doi 10.3969/j.issn.1671-1122.2016.06.001

Spark 框架下基于无指导学习环境的网络流量异常检测研究与实现

吴晓平，周舟，李洪成

（海军工程大学信息安全系，湖北武汉 430033）

摘　要：针对海量数据进行入侵检测的困难性问题，文章设计并实现了一套基于 Spark 框架的网络流量无指导学习异常检测系统。数据的预处理采用 Python 和 Python 的数据升级版 IPython 实现，异常检测采用无指导学习环境下的快速聚类方法 *K*-means 预测以及划分流量方法，记录所代表的攻击类型。为了避免 MapReduce 等传统分布式计算框架频繁的硬盘读写带来的巨大时间开销，文章设计实现了 Spark 框架下的 *K*-means 异常检测方法，通过将每轮迭代产生的临时数据存入内存而非硬盘中，有效提高了 *K*-means 聚类检测算法的计算效率。此外，为解决 *K*-means 算法中 *K* 值选取难的问题，通过 Spark 迭代计算与比较不同 *K* 值下的 *K*-means 算法中各聚类中心到所属簇中所有点距离的平均值，实现最佳 *K* 值的选取。最后，对系统进行了性能和功能测试，测试结果表明该系统达到了预定的设计要求，具有很高的计算效率和检测准确性。

关键词：网络流量检测；Spark；指导学习

中图分类号：TP309　　文献标识码：A　　文章编号：1671-1122（2016）06-0001-07

中文引用格式：吴晓平，周舟，李洪成 . Spark 框架下基于无指导学习环境的网络流量异常检测研究与实现 [J]. 信息网络安全，2016（6）：1-7.

英文引用格式：WU Xiaoping, ZHOU Zhou, LI Hongcheng. Research and Implementation on Network Traffic Anomaly Detection without Guidance Learning with Spark[J]. Netinfo Security, 2016 (6) : 1-7.

Research and Implementation on Network Traffic Anomaly Detection without Guidance Learning with Spark

WU Xiaoping, ZHOU Zhou, LI Hongcheng

(*Department of Information Security, Naval University of Engineering, Wuhan Hubei* 430033, *China*)

Abstract: In view of the massive data intrusion detection, this paper designs and implements a network traffic anomaly detection system based on Spark framework. Data preprocessing use Python and Python data, an upgraded version of the IPython implementation. Anomaly detection uses *K*-means predict and classify flow records represent the type of attack. In order to avoid time overhead uses traditional distributed computing framework, this paper designs and implements an anomaly *K*-means detection method under the framework of Spark. The method storages temporary data into memory rather than the hard drive, and improve the computational efficiency. In order to solve the problem of *K* value select difficult, through the Spark iterative calculation and comparison of the different K-means value of the *K* algorithm in the cluster center to all points in the cluster average value of all points, to achieve the best selection of *K* value. Finally, the performance and function of the system are tested. The test result shows that the system achieves the predetermined design requirements, and has high computational efficiency and detection accuracy.

Key words: network traffic detection; Spark; guiding learning

收稿日期：2016-04-28

基金项目：国家自然科学基金 [61100042]；湖北省自然科学基金 [2015CFC867]

作者简介：吴晓平（1961—），男，山西，教授，博士，主要研究方向为信息安全、密码学；周舟（1994—），男，云南，本科，主要研究方向为网络安全、并行计算、数据挖掘；李洪成（1991—），男，河南，博士研究生，主要研究方向为信息安全、数据挖掘。

通信作者：周舟 super_big_hero@sina.com

0 引言

目前，通过长期潜伏对目标进行监测并进行广泛部署的攻击行为，越来越受到黑客的青睐。面对日益严峻的安全形势，安全人员如何提前预警攻击是亟需解决的关键问题[1]。海量网络流量的异常检测是新型网络攻击检测的重要技术，但数据的海量性使得处理效率无法满足实时性要求[2-4]。云计算下的数据挖掘可以利用多台计算机之间形成的计算群来计算巨大的数据，具有计算可扩展、计算可广播、不受地域影响等优点，因此有必要研究利用云计算工具挖掘网络流量中的异常行为的方法，为海量网络流量分析提供解决途径[5-7]。

