基于多粒度级联孤立森林算法的异常检测模型

杨晓晖，张圣昌
（河北大学网络空间安全与计算机学院，河北 保定 071002）

摘 要：孤立森林算法是基于隔离机制的异常检测算法，存在与轴平行的局部异常点无法检测、对高维数据异常点缺乏敏感性和稳定性等问题。针对这些问题，提出了基于随机超平面的隔离机制和多粒度扫描机制，随机超平面使用多个维度的线性组合简化数据模型的隔离边界，利用随机线性分类器的隔离边界能够检测更复杂的数据模式。同时，多粒度扫描机制利用滑动窗口的方式进行维度子采样，每一个维度子集均训练一个森林，多个森林集成投票决策，构造层次化集成学习异常检测模型。实验表明，改进的孤立森林算法对复杂异常数据模式有更好的稳健性，层次化集成学习模型提高了高维数据中异常检测的准确性和稳定性。

关键词：异常检测；孤立森林；隔离机制；多粒度扫描；随机超平面
中图分类号：TP181
文献标识码：A
doi: 10.11959/j.issn.1000−436x.2019132

Anomaly detection model based on multi-grained cascade isolation forest algorithm

YANG Xiaohui, ZHANG Shengchang
School of Cyber Security and Computer, Hebei University, Baoding 071002, China

Abstract: The isolation-based anomaly detector, isolation forest has two weaknesses, its inability to detect anomalies that were masked by axis-parallel clusters, and anomalies in high-dimensional data. An isolation mechanism based on random hyperplane and a multi-grained scanning was proposed to overcome these weaknesses. The random hyperplane generated by a linear combination of multiple dimensions was used to simplify the isolation boundary of the data model which was a random linear classifier that can detect more complex data patterns, so that the isolation mechanism was more consistent with data distribution characteristics. The multi-grained scanning was used to perform dimensional sub-sampling which trained multiple forests to generate a hierarchical ensemble anomaly detection model. Experiments show that the improved isolation forest has better robustness to different data patterns and improves the efficiency of anomaly points in high-dimensional data.

Key words: anomaly detection, isolation forest, isolation mechanism, multi-grained scanning, random hyperplane

1 引言

异常检测的作用是分类出与多数数据有不同行为模式的稀有数据。Grubbs[1]对异常点有如下定义：异常点是一种模式，在此模式下的数据点偏离了大部分数据点的模式特征，甚至不是同一种机制产生的。本文将异常点定义为分布稀疏且距离密度较高的数据簇较远的点。异常检测在诸多领域中有广泛应用，例如，在电子现金支付过程中，异常点代表着套现欺诈行为；在科学计算领域，异常数据和正常数据具有相等的利用价值[2]，如天文图像检测中的异常点可能意味着新星的出现；在网络安全领域，异常点可能是恶意用户的非法入侵。

近年来，基于密度评估的异常检测方案深受关

收稿日期：2019–01–08；修回日期：2019–05–03
基金项目：国家重点研发计划基金资助项目（No.2017YFB0802300）
Foundation Item: The National Key Research and Development Program of China (No.2017YFB0802300)

注[3]。基于密度评估的异常检测方案将异常点定义如下：异常点是低密度区域的数据对象，密度的核心概念是近邻距离。对密度概念的改进衍生出不同的算法，例如局部异常因子算法（LOF, local outlier factor）[4]、密度偏移抽样算法[5]等。LOF 通过 k 近邻距离计算局部可达密度，得到每个点的局部离群因子，根据阈值判断点是否异常。LOF 的优势是既可以计算局部异常点，也可以计算全局异常点，在小数据集中效果极佳。利用聚类进行异常检测也是基于密度概念，此类算法利用数据点的分布规律对数据集进行分簇，按每个数据点到簇中心的距离排序，根据超参阈值比较，超过阈值的数据点称为异常点。例如经典 k-means 聚类算法检测网络流量异常[6]，利用遗传算法对 k-means 聚类改进解决了局部最优问题[7]；Tang 等[8-9]提出了基于特征选择的模糊聚类异常检测模型，利用层次聚类和遗传算法改进了聚类模型，进一步降低了异常检测的误报率。

随着大数据时代的到来，数据的量和维度发生了爆炸式增长。高维数据存在 2 个问题：1) 距离计算上的“维数灾难”[10]，数据相似度的计算离不开距离计算，比如欧氏距离，随着数据维度的增加，点与点间距离的区分度变小，数据分布稀疏，异常点不再敏感；2) 时间复杂度过高，高维数据间的距离计算所需时间开销过大，对实时检测应用来说无法满足需求，例如网络入侵检测和信用卡欺诈检测都对低时间开销有较高要求。基于密度评估的异常检测方案时间复杂度均在 $O(n^2)$[11]，因此设计出对高维度、大数据集进行异常检测的高效方法具有重要意义。

质量评估思想在数据分类、回归异常检测等领域有显著效果，该思想重新定义数据点靠近数据簇中心或靠近数据簇边缘的度量，并称该度量为质量。相比于密度评估方法，基于质量评估的方法有以下 2 个优势[12]：1)数据质量的计算量小，数据质量计算只统计一个区域内的数据量，不需要计算距离；2)数据质量的大小反映了数据点是靠近还是远离数据簇中心。

基于质量评估的异常检测方案利用隔离机制来计算数据质量。根据隔离机制的不同，衍生出许多异常检测算法，例如 half-space tree[13]、SCiForest[14]、基于近邻距离的隔离机制[15]等。

孤立森林（iForest，isolation forest）[16]属于集成学习方法，是随机森林算法的无监督版本，广泛应用于异常检测领域。iForest 对数据空间进行随机隔离，以此构造决策树桩（decision stump），也称为孤立树（iTree, isolation tree）。iForest 也符合质量评估思想，质量被定义为 iTree 中叶节点的深度，深度越小，越有可能为异常点。

iForest 解决了高维数据集中异常检测的 2 个问题[17]：1) iForest 不需要计算距离，算法的时间开销不随数据维度的增加而增加，为线性时间复杂度；2) iForest 对大型数据集的检测性能好，并且是集成学习算法，iTree 越多，iForest 越稳定。

