基于 Canopy-Kmeans++ 算法的网络威胁流量检测

叶帅辰 [1]
YE Shuaichen

摘 要 针对传统的无监督 Kmeans 算法聚类时间长、准确率低等问题，提出一种改进型的 Canopy-Kmeans++ 算法对网络威胁流量进行检测。首先，通过前置粗聚类环节确定实际聚类簇数，避免了因簇数选择不准确导致的类标签与实际类特征存在差异的问题；其次，进一步使用 Kmeans++ 算法，引入基于距离加权的概率分布质心选择方案；最后，通过对 KDD Cup99 数据集中抽样选取的测试子集进行算法效果比对，可以看出所提出方法相比于传统 Kmeans、Kmeans++ 方法在聚类效果和聚类速度上均有提升。相关研究结果可为异常网络流量的聚类算法性能优化提供参考。

关键词 网络安全；流量检测；聚类算法；聚类效果

doi：10.3969/j.issn.1672-9528.2024.07.033

0 引言

聚类是一种常用的网络威胁检测分析方法，其主要思想是将所采集的网络流量数据按照某些特征划入互不相交的子集（又称为簇），并为每一子集定义不同标签，进而依据标签判断子集中数据是否为带有威胁的异常流量 [1]。聚类算法可根据数据特征及实现方式分为划分聚类、层次聚类、密度聚类、网格聚类等，其中划分聚类由于实现简单，且具有较强的迭代优化能力，常用于大规模网络流量的快速威胁检测 [2]。

划分聚类最初是以 Kmeans 算法为基础，后根据目标数据集的数据类型、分布情况、噪声及异常值等因素，衍生出了如 Kmedoids[3]、Kmodes[4] 和模糊 Kmeans[5] 等算法来适应不同场景的应用需求。不过，大部分衍生算法普遍是以改进传统 Kmeans 算法对于噪声的鲁棒性为目的 [6]，对 Kmeans 算法聚类中心选择随机性和簇数选择不确定性等问题尚未形成统一的解决方法。

针对上述两问题中的簇数选择不确定性，已有相关研究通过前置 Canopy 粗聚类环节来框定簇个数，以提升正常和异常网络流量的聚类检测准确性 [7-8]，并通过引入局部密度思想来处理原始数据集中的离群点 [9]。不过，由于未解决聚类中心选择随机性问题，相关前置粗聚类方法虽然提升了威胁流量检测的准确性，但从执行效率上相比于传统 Kmeans 方法尚无明显优势。本文在此基础上统筹考虑传统 Kmeans 算法的两核心缺陷，通过构建 Canopy-Kmeans++ 算法在确保聚类效率的前提下提升异常网络流量检测的准确率，并将三种不同聚类算法应用于 KDD Cup99 测试子集中进行比对，相关计算结果验证了所提出算法相比于传统 Kmeans 及 Kmeans++ 方法的准确性改善，以及相比于传统 Kmeans 的聚类效率提升。

1 Canopy-Kmeans++ 算法构建

1.1 传统 Kmeans 算法的局限性

Kmeans 算法的目的是将一个复杂数据集按照数据相似性划分为若干簇，并确保每一簇中数据相似程度最高。具体过程为：（1）在数据集中随机选定 K 个质心；（2）遍历数据集，依据与质心距离的最小化原则将数据归入各簇；（3）结合各簇增删数据情况重新确定质心；（4）如此往复，直至质心不再发生变化。

从该过程可以看出，Kmeans 算法的核心聚类效果取决于两个因素，一是初始 K 值的确定，另一个是各簇质心的选择。首先，在 K 值确定方面，对于具有多维海量数据的网络流量监测数据集，无法通过绘图直观确定聚类簇数，只能通过人工经验主观选择，但该方式无法保证所选值与实际最优值一致，进而导致聚类结果的显著误差。如图 1 所示，某数据集的真实聚类情况是划分为 5 个簇（即 K=5），其中 4 个簇是正常流量数据，1 个簇为异常流量数据，但如果依据人工经验设定该数据集 K=4，那么最终结果可能是将一个本来为正常的簇完全划进了异常数据簇当中，导致聚类结果与实际不符，影响聚类准确性。其次，在簇质心选择方面，同样在未能直观感知数据集全貌的前提下主观确定质心可能导致其与真实质心偏差较大，只能通过多轮迭代使指定质心逐步收敛