现有的基于数据挖掘的流量异常检测方法主要有基于有指导学习环境的流量异常检测方法和基于无指导学习环境的流量异常检测方法两类。其中，基于有指导学习环境的流量异常检测方法首先需要事先获得一定数量的带标记的训练数据集，然后对另外的测试目标流量数据进行检测，所以难以进行实时流量检测；基于无指导学习环境的流量异常检测方法可以直接对目标数据进行分析，得出检测结果，检测效率较高[8-10]。因此本文选用基于无指导学习环境的流量异常检测方法。

现有的典型云计算数据处理平台主要有 Hadoop 平台和 Spark 平台。其中，Hadoop 平台的运行原理相对简单，但每轮迭代产生的临时数据均会在硬盘中进行读写，严重影响数据挖掘算法的运行效率；Spark 平台则将临时数据存储于内存中，非常适合数据挖掘算法。因此本文利用 Spark 分布式计算平台进行无指导学习异常流量检测算法的部署[11,12]。

本文首先分析了 Hadoop 和 Spark 等大数据计算平台的运行机制和现有的基于无指导学习环境的流量异常检测方法，然后介绍了 Spark 框架下基于无指导学习环境的网络流量异常检测系统的设计方案和实现方法。本文的研究对海量流量的实时异常检测具有一定的意义。

1 相关研究

1.1 大数据分布式计算模型

为了满足日益增长的大数据分析需求，研究者提出了许多用于大数据分析的开源计算模型。这些方案主要基于两种不同的基础原型：Hadoop 和 Spark。

Hadoop 是由 Apache Software Foundation 公司于 2005 年作为 Lucene 的子项目 Nutch 的一部分正式引入的，它是受到最先由 GoogleLab 开发的 MapReduce 和 GoogleFile System（GFS）的启发而开发的。2006 年，MapReduce 和 Nutch Distributed File System（NDFS）分别被纳入 Hadoop 的项目中。Hadoop 是最受欢迎的在 Internet 上对搜索关键字进行内容分类的工具，它也可以解决许多要求极大伸缩性的问题。Hadoop 原本来自于谷歌一款名为 MapReduce 的编程模型包。谷歌的 MapReduce 框架可以把一个应用程序分解为许多并行计算指令，通过大量的计算节点运行非常巨大的数据集。使用该框架的一个典型例子就是运行在海量网络数据上的搜索算法。Hadoop 最初只与网页索引有关，现在迅速发展成为分析大数据的典型平台。

Spark 由加州大学伯克利分校 AMP 实验室开发，可用来构建大型的、低延迟的数据分析应用程序。Spark 是在 Scala 语言中实现的，它将 Scala 作为其应用程序框架。Spark 和 Scala 能够紧密集成，Scala 可以像操作本地集合对象一样轻松地操作分布式数据集。尽管创建 Spark 是为了支持分布式数据集上的迭代作业，但实际上它是对 Hadoop 的补充，可以在 Hadoop 文件系统中并行运行，通过名为 Mesos 的第三方集群框架可以支持此行为。Spark 设计理念与核心基于 RDD（Resilient Distributed Dataset），可以提供一站式多维度的大数据计算模型，可以在同一个技术堆栈中快速对数据集进行批处理、即席查询、机器学习、图计算和准实时流处理等。

与 Hadoop 相比，为了在处理大量数据上取得满意的速度，Spark 启用了内存分布数据集，除了能够提供交互式查询外，还可以优化迭代工作负载。Spark 拥有 Hadoop 中 MapReduce 所具有的优点，但不同于 MapReduce 的是，Job 中间输出结果可以保存在内存中，从而不再需要读写 HDFS，因此 Spark 能更好地适用于数据挖掘与机器学习等需要迭代的算法[13-15]。

1.2 国内外研究现状

网络流量异常检测过程首先需要使用 Sniffer、NetFlow、fprobe 和 flow-tools 等数据流抓取工具[16-18] 来采集海量的网络数据流信息，然后从数据中提取出可用于检

