对象 $p$ 相对于对象 $o$ 的可达距离(见图 1)为：

$$reach\text{-}dist(p,o)=\max\{k\text{-}distance(o),d(p,o)\} \tag{2}$$

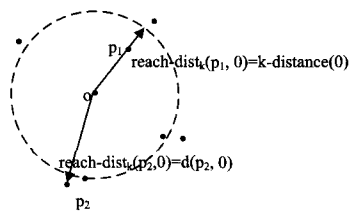


图 1 可达距离示意图

**定义 4**(对象 $p$ 的局部可达密度，$lrd_k(p)$) 对象 $p$ 的局部可达密度是对象 $p$ 与它的 $N_{k\text{-}distance}(p)$的平均可达距离的倒数，计算公式为：

$$lrd_k(p)=1/\left[\frac{\sum reach-dist_k(p,o)}{|N_k(p)|}\right] \tag{3}$$

**定义 5**(对象 $p$ 的局部异常因子，$LOF_k(p)$) 对象 $p$ 的局部异常因子表征对象 $p$ 为异常的程度，局部异常因子越大，对象 $p$ 为异常的可能性越大；反之，则越小。计算公式为：

$$LOF_k(p)=\frac{\sum\limits_{o\in N_k(p)}\frac{lrd_k(o)}{lrd_k(p)}}{|N_k(p)|} \tag{4}$$

## 3 NLOF 算法

### 3.1 邻域查询优化

LOF 算法的时间复杂度主要取决于邻域查询操作。局部异常因子 LOF 用于表征数据集中每个数据对象的局部异常程度，这种异常与所求数据对象一定范围内的邻居分布有关，是局部的。NLOF 算法充分将这一思想应用到计算局部异常因子的实际操作中，即在数据对象邻域查询过程中，利用已知信息优化邻近对象的邻域查询操作，从而提高查询效率。

如图 2 所示：圆 1 是以对象 $a$ 为圆心、半径为 $r$ 的圆；圆 3 是以 $a$ 为圆心、半径为 $3r$ 的圆；圆 2 是以 $e$ 为圆心($e$ 在圆 1 的边界上)、半径为 $2r$ 的圆。假设 $k$ 取值为 5，则有 $N_{k\text{-}distance}(a)=\{b,c,d,e,f\}$。按照 LOF 算法的思想，获得 $N_{k\text{-}distance}(a)$和相关的距离之后，就完成了对象 $a$ 的邻域查询，该对象邻域查询结束后这些信息被放弃，应该选取另外一个点进行邻域查询。实际上，对象 $a$ 的邻域内的对象的邻域查询范围只需要在圆 3 内部进行，而不用在整个数据集中进行查询。

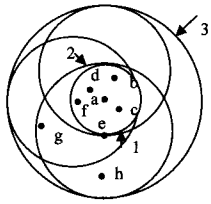


图 2　邻域查询优化示意图

但是上面的邻域查询方法还可以进一步优化，因为在以对象 $a$ 为圆心、$2\times dist(a,b)+Mpd(a)$ 为半径的区域内就包含了属于 $N_{MinPts}(b)$ 的所有对象。优化后的邻域查询步骤如下：

(1)从数据集中顺序选择一个尚未处理的对象 $a$，在数据集 $D$ 内进行查询，获得 $k\text{-}distance(a)$、$N_{k\text{-}dsitance}(a)$ 和所有其它对象与 $p$ 之间的距离。

(2)在 $N_{k\text{-}dsitance}(a)$ 内选取与 $a$ 的距离最小并且没有处理过的对象 $o$，在以公式 $dist(a,s)\leqslant 2\cdot d(a,o)+k\text{-}distance(a)$ 所表示的范围内进行查询。然后从与 $a$ 的距离次小且没有处理过的对象循环上述邻域查询过程，直至满足给定的结束条件，即查询区域内对象数量超过某指定阈值，转至步骤(1)，开始下一轮循环。