虽然 iForest 适用于高维数据集的异常检测，但随着数据分布复杂性的增加，检测效率也会降低，而且在极高维数据的异常检测中，算法的波动性较高。因此，本文提出基于多特征决策的随机超平面隔离机制，以及基于滑动窗口的多粒度扫描机制，进而构造层次化集成学习模型。

2 iForest 方案

2.1 质量评估思想

设数据集 $\boldsymbol{D}=\left\{\vec{\boldsymbol{X}}_1,\vec{\boldsymbol{X}}_2,\vec{\boldsymbol{X}}_3,\cdots,\vec{\boldsymbol{X}}_n\right\}$ 且 $\boldsymbol{D}\subset\boldsymbol{\mathcal{R}}^u$，$\boldsymbol{\mathcal{R}}^u$ 为实数集，u 为最高维数，$\vec{\boldsymbol{X}}_j=\left\{x^1,x^2,x^3,\cdots,x^u\right\}$，$j\in[1,n]$。

定义 1 隔离超平面。假设数据集 $\boldsymbol{D}$ 在维数 i 上有序，$s_j^i:\vec{\boldsymbol{W}}\vec{\boldsymbol{X}}+b$，当且仅当 $\vec{\boldsymbol{W}}\vec{\boldsymbol{X}}_j+b<0$ 且 $\vec{\boldsymbol{W}}\vec{\boldsymbol{X}}_{j+1}+b>0$ 时，s_j^i 为数据点 $\vec{\boldsymbol{X}}_j$ 和 $\vec{\boldsymbol{X}}_{j+1}$ 在维度 i 上的隔离超平面。

定义 2 基本质量函数。数据集 D 中每个点 $\vec{X}_j$ 都有

$$m_j(\vec{\boldsymbol{X}})=\begin{cases}m_j^{\mathrm{L}},x^i\text{位于}s_j^i\text{左边}\\ m_j^{\mathrm{R}},x^i\text{位于}s_j^i\text{右边}\end{cases}\tag{1}$$

其中，m_j^{L} 为数据集 D 在维度 i 上在 s_j^i 左边的数据点数目，m_j^{R} 为维度 i 上在 s_j^i 右边的数据点数目。

定义 3 数据点质量函数。计算式如式(2)所示。

$$\mathrm{mass}\left(\vec{\boldsymbol{X}}\right)=\sum_{i}^{u}\sum_{j}^{n-1}m_j\left(\vec{\boldsymbol{X}}\right)p\left(s_j^i\right)\tag{2}$$

其中，$\mathrm{mass}(\vec{\boldsymbol{X}})$ 是向量 $\vec{\boldsymbol{X}}$ 在各个维度上的基本质量函数加权和；$p(s_j^i)$ 是超平面 s_j^i 在数据空间中的概率，常常使用蒙特卡洛模拟方法[13]进行计算。

2.2　iForest 的构建

定义 4　孤立树。若 Node 是孤立树的节点，则是具有$(\text{Node}_\text{L},\text{Node}_\text{R})$子节点的内部节点，或是无子节点的终端节点。$\text{Node}_\text{L}$ 与 Node_R 的定义为在特征集合中选择 i，该特征上值区间内随机选择数据 j，小于 j 的数据划分为左子树 Node_L，大于 j 的数据划分为右子树 Node_R。

iForest 由 T 个 iTree 构成，如式(3)所示。

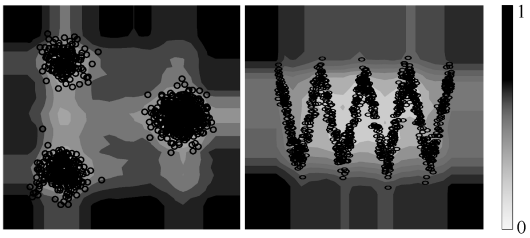
$$\text{IF}=\{t_1,t_2,\cdots,t_T\} \tag{3}$$

2.3　iForest 的问题

轴平行（axis-parallel）是指在单一特征的决策过程中，决策边界与坐标轴平行的现象。轴平行是决策树的一种特性，由于 iForest 的决策模式类似于决策树，因此也受轴平行特性的影响。

在密集的数据集中，受轴平行特性的影响，iForest 会产生重叠和覆盖效应，导致决策精度降低，同时会增加 iTree 的高度和训练过程的时间开销，无法高效生成 iTree，因此 iForest 更适用于具有分布稀疏特性的数据集。文献[16]提出类似于随机森林中子采样的方法解决了这个问题。设定 ψ 为随机子采样的样本数量，iTree 由随机子采样的样本集生成。

图 1(a)构造了 3 个服从高斯分布的数据集。左上数据簇的数量为 300，左下为 500，右侧为 1 000。图 1(b)中的数据量为 1 000，数据的分布模式符合余弦函数趋势。黑白梯度线为异常分数的等高线，黑色表示 1，白色表示 0，异常分数越大，表示越有可能为异常点。如图 1(a)所示，异常分数梯度线在数据簇的平行轴线上偏差较大；图 1(b)失去了余弦函数趋势，无法正确检测异常点。



(a) 3个数据簇　　(b) 余弦数据分布

图 1　iForest 对不同数据集的异常分数

3　基于多粒度级联孤立森林算法

为解决 iForest 的不足，本文提出基于多维度随机超平面的孤立森林（MRHiForest, multi-dimensional random hyperplane iForest）隔离机制，在数据集隔离的过程中，使用多元线性组合构成多样化的随机超平面。同时，利用多粒度扫描器（MGS，multi-grained scanner）进行高维数据的特征子采样，类似随机森林，但样本的选取采用滑动窗口的方式，特征样本存在连续性。每个特征样本构造新的数据集训练孤立森林，以此构造基于多粒度级联孤立森林算法的异常检测模型。