测异常流量的数据属性。常用的数据属性提取方法有以下两种：

1）直接以网络流量数据包头的各维数值作为数据属性[19]，如数据包的源/目的IP、源/目的端口、协议类型、数据包长度和时间等；

2）以网络流量的统计特性作为数据属性[20-22]，如固定时间内两主机间流量字节数、分组个数、数据流个数和流量熵等。

直接以网络流量数据包头的各维数值作为数据属性的检测方法主要包括基于无监督学习、监督学习和半监督学习的方法。其中，无监督学习的方法可以自动提取异常模式，但算法复杂度高、检测率低[23]；监督学习在时效性和准确性上较优，但需要大量经过标记的样本[24-26]；半监督学习将监督学习和无监督学习进行结合，在准确性和标记样本之间取得了很好的折中[27]。由于网络流量数据体量巨大，且具有实时更新性，因此需要采用滑动窗口等方法[28]提高检测效率，进而实现在线检测。滑动窗口法的原理是：每隔一个测量间隔时间，将最新的测量数据加入到滑动窗口中并将最旧的测量数据剔除，保持滑动窗口长度不变，然后在该时间窗口内进行异常检测。

以网络流量的统计特性作为数据属性的异常流量检测方法比基于数据包头各维数值的检测方法更加高效。这类方法按照检测的范围可分为基于单链路流量的异常检测[29]和基于全网络流量矩阵的异常检测[30]两类。前者主要利用流量数据的时间相关性检测入侵，后者则综合利用流量的时间相关性和空间相关性[31]，采用多元统计分析方法进行检测。由于全网络流量矩阵具有高维特性，因此许多研究采用具有降维功能的主成分分析法（Principal Component Analysis，PCA）进行统计分析[32-34]。此外，考虑到PCA无法反映流量矩阵各维度之间的相关性，所以一些研究采用支持向量机（Supported Vector Machine，SVM）等机器学习方法[35]，通过关联各维数据来判断异常是否真的发生，进而降低误报率。

1.3 Spark运行模式与工作流程

Spark的运行模式多种多样，部署在单机上时，既可以采用本地模式运行，也可以采用伪分布式模式运行；当以分布式集群的方式部署时，也有众多的运行模式可供选择，这取决于集群的实际情况。底层的资源调度既可以依赖于外部的资源调度框架，也可以使用Spark内建的Standalone模式。目前，实现外部资源调度框架的模式包括相对稳定的Mesos模式，以及还在持续开发更新中的Hadoop YARN模式。

Spark基本工作流程如图1所示。

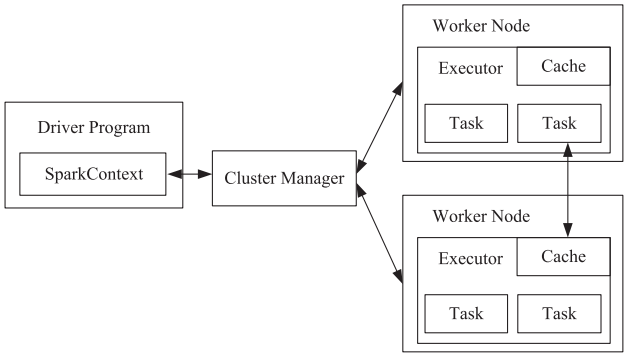

图1 Spark基本工作流程

从图1可以看到，所有的Spark应用程序都离不开SparkContext和Executor两部分。Executor负责执行任务，运行Executor的机器称为Worker Node；SparkContext由Driver Program启动，通过Cluster Manager和Executor通信。SparkContext和Executor两部分的核心代码实现在各种运行模式中都是公用的，在它们之上根据运行部署模式的不同，包装了不同调度模块以及相关的适配代码[36-38]。

以SparkContext作为程序运行的总入口，在SparkContext的初始化过程中，Spark会分别创建DAGScheduler（作业调度）和TaskScheduler（任务调度）两级调度模块。其中，作业调度模块是基于任务阶段的高层调度模块，首先它为每个Spark作业计算具有依赖关系的多个调度阶段（通常根据shuffle来划分），然后为每个阶段构建出一组具体的任务（通常会考虑数据的本地性等），最后以TaskSets（任务组）的形式提交给任务调度模块来具体执行。任务调度模块则负责具体启动任务、监控和汇报任务运行情况。