### 3.2　数据对象预处理

(1)DBSCAN 算法通过检查数据集中每个对象的 ε-邻域来寻找聚类，对象的 ε-邻域查询过程采用 3.1 节中的邻域查询优化算法，在邻域查询优化算法的第(1)步得到的对象的 $k\text{-}distance$ 邻域中来查询该对象的 ε-邻域。

(2)如果一个对象 $a$ 的 ε-邻域包含多于 $MinPts$ 个对象，则创建以 $a$ 为核心对象的新簇。反复寻找这些核心对象直接密度可达的对象，这一过程可能会涉及一些密度可达簇的合并。

(3)当没有新的对象可以添加到任何簇时，算法结束。

(4)不属于任何簇的对象的集合为初步异常数据集。

由于聚类内部的数据引起 LOF 值近似为 1，不是值得关注的离群点，因此只对聚类外部的标记为噪声的初步异常数据集中的对象进行分析处理。

### 3.3　数据对象离群属性权值量化

#### 3.3.1　信息熵

信息熵[12]是信息论中信息有用程度的体现，是信息的不确定性的一种度量。设 $x$ 是一个随机变量，其取值集合为 $S(x)$，$P(x)$ 表示 $x$ 的概率函数，则 $x$ 的信息熵定义为：

$$E(X)=-\sum_{X\in S(X)}p(x)\log(p(x)) \tag{5}$$

熵值越大，变量的不确定性就越大；熵值越小，变量的不确定性就越小。

#### 3.3.2　去一划分信息熵增量

设 $S$ 为对象集合 $D$ 的一个子集，且 $|S|>1$，$x$ 是 $S$ 中的一个对象。$S$ 被对象 $x$ 划分为两个部分：$S-\{x\}$ 和 $\{x\}$，记为 $\hat{C}=\{C_1,C_2\}$，得到

$$E(\hat{C})=\sum_{k=1,2}\left(\frac{|C_k|}{|D|}(E(C_k))\right) \tag{6}$$

$$\Delta(x)=E(S)-E(\hat{C}) \tag{7}$$

式中，$E(S)-E(\hat{C})$ 为集合 $S$ 的去一划分信息熵增量，记作 $\Delta(x,S)$，$S$ 明确的情况下，下文中简记为 $\Delta(x)$。从公式可以看出，$\Delta(x)$ 表示子集 $S$ 被划分前后的信息熵的变化，$\Delta(x)$ 值越大，表示把对象 $x$ 从集合 $S$ 中划分出去后使得整个数据集的“混乱”或“不确定性”减少得越多。本文中我们将对象的属性看作一个集合，用 $\Delta(x)$ 来对集合中对象的属性权值做具体的量化。

#### 3.3.3　加权距离

设 $p,q\in D$，其中 $f(p)$ 和 $f(q)$ 是第 $i(i=1,2,\cdots,d)$ 维属性的值，则对象 $p$ 与 $q$ 的 $d$-维属性的加权距离为：

$$d(p,q,w_j)=\sqrt{\sum_{i=1}^{d}w_j(f(p)-f(q))^2} \tag{8}$$

式中，$w_j$ 是第 $j$ 维属性的权值，且 $0\leqslant w_j\leqslant 1$(这里对 $w_j$ 做了归一化处理)。

### 3.4　NLOF 算法描述

输入：$d$ 维数据集 $D$，$k$，簇中最小对象数目 $MinPts$，半径 ε，查询区域内对象数量阈值 $\eta$，离群因子阈值 $\xi$

输出：数据集 $D$ 的离群点集合

算法过程：

(1)利用 3.2 节中数据对象预处理方法对数据集进行预处理，得到初步离群数据集。

(2)在初步异常数据集中选择一个没被处理过的点，利用式(6)、式(7)计算初步该对象的各个属性的去一划分信息熵增量 $\Delta(p_j)$(对于 $p$，在 $d$ 维属性中第 $j$ 维属性的权值记为 $\Delta(p_j)$)。