3.1　随机超平面隔离

定义 5　随机超平面。随机超平面为 iForest 的隔离机制产生的超平面，S^u 是所有随机超平面的集合。$p(x,y)$ 表示为点 x 及点 y 被随机超平面 K 隔离的概率，如式(4)所示。

$$K:\vec{\boldsymbol{W}}\vec{\boldsymbol{X}}+b=\sum_{i=1}^{d}w^ix^i+b \tag{4}$$

其中，$\vec{\boldsymbol{W}}=\{w^1,w^2,\cdots,w^d\}$ 为 K 的斜率，b 代表截距。

iForest 中的隔离机制为式(4)的特例。iForest 随机选择一个特征 η，令

$$w^i=\begin{cases}1, i=\eta\\ 0, i\neq\eta\end{cases},i\in[1,d] \tag{5}$$

联合式(4)和式(5)得到 iForest 的隔离超平面为

$$\text{iforest}(K):x^\eta+b=0 \tag{6}$$

iForest 仅就一个特征 η 进行隔离，丢失了大部分特征信息，因此随着数据维数的增加，iForest 的性能不稳定。相比之下，随机超平面的隔离机制包含所有特征信息。

在 iForest 中，数据的隔离是随机的，针对随机超平面的随机选择容易出现偏离数据集现象，造成无效开销。本文利用法向量随机生成斜率向量 $\vec{\boldsymbol{W}}$。首先随机选择 2 个点，然后求 2 个点的法向量作为斜率向量，从而保证随机超平面存在于数据集中。

图 2(a)是二维数据空间中一个 iTree 的生成过程实例。选取平行于轴的超平面来隔离数据，数据质量高的数据点被隔离多次才会被划分出去（如图 2(a)中的点 n），而质量低的数据点经过少数的几次隔离就会被划分出去（如图 2(a)中的点 a）。图 2(b)展示了 MRHiForest 中随机超平面的生成过程，图中两点 $\vec{\boldsymbol{X}}_a$ 和 $\vec{\boldsymbol{X}}_b$ 是随机选取的，两点的法向量表示随机超平面的方向，灰色区域为截距 b 的区间选取范围。从图 2 中可以明显看出，iForest 隔离超平面是平行于坐标轴的，MRHiForest 隔离超平面的方向是随机的。

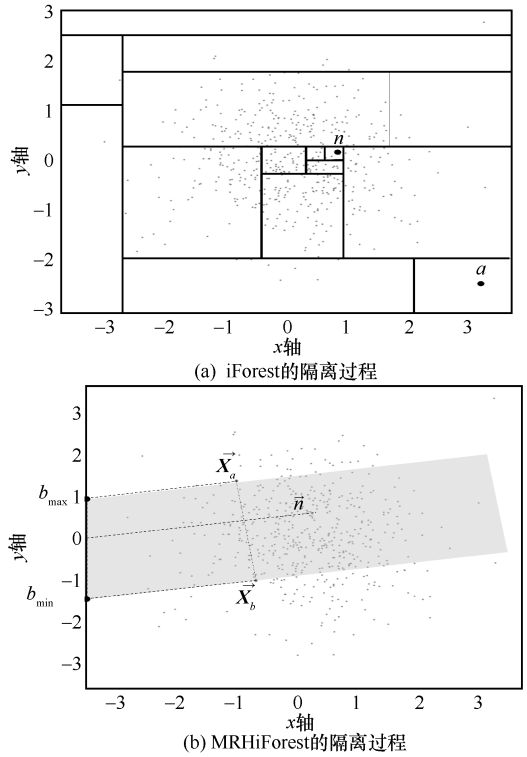


图 2　隔离超平面的构造过程

生成 MRHiTree 的伪代码如算法 1 所示。

算法 1　生成 MRHiTree

参数　数据集 $\boldsymbol{X}$，当前树高度 h，阈值 yz

1) if　$h<\text{yz}$ or $|\boldsymbol{X}|<1$

2)　　return　extendNote{Size=$|X|$}

3) else

4)　#随机选取 2 个数据点

5)　$\vec{\boldsymbol{X}}_a,\vec{\boldsymbol{X}}_b$ =RandomGenerateData(X)

6)　#计算两点间垂直法向量

7)　$\vec{\boldsymbol{W}}$ =GetNormal($\vec{\boldsymbol{X}}_a,\vec{\boldsymbol{X}}_b$)

8)　#随机生成截距

9)　b =RandomGIntercept($\vec{\boldsymbol{W}}\vec{\boldsymbol{X}}_a$, $\vec{\boldsymbol{W}}\vec{\boldsymbol{X}}_b$)

10)　$\boldsymbol{X}_{\text{L}}$ = filter($\boldsymbol{X}$, $\vec{\boldsymbol{W}}\boldsymbol{X}+b\leqslant 0$)

11)　$\boldsymbol{X}_{\text{R}}$ = filter($\boldsymbol{X}$, $\vec{\boldsymbol{W}}\boldsymbol{X}+b>0$)

12)　#生成内部节点，构造子树

13)　return Node{Left=MRHiTree($\boldsymbol{X}_{\text{L}}$, h+1, yz), Right=MRHiTree($\boldsymbol{X}_{\text{R}}$,h+1,yz), Scope= $\vec{\boldsymbol{W}}$, Intercept=b}

14) end if

3.2　多粒度扫描采样

定义 6　多粒度扫描。设数据的特征集合 $P=\{d_1,d_2,d_3,\cdots,d_u\}$，特征的最大值为 u。多粒度扫描定义窗口大小 q，当且仅当 $u>q$，根据窗口 q 重新构成新的特征集合，定义滑动窗口步长 step，生成多个子特征集合，新的特征集合构成新的数据集。如式(7)所示。

$$\text{NEW}_P=\{P_1,P_2,P_3,\cdots,P_L\}$$
$$P_i=\left\{d_{(i-1)\text{step}+1},d_{(i-1)\text{step}+2},\cdots,d_{(i-1)\text{step}+q}\right\},i\in[1,L] \quad (7)$$