1.4 无指导学习环境下的流量异常检测方法

目前，研究者们提出了大量无指导学习环境下的聚类学习算法。而在具体应用中，聚类算法的选择取决于数据的类型、聚类的目的。如果聚类分析被用作描述或探查的工具，可以对同样的数据尝试多种算法，以发现数据可能揭示的结果。

主要的聚类算法可以划分为以下几类：划分方法、层

次方法、基于密度的方法、基于网格的方法以及基于模型的方法。

每一类方法中都有广泛应用的算法，如划分方法中的*K*-means 聚类算法，层次方法中的凝聚型层次聚类算法，基于模型的方法中的神经网络聚类算法等 [39,40]。

目前，聚类问题的研究不仅仅局限于上述的硬聚类，即数据只能被归为一类，模糊聚类也是聚类分析研究中较为广泛的一个分支。模糊聚类通过隶属函数来确定数据隶属于各个簇的程度，而不是数据对象硬性地归类到某一簇中。目前已有很多关于模糊聚类的算法被提出。

与其他聚类算法相比，*K*-means 聚类检测算法具有算法简单、计算快速等优点，因此在网络流量实时异常检测中得到了广泛的应用。*K*-means 聚类检测算法流程如图 2 所示。

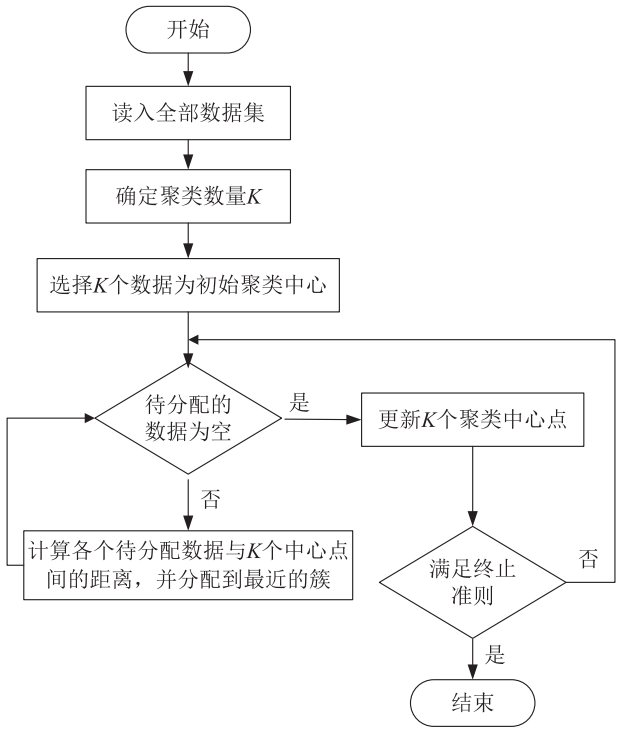

图2 *K*-means聚类检测算法流程

图 2 中，*K*-means 算法的目标是将一组数据点划分成 *K* 个簇。首先，设置 *K* 个聚类中心，计算每个类簇中聚类中心到类簇中所有点的平均值。然后，更换下一个 *K* 值，重复上一个步骤。最后，比较所有 *K* 值下的平均值，取平均值最小的 *K* 值作为最佳 *K* 值。

K-means 算法也存在缺点。算法初始值的选取比较复杂，很多时候 *K* 值的选取时间开销甚至大于 *K*-means 聚类的时间开销，因此在进行 *K*-means 聚类时必须解决 *K* 值确定等问题。

2 流量异常检测系统设计与实现

2.1 设计思想的提出与分析

随着大数据计算平台的普及，很多互联网公司都有了自己的大数据计算平台。一般这些数据挖掘平台负责分析用户的喜好，根据喜好给用户推荐商品或者文章等信息。本文将这部分计算资源分出一部分用于对访问的网络流量进行分析，这样可以极大地缓解运维部门面对新型网络攻击时所承受的压力。