(3)在该对象的第 $k$ 距离邻域内，利用式(8)重新计算该对象到邻域内其他点的加权距离 $d(p,q,w)$ 和该点的第 $k\text{-}weighted\text{-}distance$(两个对象之间的距离的度量用的是加权明考斯基距离，其中对象属性的权值由去一划分信息熵增量确定)。

(4)利用式(3)计算各对象的局部可达密度。

(5)利用式(4)计算各对象的局部离群因子。

(6)若某个对象的离群因子 LOF 大于给定的阈值 $\xi$，则输出。

### 3.5　算法分析

基于离群度的局部离群点挖掘算法的复杂度主要取决于于邻域的求解。LOF 算法的时间复杂度主要取决于邻域查询操作，全部邻域查询使得 LOF 算法的时间复杂度为 $O(n^2)$。本文提出的局部离群点检测 NLOF 算法，首先通过聚类算法 DBSCAN 对数据集进行预处理，得到初步的异常数据集，DBSCAN 算法是通过检查数据集中每个对象的 ε-邻域来寻找聚类，在邻域查询优化算法得到的对象的第 $k$ 距离邻域中来查询该对象的 ε-邻域。对初步异常数据集中的数据的离群度的计算，是在邻域查询优化算法得到的对象的 $k$ 距离邻域中重新计算该对象到邻域内其他点的加权距离 $d(p,q,w)$ 和该点的第 $k\text{-}weighted\text{-}distance$，用去一划分信息熵增量确定对象属性的权重。下面从 7 个角度对 NLOF 算法、LOF 算法和 DLOF 算法[11]进行比较分析。

(1)NLOF 算法中通过聚类算法 DBSCAN 对数据集进行预处理，使得离群点挖掘分析对象更有针对性；LOF 算法是对整个数据集进行离群点挖掘，缺乏针对性；DLOF 算法也是对整个数据集进行离群点挖掘，缺乏针对性。

(2)随着数据量的增多,聚类规模可能增大,原本的噪声数据也可能聚集为新的聚类,所以 NLOF 算法未必会因为数据量的增多而使运行时间非线性递增。所以 NLOF 算法在大规模数据集上很有优势,并且数据集规模越大,优势会越明显。但是 NLOF 算法的计算复杂度与数据集中对象的个数和分布情况密切相关,很难从理论上衡量其复杂度;LOF 算法会因数据量的增多,运行时间迅速增多;DLOF 算法中虽然要计算整个数据集的各对象之间的加权距离,但在两步优化中采取了以空间换取时间的方法,有效地降低了时间复杂度。

(3)NLOF 算法中对象的 $\varepsilon$-邻域的查找是在邻域查询优化算法得到的第 $k$ 距离邻域中进行的,可以大大缩小查询范围。虽然进行了二次查询,但两次查询都是在对象的单独的查询区域内进行的,总体上相比 LOF 算法能够很有效地缩短运行时间,即 NLOF 算法的时间复杂度肯定会小于 $O(n^2)$。

(4)NLOF 算法相比 LOF 算法能够有效地提高算法的检测精度;在 DLOF 算法中对象属性权值 $\lambda$ 是人为输入的,其检测精度低于 NLOF。

(5)参数影响方面,LOF 算法的运行时间只与数据集大小有关,与参数无关,但是检测精度与参数有关;NLOF 算法中参数的作用影响很大,研究参数与运行时间和运行结果的关系,正确合理地找出参数值是下一步要研究的内容;DLOF 算法对参数的依赖性也很大,且对象属性权值参数 $\lambda$ 是由专家确定的,有很大的人为因素,影响了离群点检测的精度。

(6)数据集规模方面,NLOF 和 DLOF 适于大规模数据集的离群点检测;LOF 算法不适于大规模数据集的离群点检测。

(7)数据集维数方面,LOF 算法中,随着数据集维数的增加,检测精确度和效率可能急剧恶化;随着数据集维数的持续增加,NLOF 算法的效率要比 DLOF 好,但是精确度可能会有所下降。