其中，L 为特征子采样的最大值，如式(8)所示。

$$L=\frac{u-q}{\text{step}}+1 \quad (8)$$

多粒度扫描的滑动窗口过程如图 3 所示。影响特征空间大小的因素为 step 和 q，随着 step 的减小，特征空间数量越多，但是时间开销就越高。相反，随着 step 增大，生成新的特征空间数量就越少，当 $\text{step}>q$ 时，会产生特征丢失现象，因此 step 的理论峰值为 q。MGS 伪代码如算法 2 所示。

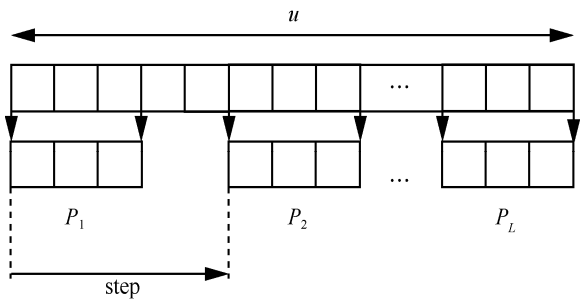


图 3　多粒度扫描过程

算法 2　MGS

参数　数据集 $\boldsymbol{X}$，维度集 Dims，维度数目 u，步长 step

1) if　$u>q$

2)　　return { $\boldsymbol{X}$ }

3) else

4)　new_P={ ϕ }

5)　$i,j=1$

6)　while　$i<u$

7)　#从 $\boldsymbol{X}$ 中根据 d_i 到 $d_{i+\text{step}}$ 生成新的数据集

8)　　$\boldsymbol{X}_j$ = GenerateData($\boldsymbol{X}$, d_i , step)

9)　　new_P += { $\boldsymbol{X}_j$ }

10)　　i+ = step

11)　　j+ = 1

12)　end while

13)　return new_P

14) end if

3.3　层次化集成学习异常检测模型

本文首先利用多粒度扫描机制 MGS 作为特征选择过程，然后利用多维度随机超平面隔离机制 MRH 对基于孤立森林 iForest 的异常检测模型进行优化，从而构建基于多粒度扫描与多维度随机超平面的孤立森林算法（MGS-MRHiForest）的层次化集成学习异常检测模型。模型结构如图 4 所示，伪代码如算法 3 所示。

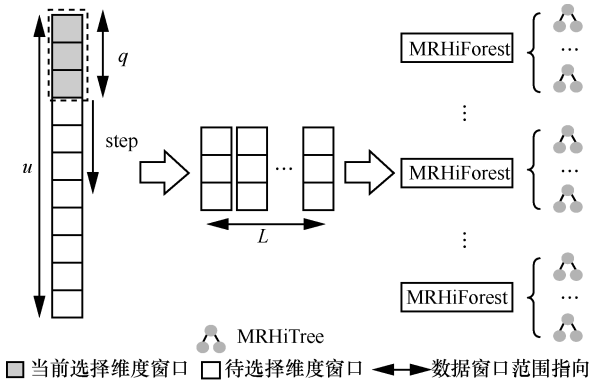


图 4　基于 MGS-MRHiForest 的层次化集成学习异常检测模型

算法 3　MGS_MRHiForest

参数　数据集 X，MRHiTree 的数量 T，子样本数 ψ

1) MGS_MRHiForest = { ϕ }

2) #初始化参数，阈值 ts 以及步长 step

3) #多粒度扫描得到新数据集, Dims 为数据集的维数特征，len(Dims)计算 Dims 的特征数目

4) MGS_X = MGS(X, Dims, len(Dims), step)

5) for each NEW_X in MGS_X

6)　MRHiForest = { ϕ }

7)　for i=1 to T do

8)　#随机选择 ψ 个样本形成新数据集

9)　$\boldsymbol{X}'$ = sample(NEW _ X , ψ)

10)　MRHiForest+={MRHiTree($\boldsymbol{X}'$,0,ts)}

11)　MGS_MRHiForest+={MRHiForest}

12)　end for

13) end for

14) return MGS_MRHiForest

MRHiForest 经过多粒度扫描后形成森林集合 $\mathrm{RFs}=\{\mathrm{RF}_1,\mathrm{RF}_2,\cdots,\mathrm{RF}_L\}$，$h(\vec{\boldsymbol{X}})$ 表示叶子节点 $\vec{\boldsymbol{X}}$ 的深度，经过集成学习计算的过程如式(9)所示。

$$h(\vec{\boldsymbol{X}})=\frac{1}{TL}\sum_{\mathrm{RF}\in\mathrm{RFs}}\sum_{t\in\mathrm{RF}}h_t(\vec{\boldsymbol{X}}) \tag{9}$$

对 $h(\vec{X})$ 归一化处理后得到异常分数 $S(\vec{X},\psi)$，如式(10)所示。

$$S(\vec{\boldsymbol{X}},\psi)=2^{-\frac{h(\vec{X})}{c(\psi)}} \tag{10}$$

$$\mathrm{c}(\psi)=2\ln(\psi-1)+\gamma-\frac{2(\psi-1)}{\psi} \tag{11}$$

其中，ψ 表示 MRHiForest 的随机子采样大小，欧拉常数 $\gamma=0.577\,215\,664\,901\,532\,8$，$\mathrm{c}(\psi)$ 表示孤立树中查找点失败的平均路径。

由式(11)可知，$h(\vec{\boldsymbol{X}})$ 越接近 0，$S(\vec{\boldsymbol{X}},\psi)$ 就越接近 1，代表 $\vec{\boldsymbol{X}}$ 越容易被隔离，其为异常点的可能性就越高。

iForest 的时间复杂度为 $O(T\psi\mathrm{lb}\psi)$ [18]，多粒度级联会产生 L 个森林，所以 MGS_MRHiForest 的时间复杂度为 $O(LT\psi\mathrm{lb}\psi)$。