传统的入侵检测系统面对当前的新型隐蔽网络攻击显得无能为力，大数据时代下的网络攻击防御系统要面对更加持续和有针对性的攻击者。攻击者可以在一年甚至十年时间内对攻击目标进行研究和渗透，一旦开始攻击，留给运维人员的防御时间是微乎其微的，这就造成了攻击者和被攻击者在时间上的不对称性，从而使得攻击者可以用时间上的优势来达到自己的目的。

为了打破这种时间上的不对称性，需要平时对网络流量进行监测和记录，形成详细的日志文件和分析数据，并利用高性能的数据分析平台对海量数据进行检测分析。本文实现的 Spark 框架下的流量异常检测系统，将实现实时分辨攻击类型的功能。

2.2 系统框架设计

本文流量异常检测系统的设计目标是实现对访问流量的关联分析，并预测攻击行为的发生，将处理好的数据放入大数据平台进行并行计算。系统框架如图 3 所示。

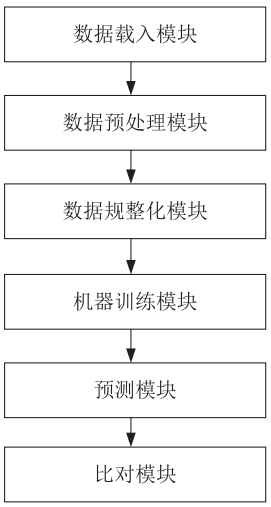

图3 流量异常检测系统框架

流量异常检测系统处理过程主要分为 6 个部分。

1) 数据载入模块

通过 SparkContext 将存储在本地的流量文件导入 Spark RDD 中，方便下一步的处理和分析。

2) 数据预处理模块

该模块主要负责将流量文件中的非数值型数据替换成数值型数据，方便下一步的计算和研究。

3) 数据规整化模块

由于数据预处理模块产生的数据存在矩阵过于稀疏的问题，该模块将稀疏矩阵转换成密集矩阵。

4) 机器训练模块

使用机器学习技术训练得到聚类模型，并通过计算各点和聚类中心距离的最小值，得到最优的聚类模型。

5) 预测模块

使用训练好的模型对流量数据进行预测和分类。

6) 比对模块

预测出流量数据所属的类簇，并和标签比对，计算检测准确性。

2.3 核心模块实现

由于其他模块可以直接调用 API 实现，因此本节主要关注数据预处理模块、机器训练模块和比对模块的实现。

2.3.1 数据预处理模块实现

启动 Python 进行数据预处理操作。数据预处理模块实现流程如图 4 所示。

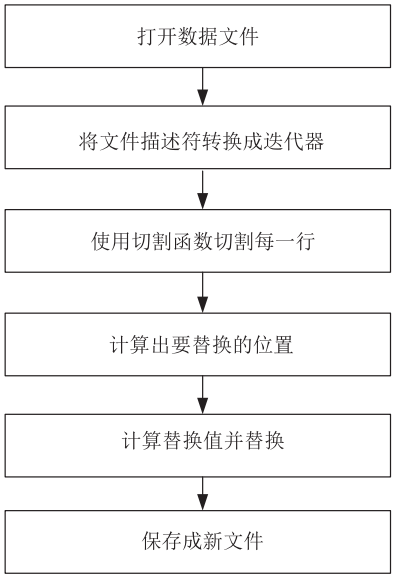

图4 数据预处理模块实现流程

数据预处理模块整个处理过程主要分为 6 个步骤。

1) 打开数据文件

使用 Python 的 *open*() 函数将文件以只读的形式打开，并载入内存中。

2) 将文件描述符转换成迭代器

使用 Python 自带的 *readlines*() 函数将返回的文件描述符转换成可迭代使用的迭代器。

3) 使用切割函数切割每一行

对使用迭代器返回的每一行文件，使用 *split*() 函数将文件以“ , ” 为分隔符切割。

4) 计算出要替换的位置

通过程序中的记数函数，计算出要替换的文字在列表中的位置，并返回该值。

5) 计算替换值并替换

根据返回的值计算出替换值，并将列表中的文字替换成该值。

6) 保存成新文件

调用 Python 中的 *writelines*() 函数将得到的结果保存为新文件。

2.3.2 机器训练模块实现

本系统利用 Spark 下的 *K*-means 函数对数据进行机器训练。在 Spark 函数库 MLlib 中的 *K*-means 算法具有以下参数 ：

```
class KMeans private (
    private var k: Int,
    private var maxIterations: Int,
    private var runs: Int,