NLOF 算法中的邻域优化查询过程的运行时间与数据集中数据对象的个数和分布情况有很大关系,很难从理论上衡量其复杂度。所以本文采用比较权威的数据集进行测试,以从实验的角度进行对比。

## 4 实验结果分析

本节通过实验对 LOF 算法、DLOF 算法、NLOF 算法的执行效率、检测精确度和对数据维度的伸缩性来进行分析和对比。

实验测试环境:Intel Core22,10GHz CPU,2G 内存,操作系统为 windowsXP,编程与语言为 java 平台 myeclipse6.5。

实验所用数据为网络入侵检测数据集 KDD-CUP1000 和 SEQUIOA2000 数据集,该数据集中的数据对象分为 5 大类,包括正常的连接、各种入侵和攻击等。为了进行实验,对 KDD-CUP1000 数据进行适当的修改,使得攻击(即利群点)占数据集的 3%。选择了其中的 20 个连续值属性维,对非数值属性维进行数值化处理。

(1)使用离群点检测精度衡量算法性能。精确度 = $\frac{\text{正确找到的离群点数}}{\text{离群点总数}}$。

由实验结果(见图 3)看出,本文所提出的 NLOF 算法比 LOF 算法和 DLOF 算法精确很多。

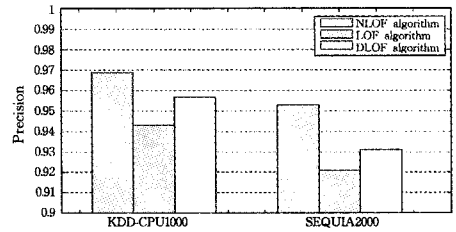


图 3 3 种算法的精确度对比

(2)使用计算复杂度衡量算法性能。由于无法从理论上衡量 NLOF 算法,因此采用权威的数据集进行测试。实验数据为 SEQUIOA2000 数据集。NLOF 算法中(4)—(6)步与 LOF 算法的不同之处在于在对象的第 $k$ 距离邻域内重新计算了该对象到邻域内其他点的加权距离 $d(p,q,w)$ 和该点的第 *k-weighted-distance*(两个对象之间的距离的度量用的是加权明考斯基距离,其中对象属性的权值是由去一划分信息熵增量确定)。这 3 步的实现相比 LOF 多了一些运行时间。但对于 60000 条数据的处理,NLOF 算法只用了 LOF 算法大约 46.8%的时间,$k$ 取值为 50。从实验可知,本文提出的算法能够有效地降低离群点检测的计算复杂度。实验结果转换为线形图,如图 4 所示。

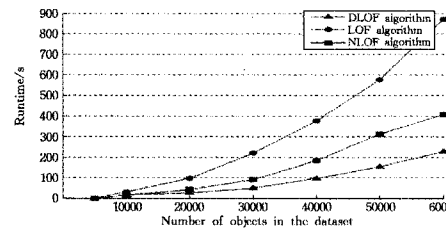


图 4 算法执行效率对比

(3)为了测试算法对数据维度的伸缩性,采用模拟数据集,将维度参数设置为 5、10、15、20、25、30、35,对 LOF 算法、DLOF 算法和 LOF 算法的运行情况进行了分析。

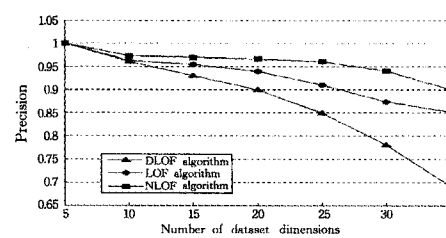


图 5 3 种算法的维度伸缩性对比

由图 5 可以看出,NLOF 算法相比其它两种算法对数据集维数有更好的伸缩性。

(4)随着数据集维数的增加,NLOF 算法的效率将高于 DLOF 算法,但是 NLOF 算法的检测精度可能会有所下降,下一步将在试验中进行验证。参数与运行时间的关系、参数和运行结果之间的关系和如何正确合理地找出参数值是下一步要研究的内容。