4　实验结果与分析

实验环境为 Intel Core i7-6700 3.4 GHz；16 GB 内存：Windows 10 操作系统。本文所有算法都基于 Python 语言的 Sklearn 库实现，MRHiForest 在原始 iForest 基础上增加了多粒度扫描算法和随机森林算法。本文使用 Area Under ROC Curve(AUC)作为算法性能评测标准，AUC 越大，代表学习模型的泛化能力越强。所有实验均经过 5 次运算得到测量结果，并以其算术平均值作为最终的实验结果。

iForest 的默认参数设定为 iTree 的数量 $T=100$，子样本数量 $\psi=256$。这是因为 iForest 在此参数下有最好的检测效果。

MGS 的默认参数设定为维数阈值 $q=100$，粒度扫描步长 step = 1。q 体现多粒度扫描的特征选择过程，step 则关系着样本集的多样性，步长越小，样本集的多样性就越高，iForest 的泛化能力就越强，但代价是时间开销会增加。

4.1　复杂数据模式的局部异常点检测

为了验证在复杂数据分布的数据集中进行异常检测的效果，使用阿基米德螺旋方程构造了包含 1 000 个点的螺旋数据分布数据集，分别使用 iForest 和 MRHiForest 算法生成异常分数图，以展示算法对异常点的梯度分布。

实验结果如图 5 所示。其中亮区表示异常分数较低，暗区表示异常分数较高，两部分区域构成了

复杂数据模式下正常数据和异常数据的数据分布规律。由图 5(a)可知，iForest 生成的异常分数梯度偏差较大，图 5(b)中 MRHiForest 的异常分数梯度更符合螺旋数据的分布规律。上述实验结果与算法的隔离机制有关，iForest 轴平行的特点导致偏差，MRHiForest 对隔离机制的改进使之对复杂数据模式具有更好的隔离能力。

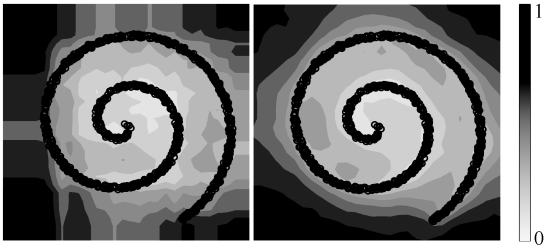


图 5　iForest 和 MRHiForest 在螺旋数据集中的异常分数图

为了测试算法对异常数据的稳健性，在上述螺旋数据集中逐步添加异常点，分别计算 2 种算法的 AUC。螺旋数据集中添加 100 个异常点如图 6 所示。

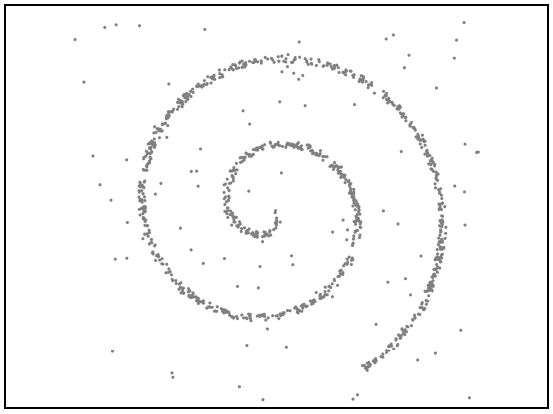


图 6　螺旋数据集中插入 100 个均匀分布异常点

实验结果如图 7 所示，2 种算法的 AUC 曲线表明，MRHiForest 整体性能高于 iForest，说明 MRHiForest 的随机超平面隔离机制更好地隔离了复杂数据模型的局部异常点。当异常点数为 115 时，iForest 的 AUC 减少到 0.9 以下；当异常点为 200 时，MRHiForest 的 AUC 下降到 0.9 以下，此时 iForest 的 AUC 为 0.78。由此可知，MRHiForest 的稳健性强于 iForest，AUC 的持续下降是因为异常点的数量占比达到数据集的 16%以上，iForest 的适用前提正是数据集中异常点分布的稀疏特性。

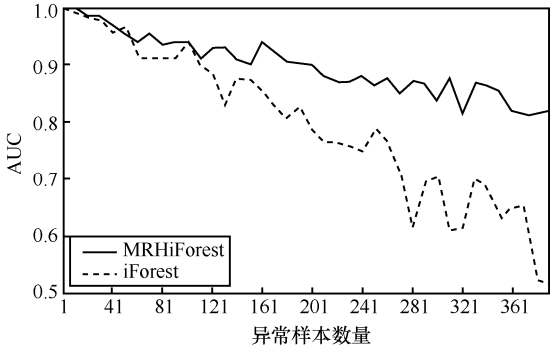


图 7　iForest 和 MRHiForest 在不同异常点样本数量上的 AUC

上述实验中，MRHiForest 将 iTree 数量 T 直接设定为 iForest 的最佳参数值 100。为进一步探讨 MRHiForest 中 T 的最佳设定，在异常样本集为 100 的螺旋数据集中，令 T 分别为 20、50、100、150、200、250、300、350，逐一计算 MRHiForest 的 AUC，以 5 次实验结果的算术平均值作为最终结果，并以方差作为算法的稳定性指标。实验结果分别如图 8 和图 9 所示。

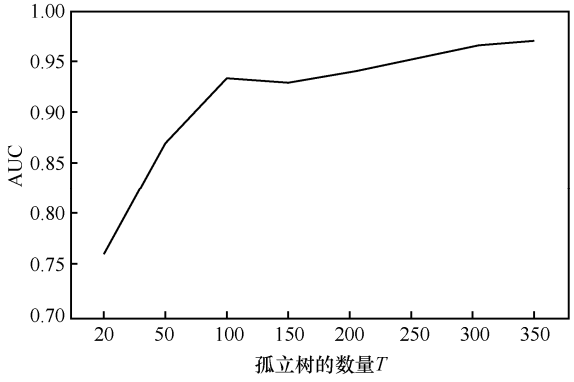


图 8　MRHiForest 在不同数量孤立树下的 AUC

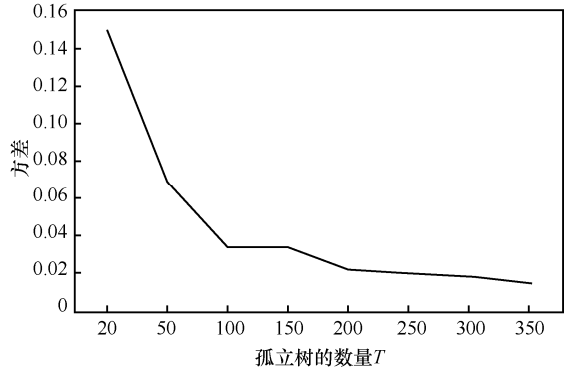


图 9　MRHiForest 在不同数量孤立树下 AUC 的方差

随机性是导致 iForest 性能不稳定的主要原因[19]。而图 8 和图 9 表明，随着 iTree 数量 T 的增加，MRHiForest 的性能和稳定性也在增强，