1. 中国信息通信研究院安全研究所 北京 100191

于真实质心，这就会导致整个算法执行时间过长，降低聚类效率。

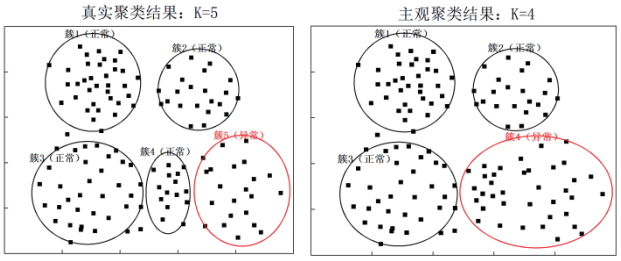

图 1 针对同一数据集不同 K 值选择的聚类差异

1.2 Canopy 算法粗聚类

Canopy 算法同样是一种适用于高维数据的无监督聚类算法，不过其无法对数据集中的每个样本进行精确划分，而是会将数据点划分至相互重叠的簇中，适用于作为 Kmeans 等精确聚类算法的前置步骤，以便快速确定簇个数（即 K 值）[10]。该算法的实现步骤如图 2 所示。

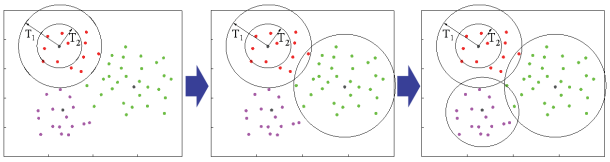

图 2 Canopy 粗聚类原理图

步骤一：根据数据集中样本分布情况，选定两个距离阈值 T_1（松阈值）、T_2（紧阈值），要求 $T_1>T_2$。

步骤二：在数据集中随机选择一个数据点 O 作为第一个簇的质心。

步骤三：依次计算数据集中其余点与该质心的距离，若距离小于 T_1，则将该点划分在该簇中；若距离小于 T_2，则将该点从原数据集中删除。

步骤四：对数据集中剩余数据点重复上述步骤，直到数据集为空。

可以看出，在 Canopy 执行过程中最具有决定性的指标就是两个距离阈值。如果阈值设置过大，则会将所有数据点划分至一个簇中，达不到粗聚类的目的；但如果设置过小，会使计算迭代次数大幅增加，影响整体聚类效率，所以阈值要根据数据集中各点间的距离综合确定。本文选择较为简单快速的均值确定法，首先将原始数据集各点间距离的均值计算如下：

$$avg(d)=\frac{2}{n(n-1)}\sum_{i=1}^{n}\sum_{j=i+1}^{n}d_{ij} \quad (1)$$

式中：d_{ij} 表示数据点 i 与 j 之间的欧氏距离。进一步地，将其与各点间距离中位数进行比较，两者取较大值作为松阈值 T_1，较小者作为紧阈值 T_2，如果两者相同，则根据距离排序删除离群点后进行重新比对。该方法的优势除实现简便外，还具备可根据聚类执行过程中原数据集的变化动态调整阈值的能力，防止当数据集剩余点较少时使用初始阈值可能导致的局部最优问题。

1.3 Kmeans++ 算法实现

在完成粗聚类及 K 值确定后，需对传统 Kmeans 算法的随机质心选择过程进行改进，以确保各簇质心相似程度较低（距离较远），从而减少迭代次数。本文采用 Kmeans++ 方法，通过引入距离权重来调整随机过程概率，具体步骤如下。