**结束语** 由实验结果可以看出,NLOF 算法是有效可行的。本文提出的 NLOF 算法利用邻域查询优化方法降低邻域查询的计算复杂度。通过聚类算法 DBSCAN 对数据集进行预处理,得到初步的异常数据集,使得离群点挖掘对象更有针对性。在计算数据对象的局部异常因子的过程中,引入去一划分信息熵增量,用去一划分信息熵差确定属性的权重,对属性的权值做具体的量化,在计算各对象之间的距离时采用

加权距离。经验证，NLOF 算法能够有效提高离群点检测精度，降低计算复杂度。但是在 NLOF 算法中，参数 $k$ 的作用重大，直接影响到邻近点的邻域查询范围。因此，下一步将研究参数与运行时间、运行结果之间的关系，找出自动提供合理的各参数值的方法。

## 参 考 文 献


[1] Hawkins D. Identification of Outliers [M]. Londen: Chapman and Hall, 1980: 188

[2] Han Sang-jun, Cho S-B, et al. Evolutionary Neural Networks for Anomaly Detection Based on the Behavior of a Program [J]. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 2006, 36(3): 559-570

[3] Ramaswamy S, Rastogi R, Shim K. Efficient algorithms for mining outliers from large data sets[J]. ACM Sigmoid Record, 2000, 29(2): 427-438

[4] Hung Wen-liang, Yang Min-shen. An Omission Approach for Detecting Outliers in Fuzzy Regression Models [J]. Fuzzy Sets and Systems, 2006, 157(23): 3109-3122

[5] Liu Xiao-hui, Cheng Gong-xian, Wu J X. Analyzing Outliers Cautiously[J]. IEEE Transactions on Knowledge and Data Engineering, 2002, 14(2): 432-437

[6] Breunig M, Kriegel H P, Ng R, et al. LOF: Identifying Density-based Local Outliers, 2000[C]// Proc. of the ACM SIGMOD International Conference on Management of Data. [s. l.]: ACM press, 2000: 93-104

[7] Tang J, Chen Z, Fu A, et al. Enhancing effectiveness of outlier detections for low-density patterns, 2002[C]// Proceeding of Advances in Knowledge Discovery and Data Mining 6th Pacific Asia Conference, Lecture Notes in Computer Science. Taipei, China, 2002: 535-548

[8] Ni Wei-wei, Chen Geng, Lu Jie-ping, et al. Local Entropy Based Weighted Subspace Outlier Mining Algorithm[J]. Journal of Computer Research Development, 2008, 45(7): 1189-1194

[9] Papadimitirou S, Kitagawa H, Gibbons P B, et al. LOCI: Fast outlier detection using the local correlation integral [C]// Proc of the 19th Int Conf on Data Engineering. Los Alamitos: IEEE Computer Society, 2003: 315-326

[10] 薛安荣，鞠时光，何伟华，等. 局部离群点挖掘算法研究[J]. 计算机学报，2007，30(8)：1455-1463

[11] 胡彩平，秦小麟. 一种基于密度的局部离群点检测算法 DLOF [J]. 计算机研究与发展，2010，47(12)：2110-2116

[12] 张净，孙志挥，等. 基于信息论的高维海量数据离群点挖掘[J]. 计算机科学，2011，38(7)：148-161


(上接第 148 页)

重点，本文重点检测的是正在活动的恶意域名。

表 3 恶意域名及 C&C 的 IP 地址

| 成功域名 | C&C 地址 |
|---|---|
| jalkd. cn | 221. 8. 69. 25 |
| izgj. cn | |
| ywhweot. cn | |
| yelicsefau. org | 143. 215. 130. 33 |
| ycbptbn. org | 143. 215. 143. 11 |
| xvnfoedv. org | 149. 20. 56. 32 |
| xuhsjusol. org | 149. 20. 56. 33 |
| | 149. 20. 56. 34 |