当 T=100 时，AUC 的增长趋势和方差逐渐平缓。这是因为 MRHiForest 属于集成学习方法，弱学习器 iTree 的增加会降低错误率，增加稳定性[20]。考虑到 iTree 带来的时间开销，将 T 的默认值设定为 100。

4.2　高维数据集异常点检测

为了检测多粒度扫描机制 MGS 对高维数据集异常点检测的性能，使用维数为 617 的实验数据集 isolet，分别测试 iForest 和 MRHiForest 的平均异常分数来对比算法对异常点的分离程度。平均异常分数 AveScore 定义为数据集中所有异常点的异常分数的算术平均值，如式(12)所示。

$$\text{AveScore}=\frac{1}{n_a}\sum_{x_a\in\text{isolet}}S(x_a,\psi) \tag{12}$$

其中，n_a 为异常数据点总数，$S(x_a,\psi)$ 为异常点 x_a 在 iForest 中的异常分数。

实验分为 iForest 组和 MRHiForest 组，分别使用不带 MGS 的 iForest 和 MRHiForest 与带 MGS 的 MGS-iForest 和 MGS-MRHiForest 进行评测并计算其 AveScore。由于 MGS 会产生多个子数据集，每个子数据集再分别使用 iForest 或者 MRHiForest 进行层次化集成学习，因此，对未启用 MGS 的 iForest 或者 MRHiForest 也构造层次化集成学习模型，仅对完整的数据集进行 L 次重复训练，从而生成 L 个 iForest 或者 MRHiForest。对各组分别进行 500 次实验，结果分别如图 10 和图 11 所示。

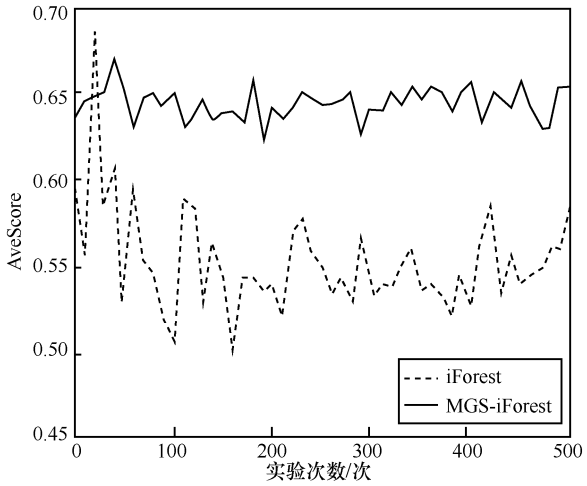


图 10　iForest 组在高维数据集中的平均异常分数

图 10 的测试结果显示，iForest 的 AveScore 在 0.55 附近波动，MGS-iForest 的 AveScore 在 0.65 附近波动，说明 MGS-iForest 对异常点的检测和分离能力高于 iForest。

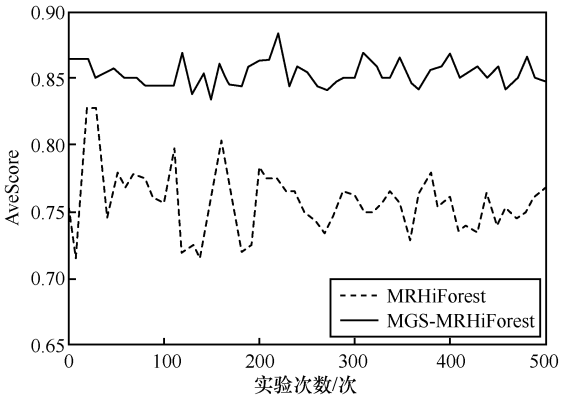


图 11　MRHiForest 组在高维数据集中的平均异常分数

图 11 的测试结果显示，MRHiForest 的 AveScore 在 0.76 附近波动，MGS-MRHiForest 的 AveScore 在 0.85 附近波动，说明 MGS-MRHiForest 对异常点的检测和分离能力高于 MRHiForest。

结合两组实验结果，使用 MGS 的算法 AveScore 普遍高于不使用 MGS 的算法，说明多粒度扫描机制提高了算法对高维数据集异常点的检测性能。

进一步的实验结果如表 1 所示，在不启用 MGS 的 iForest 和 MRHiForest 的 500 次实验中，平均异常分数最大值与最小值之间的极差分别达到 0.19 和 0.11；而在启用 MGS 的对照实验中，平均异常分数最大值与最小值之间的极差降为 0.05。这说明多粒度扫描机制提高了异常检测算法的稳定性。

表 1　　4 种算法的平均异常分数

算法	最大值	最小值	极差
iForest	0.69	0.50	0.19
MGS-iForest	0.67	0.62	0.05
MRHiForest	0.83	0.72	0.11
MGS-MRHiForest	0.88	0.83	0.05

4.3　真实数据集

本节对表 2 所示的 4 个真实数据集分别进行异常检测算法的性能评估。

表 2　　真实数据集详细情况

数据集	数据集大小/个	异常点比例	维度
isolet	7 797	1.4%	617
P53Mutant	31 159	0.5%	5 408
http	56 7497	0.4%	3
mnist	20 444	3.3%	96