步骤一：在数据集中随机选择一个点作为第一簇的质心。

步骤二：分别计算数据集中剩余点与该质心的欧式距离。

步骤三：计算经距离加权后的随机概率分布，确保选择各点的概率与上一步所计算距离成正比，进而确定第二簇质心。

步骤四：依此往复，直到确定 K 个质心为止。

该方法的直观展示如图 3 所示，图 3 中表示某一二维数据集的 7 个数据点，假设已确定该数据集有两个簇，并随机选定了第一个簇的质心为 (1, 2)，那么该数据集的剩余 6 个数据点与质心 1 的距离及其被选为第二个簇质心的概率如表 1 所示。从表 1 中可以看出，第二簇质心被选在数据 4、5、6 中的总概率为 0.837，远大于数据 1、2、3 被选的概率，这便达到了质心间期望相似度低的目标。

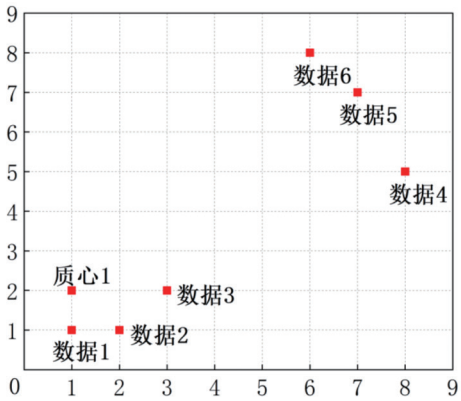

图 3 Kmeans++ 实现原理图

表 1 某数据集第二簇质心选择概率分布

	数据 1	数据 2	数据 3	数据 4	数据 5	数据 6
与质心 1 距离	1	1.41	2	7.6	7.81	7.81
被选为质心 2 的概率	0.036	0.051	0.072	0.275	0.283	0.283

2 算例分析

本文选择对同一测试数据集分别使用传统 Kmeans 算法、Kmeans++ 算法和 Canopy-Kmeans++ 算法进行聚类，以验证所提出算法的聚类准确性。计算的硬件环境为 Intel(R)

Core(TM) i5-10210U CPU @ 1.60 GHz，64 位 Windows 10 操作系统，16 GB 内存；采用编程语言为 Python 3.8.8，运行环境为 PyCharm 2018。具体实现流程如图 4 所示，在执行三种算法之前均需对测试数据集中的枚举变量进行数值化及标准化，从而消除各维数据对聚类结果的权重差异。

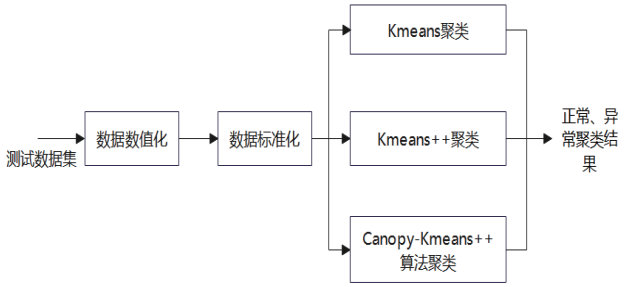

图 4 聚类算法比对原理图

2.1 测试数据集的选取

本文算例分析采用的实验数据为从 KDD Cup 99 数据集中提取的测试子集。该数据集由美国国防高级规划署于 1999 年发布，是从一模拟美国空军局域网中采集的为期 9 个星期的网络连接数据，共计 500 余万条。每条数据都包含 41 个流量特征位和 1 个正常异常标志位，41 个流量特征可归纳为：连接基本特征、连接内容特征、基于时间的流统计特征、基于连接数的流统计特征 4 项；标志位中异常情况包括拒绝服务攻击（DOS）、远程主机非法访问（R2L）、用户越权访问（U2R）、扫描探测（Probing）4 大类 39 子类。

因原始数据集过于庞大，为提升算法执行效率，从中抽取部分数据组成测试数据集如表 2 所示。为避免聚类结果的偶然性，测试集共包含 2 个子集，子集 1 总连接数为 13 000 条，子集 2 总连接数 1000 条，各子集内正常连接和异常连接的比例均大于 5:1。

表 2 测试数据集

	测试子集 1		测试子集 2	
	正常连接数	异常连接数	正常连接数	异常连接数
DOS 攻击	11 782	1218	865	135
R2L 攻击	10 367	2633	905	95
U2R 攻击	12 014	86	897	103
Probing	11 215	1785	920	80