### 4.2 最终检测结果

当失效域名聚类阈值为 0. 3、$\lambda \geqslant 10$，$\theta \geqslant 10$，$Sc_{ij}$，$Cc_{ij}$ 的阈值分别 0. 75，0. 5 时，对 5 天的数据进行分析，提取出 3 组 C&C 服务器地址，其中两组地址为表 3 中的 C&C 服务器地址，这两组地址连续 5 天都出现，但每天对应的域名不同，请求 IP 数量主机最多时达 467 个不同 IP 地址。另外 1 组(域名为 roish. com、IP 地址为：74. 117. 116. 65)只出现了 1 天，同时当天请求主机与另外两组的请求主机基本相同，并且域名对应的网站是合法网站，判断被提取的出原因是：DGA 算法生成域名与已有域名发生冲突。

**结束语** 本文通过研究主机请求域名的行为特征，对采用 DGA 技术的 Botnet 进行检测，通过实际环境的验证，可以有效地检测出感染主机集合及 C&C 服务器使用的 IP 地址集合。未来研究中，将结合 DGA 生成域名字符构成特征，对失效域名集合、C&C 服务器 IP 地址进行进一步的检测。

## 参 考 文 献


[1] Leder W. Know Your Enemy: Containing Conficker [R]. The Honeynet Project & Research Alliance, University of Bonn, Germany, 2009

[2] Royal P. On the kraken and bobax botnets[R/OL]. http://www.damballa. com/downloads/r_pubs/Kraken_Response. pdf, 2009

[3] Stone-Gross B, Cova M, Vigna G. Your Botnet is My Botnet: Analysis of A Botnet Takeover [C]// ACM Conference on Computer and Communications Security(CCS). 2009: 635-647

[4] Yadav S, Reddy A, Ranjan S. Detecting Algorithmically Generated Malicious Domain Names [A]// 10th Annual ACM Conference on Internet Measurement [C]. New York, USA, 2010: 48-61

[5] Stalmans E, Irwin B. A Framework for DNS Based Detection and Mitigation of Malware Infections on a Network [A]// Information Security South Africa(ISSA)[C]. 2011: 76-83

[6] Jiang N, Zhang Z. Identifying Suspicious Activities through DNS Failure Graph Analysis [A]// Network Protocols(ICNP), the 18th IEEE International Conference [C]. 2010: 144-153

[7] Yadav S, Reddy A N. Winning with DNS Failures: Strategies for Faster Botnet Detection [A]// 7th International ICST Conference on Security and Privacy in Communication Networks [C]. 2011: 133-145

[8] Hao S, Feamster N, Pandrangi. An Internet Wide View into DNS Lookup Patterns[R/OL]. http://labs. verisigninc. com/projects/malicious-domain-names. html, 2010

[9] Antonakakis M, Perdisci R, Dagon D, et al. Building A Dynamic Reputation System for DNS [A]// the Proceedings of 19th USENIX Security Symposium (USENIX Security' 10) [C]. 2010: 273-289

[10] Antonakakis M, Lee R, Dagon D. Detecting Malware Domains at the Upper DNS Hierarchy [A]// the Proceedings of 20th USENIX Security Symposium(USENIX Security '11)[C]. 2011: 23-46

[11] Bilge L, Kirda E, Kruegel C, et al. Exposure: Finding Malicious Domains using Passive DNS Analysis [A]// Proceedings of NDSS [C]. 2011: 1-17

[12] 黄彪，成淑萍，欧阳晨星，等. 无尺度网络下具有双因素的僵尸网络传播模型[J]. 计算机科学，2012，39(10)：78-81

[13] 冯丽萍，韩琦，王鸿斌. 具有变化感染率的僵尸网络传播模型[J]. 计算机科学，2012，39(11)：51-53