表 2 中，isolet 是简单的音频字母识别数据集，包括 617 个特征。P53Mutant 是 P53 基因编码中与癌症相关的数据集，包括 5 408 个特征。http 是网络入侵检测数据集，来自 KDD CUP99，提取其中 3 个特征。mnist 为手写数字 0~9 数据集，提取其中 2、3、5 的数据，并利用文献[15]的算法进行处理，特征维度是 96。对于低维度数据集 http 和 mnist，不需要进行多粒度扫描；对于高维度数据集 isolet 和 P53Mutant，则要开启多粒度扫描机制。

为评估基于质量评估算法和基于密度评估算法的优劣，将 LOF 算法作为实验的对比算法之一。在参数选择上，LOF 的参数区间设置为 10~1 000。iForest 的随机子采样数量设置为 2^n，n 取值 1~10，取 AUC 最优组为实验结果。实验结果如表 3~表 5 所示。

表 3　　3 种算法的 AUC

数据集	AUC		
	MRHiForest	iForest	LOF
isolet	1.00	0.91	0.94
P53Mutant	0.65	0.60	0.62
http	1.00	1.00	1.00
mnist	0.89	0.85	0.87

表 4　　3 种算法的时间开销

数据集	执行时间/s		
	MRHiForest	iForest	LOF
isolet	5	0.8	2
P53Mutant	54	19	43 235
http	8	66	19 965
mnist	4	2	678

表 5　　3 种算法的最优参数设定

数据集	最优参数		
	MRHiForest ψ	iForest ψ	LOF K
isolet	32	512	40
P53Mutant	256	512	2 000
http	128	256	500
mnist	128	256	300

由表 3 可知，4 个数据集中，MRHiForest 的 AUC 均优于 iForest 和 LOF。其中 iForest 仅在低维度的 http 数据集中表现出和 MRHiForest 和 LOF 相同的检测性能。在 http 数据集中，MRHiForest 的时间开销比 iForest 少，这是因为：1) MRHiForest 的随机子采样数目小于 iForest，这使 iTree 的构建速度更快；2) MRHiForest 的隔离机制提高了算法对异常点的敏感性，进而提高了异常检测效率。

由表 4 可知，除了 isolet 数据集外，其他数据集中 LOF 的时间开销最大，这是因为：1) 随着数据量的增加，LOF 的最优参数 K 随之增大，增加了处理开销；2) 随着数据集维度的增加，距离计算的时间复杂度随之提高。isolet 数据集中 LOF 的时间开销小是因为其数据量小，因此 LOF 不适用于高维度、大数据量下的异常检测。在高维数据集中 iForest 的时间开销是最少的，这是因为 iForest 的单维度隔离机制对数据集的维数没有依赖性，无论维度多高的数据集，iForest 都能以线性的时间开销进行异常检测[21]。

由表 3~表 5 可知，数据集维度和数据集大小与算法的时间开销存在相关性，因此设计了 2 个实验进一步探索其相关性。

第一个实验用来测试数据集大小与算法时间开销的关系。将 http 作为实验数据集，选择 1×10^3~500×10^3 个数据分别测试算法的执行时间。

图 12 表明，随着数据集大小的增加，iForest 和 MRHiForest 的执行时间均单调增加，但 MRHiForest 的时间开销更小。这是因为 MRHiForest 的平均查找路径更短，而 iForest 由于轴平行特性，iTree 相对较高，平均查找路径较长。

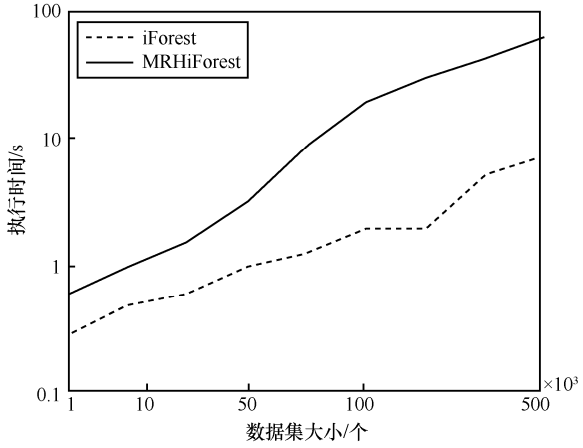


图 12　iForest 和 MRHiForest 不同数据集大小的执行时间

第二个实验用来测试数据集维度与算法时间开销的关系。将 P53Mutant 作为实验数据集，选择 5 到 1 000 个维度分别测试算法的执行时间。

图 13 表明，随着数据集维度的增加，iForest 和 MRHiForest 的执行时间均单调递增，但维度的增加对 MRHiForest 的影响更大。这是因为 MRHiForest

中每一颗树的节点均是多项式计算，维度的提升增加了计算开销；iForest 仅随机选择一个维度进行隔离，并不依赖于维度大小，因此时间开销增加不明显。

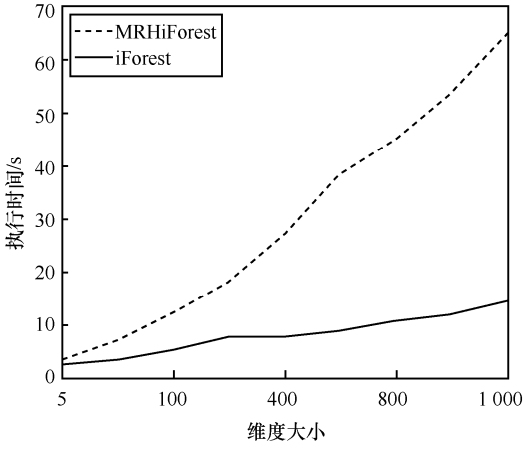


图 13　iForest 和 MRHiForest 不同数据集维度的执行时间

进一步测试多粒度扫描机制对 MRHiForest 和 iForest 的性能提升，实验结果如表 6 所示。

表 6　　多粒度扫描机制下算法的 AUC

数据集	AUC	
	MGS-MRHiForest	MGS-iForest
isolet	1.00	0.98
P53Mutant	0.75	0.69

由表 6 可知，通过多粒度扫描的特征选择，MRHiForest 和 iForest 的 AUC 均有所提升，验证了多粒度扫描机制对算法的优化效果。

5　结束语

本文提出基于多维度随机超平面的 iForest 异常检测模型 MRHiForest，同时引入多粒度扫描机制 MGS，构造了层次化集成学习异常检测模型。MRHiForest 使用随机超平面生成 MRHiTree，使隔离机制更符合数据分布特征；MGS 增加了对高维数据集检测的多样性。与传统 iForest 的对比实验结果表明，MRHiForest 对复杂的数据模型有更高的检测效率，并且在低维数据集中检测时间更少。对高维度数据集进行的对比实验结果表明，MGS-MRHiForest 能够弥补 iForest 对高维度数据异常点不敏感和检测不稳定的缺陷。