2.2 评估参数

本文在对三种聚类算法进行性能比对时，选择了 3 个评估参数，分别是：聚类准确率（AC）、聚类召回率（Recall）和聚类时间（Time）。

聚类准确率表示聚类完成后每一簇中样本真实标签与簇标签一致的比率：

$$AC=\frac{1}{K}\sum_{i=1}^{K}\frac{N_i}{T_i} \quad (2)$$

式中：N_i 表示第 i 个簇中标签与簇标签一致的样本数，T_i 表示第 i 个簇的总数据量。聚类准确率可以用于单独评价每一个簇的聚类效果。

聚类召回率是指在所有具有某一特征样本中，被成功聚类为该特征的样本数占比：

$$\mathrm{Recall}=\frac{TP}{TP+FN} \quad (3)$$

式中：TP（true positive）表示实际具有某一特征，且被正确分类的样本数；FN（false negative）表示实际具有某一特征，但未被正确分类的样本数。聚类召回率可反映关注的某一数据特征整体聚类情况。

聚类时间是指对经数值化及标准化的测试数据完成全部聚类所需的时长。

2.3 聚类效果分析

使用 Kmeans 算法、Kmeans++ 算法和 Canopy-Kmeans++ 算法针对两组测试数据集进行聚类后，其聚类准确率分别如图 5、图 6 所示。可以看出对于不同测试集，使用 Kmeans 算法、Kmeans++ 算法的准确率均远小于本文所提出的 Canopy-Kmeans++ 算法，特别是对于测试集 1 中 DOS 攻击的识别，Canopy-Kmeans++ 算法的准确率最高可达到 92% 以上，相比于另外两种方法准确率分别提升了 17.2% 和 15.6%。另外，与测试集 1 的识别结果相比，拥有小样本量的测试集 2 针对不同类型攻击的识别准确率均有降低，这说明无论采用哪种聚类方法，增加样本量都可以在一定程度上减少离群点对于结果的干扰，优化聚类效果。

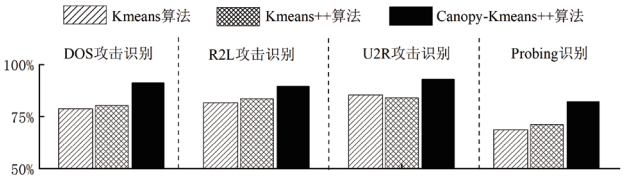

图 5 不同算法对于测试子集 1 各攻击识别准确性

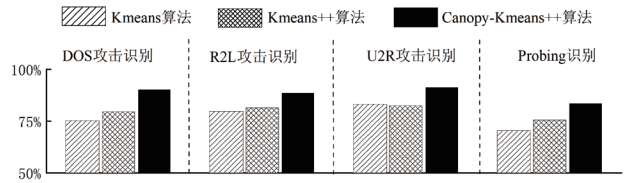

图 6 不同算法对于测试子集 2 各攻击识别准确性

同样使用三种算法完成聚类后，其针对正常流量样本的召回率分别如图 7、图 8 所示。从图中可以看出，三者相比依然是 Canopy-Kmeans++ 算法具有最高的召回率，说明其不仅对于单一特征标签的聚类效果较好，对正常异常样本的整

体鉴别能力也更强。另外，纵向对比图 5 ～图 8 可以看出，无论选取哪个测试集、采用哪类算法，四类网络风险中对于 Probing 的识别能力均较差，这是由于 Probing 并非常规意义上的网络攻击，大多数流量记录与正常流量差异并不显著，导致聚类效果一般。