MGS 未考虑到关联属性特性，增加了算法的不确定性，随着集成数量的增加，时间开销有所增加，因此后续工作将考虑对关联属性特性的处理，进一步改善异常检测模型的性能。

参考文献：

[1] GRUBBS F. Procedures for detecting outlying observations in samples[J]. Technometrics, 1969, 11(1):1-21.

[2] 毛嘉莉, 金澈清, 章志刚, 等. 轨迹大数据异常检测: 研究进展及系统框架[J]. 软件学报, 2017, 28(1): 17-34.
MAO J L, JIN C Q, ZHANG Z G, et al. Anomaly detection for trajectory big data: Advancements and framework[J]. Journal of Software, 2017, 28(1):17-34.

[3] ZHANG L, LIN J, KARIM R. Adaptive kernel density-based anomaly detection for nonlinear systems[J]. Knowledge-Based Systems, 2018, 139: 50-63.

[4] BREUNIG M M, KRIEGEL H P, NG R T. LOF: identifying density-based local outliers[C]// ACM SIGMOD International Conference on Management of Data. ACM, 2000:93-104.

[5] 付培国, 胡晓惠. 基于密度偏倚抽样的局部距离异常检测方法[J]. 软件学报, 2017, 28(10): 2625-2639.
FU P G, HU X H. Anomaly detection algorithm based on the local distance of density-based sampling data[J]. Journal of Software, 2017,28(10): 2625-2639.

[6] AHMED M, MAHMOOD A N, HU J. A survey of network anomaly detection techniques[J]. Journal of Network and Computer Applications, 2016, 60: 19-31.

[7] 李洪成, 吴晓平, 严博. 面向 MANET 异常检测的分布式遗传 *k*-means 研究[J]. 通信学报, 2015, 36(11): 167-173.
LI H C, WU X P, YAN B. Research on distributed genetic *k*-means for anomaly detection in MANET[J]. Journal on Communications, 2015, 36(11): 167-173.

[8] 唐成华, 刘鹏程, 汤申生, 等. 基于特征选择的模糊聚类异常入侵行为检测[J]. 计算机研究与发展, 2015, 52(3): 718-728.
TANG C H, LIU P C, TANG S S, et al. Anomaly intrusion behavior detection based on fuzzy clustering and features selection[J]. Journal of Computer Research and Development, 2015, 52(3):718-728.

[9] 程国振, 程东年, 俞定玖. 基于多尺度低秩模型的网络异常流量检测方法[J]. 通信学报, 2012, 33(1): 182-190.
CHENG G Z, CHENG D N, YU D J. Network traffic detection based on multi-resolution low rank model[J]. Journal on Communications, 2012, 33(1): 182-190.

[10] 张晶,冯林. 针对动态非平衡数据集鲁棒的在线极端学习机[J]. 计算机研究与发展, 2015, 52(7):1487-1498.
ZHANG J, FENG L. An algorithm of robust online extreme learning machine for dynamic imbalanced datasets[J]. Journal of Computer Research and Development, 2015, 52(7): 1487-1498.

[11] GOLDSTEIN M, UCHIDA S. A comparative evaluation of unsupervised

anomaly detection algorithms for multivariate data[J]. PloS one, 2016, 11(4): e0152173.

[12] KAI M T, ZHOU G T, LIU F T, et al. Mass estimation and its applications[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2010: 989-998.

[13] TING K M , ZHOU G T , LIU F T, et al. Mass estimation[J]. Machine Learning, 2013, 90(1):127-160.

[14] LIU F T, KAI M T, ZHOU Z H. On detecting clustered anomalies using SCiForest[C]// European Conference on Machine Learning and Knowledge Discovery in Databases. Springer-Verlag, 2010:274-290.

[15] BANDARAGODA T R, KAI M T, ALBRECHT D, et al. Isolation‐based anomaly detection using nearest‐neighbor ensembles[J]. Computational Intelligence, 2018, 34(4):968-998.

[16] LIU F T, TING K M, ZHOU Z H. Isolation forest[C]//The IEEE International Conference on Data Mining. IEEE, 2008: 413-422.

[17] MUJA M, LOWE D G. Scalable nearest neighbor algorithms for high dimensional data[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014 (11): 2227-2240.

[18] DOMINGUES R, FILIPPONE M, MICHIARDI P, et al. A comparative evaluation of outlier detection algorithms: Experiments and analyses[J]. Pattern Recognition, 2018(74): 406-421.

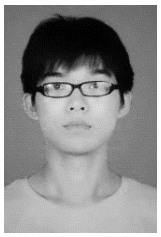
[19] PHAM B T, PRAKASH I, BUI D T. Spatial prediction of landslides using a hybrid machine learning approach based on random subspace and classification and regression trees[J]. Geomorphology, 2018(303): 256-270.

[20] KRAWCZYK B, MINKU L L, GAMA J, et al. Ensemble learning for data stream analysis: a survey[J]. Information Fusion, 2017(37): 132-156.

[21] ROY G, ROY G, ROY G, et al. Robust random cut forest based anomaly detection on streams[C]// International Conference on International Conference on Machine Learning. JMLR.org, 2016:2712-2721.

[作者简介]

杨晓晖（1975− ），男，河北巨鹿人，博士，河北大学教授、硕士生导师，主要研究方向为分布计算、信息安全与可信计算。



张圣昌（1993− ），男，河北邯郸人，河北大学硕士生，主要研究方向为分布式计算与信息安全。