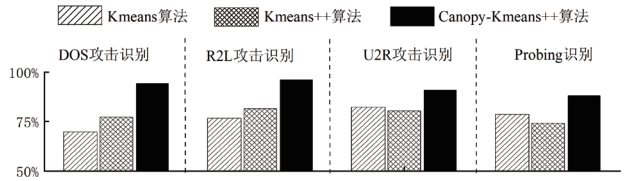

图 7 不同算法对于测试子集 1 正常网络流量的召回率

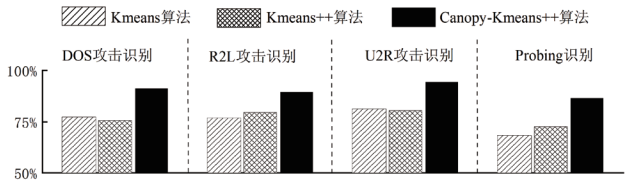

图 8 不同算法对于测试子集 2 正常网络流量的召回率

另外，三种算法针对不同数据集的平均聚类时间如表 3 所示。可以看出，无论对于大样本量还是小样本量，通过引入距离权重来调整质心选择概率的 Kmeans++ 算法和 Canopy-Kmeans++ 算法均相比于传统 Kmeans 算法具有更快的聚类速度，其中差距最显著的算例为针对测试集 1 进行聚类时，两算法的执行时间相比于传统方法分别缩减了 78.6% 和 73.5%。不过，由于所提出 Canopy-Kmeans++ 算法存在前置粗聚类过程，其针对两测试集的平均聚类时间均略大于 Kmeans++ 算法，这是在提升聚类准确性的过程中无法避免的。

表 3 不同算法聚类时长

执行时间 /s	Kmeans 算法	Kmeans++ 算法	Canopy-Kmeans++ 算法
测试子集 1	4.6	0.984	1.22
测试子集 2	0.42	0.11	0.13

3 结论

本文针对传统 Kmeans 算法中存在的执行时间长、聚类准确性低等问题，提出了改进的 Canopy-Kmeans++ 聚类算法。该方法通过前置 Canopy 粗聚类步骤和引入基于距离权重的质心选择方法，有效提升了聚类准确率及速度。通过对 KDD 99 中选取的测试数据集分别使用三种算法进行聚类后，可以看出 Canopy-Kmeans++ 算法相比于传统 Kmeans 算法和 Kmeans++ 算法聚类准确率均有提升，最高可提升 17.2%；同时，聚类时间相比于传统 Kmeans 算法最多可缩减 73.5%。因此，所提出方法可为常规网络威胁的识别及防御能力优化提供参考。

参考文献：

[1] 章永来，周耀鉴．聚类算法综述 [J]. 计算机应用，2019, 39(7): 1869-1882.

[2] MARVI M, ARFEEN A, UDDIN R. An augmented K-means clustering approach for the detection of distributed denial-of-service attacks[J].International journal of network management, 2021,31(0):2160.

[3] 季玉琦，严亚帮，和萍．基于 K-Medoids 聚类与栅格法提取负荷曲线特征的 CNN-LSTM 短期负荷预测 [J]. 电力系统保护与控制，2023,51(18):81-93.

[4] 郝荣丽，胡立华．一种基于属性值权重的 k-modes 聚类分析算法 [J]. 计算机与数字工程，2023,51(5):1001-1004+1119.

[5] 李静波，顾园园．基于模糊 K-Means 的 MBD 随机样本分类仿真 [J]. 计算机仿真，2023,40(8):473-477.

[6] HUANG S, KANG Z, XU Z. Robust deep K-means: an effective and simple method for data clustering[J].Pattern recognition,2021,117:107996.

[7] 赵鑫，汪丽娟，行艳妮．改进的 CK-means 优化及并行策略 [J]. 计算机应用研究，2020,37(11):3287-3291.

[8] ZHAO H.Research on improvement and parallelization of Canopy-Kmeans clustering algorithm[C]//2021 International Conference on Electronic Information Engineering and Computer Science.Piscataway:IEEE,2021:455-458.

[9] ZHANG G, ZHANG C, ZHANG H.Improved K-means algorithm based on density canopy[J].Knowledge-based systems,2018,145:289-297.

[10] KUMAR A, INGLE Y S, PANDE A.Canopy clustering: a review on pre-clustering approach to K-means clustering[J]. International journal of innovations & advancement in computer science,2014,3(5):22-29.

【作者简介】

叶帅辰（1994—），男，辽宁沈阳人，博士，工程师，研究方向：网络安全。

（投稿日期：2024-06-01）