    private var initializationMode: String
    private var iniaializationSteps: Int
    private var epsilon: Double,
    private var seed: Log)
    extends Serializable with Logging
```

其中，*k* 表示期望聚类的个数；*maxInterations* 表示方法单次运行最大的迭代次数；*runs* 表示算法被运行的次数（*K*-means 算法不能保证返回全局最优的聚类结果，因此在目标数据集上多次运行 *K*-means 算法，有助于返回最优聚类结果）；*initializationMode* 表示初始聚类中心点的选

择方式，目前支持随机选择或者 *K*-means++方式，默认是 *K*-means++方式；*initializationSteps* 表示 *K*-means++方式中的步数；*epsilon* 表示 *K*-means 算法迭代收敛的阈值；*seed* 表示集群初始化时的随机种子。

通常，先调用 KMeans.train 方法对数据集进行训练，该方法会返回 KMeansModel 类实例，然后使用 KMeansModel.predict 方法对新的数据点进行所属聚类的预测。KMeansModel.predict 方法接受不同的参数，可以是向量，也可以是 RDD，返回的是输入参数所属聚类的索引号。

2.3.3 比对模块实现

检测系统的比对模块实现流程如图 5 所示。

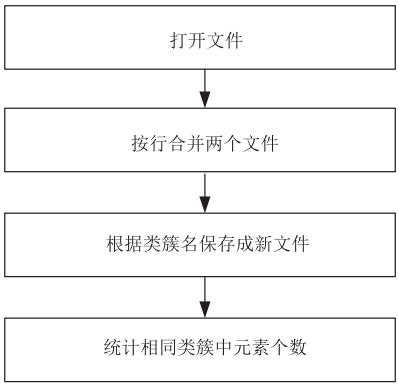

图5 比对模块实现流程

对比模块实现流程分为 4 个部分。

1）打开文件

使用 Python 的 *open*() 函数将两个文件载入内存。

2）按行合并两个文件

通过读取相同的行数将文件合并。

3）根据类簇名保存成新文件

从保存的文件中挑选出类簇名相同的文件进行合并，放入一个文件中。

4）统计相同类簇中的元素个数

对相同类簇中的元素个数做一个简单的统计。

3 系统性能和功能测试

本测试实验使用的数据集是 KDD 99 数据集。在检测中，*K* 值的选取是算法中的关键。本文将 *K* 值测试的步伐设置为一次 10 步，即第一次 *K* 值为 10，第二次 *K* 值为 20，第三次 *K* 值为 30，依此类推，最后通过比较得到一个最优的 *K* 值，并根据这个最优 *K* 值训练得到一个 best_model。通过训练出的 best_model，对数据进行分类和预测应属于的类簇。Spark 记录的 KDD 99 数据集中 corrected 数据集上的聚类时间是 70.434682 s。

类簇准确率通过合并各个 Spark 输出数据来计算每个类簇中所含的主要攻击种类个数与数据总数的比值来得到。

某个类簇准确率 *p* 的计算公式为：

$$p=\frac{m}{w} \quad \cdots\cdots (1)$$

其中，*m* 为类簇中数量占第一位的数据总数，即主要攻击种类个数，*w* 为类簇中所有数据总数。

整个数据集的总准确率 *P* 计算公式：

$$P=\frac{M}{W} \quad \cdots\cdots (2)$$

其中，*M* 为所有类簇中数量占第一位的数据总和，*W* 为所有类簇中所有的数据总和。

通过对数据的分析整合可以得出检测结果如表 1 所示，其中，本攻击记录数记录的为主要攻击种类个数。

表1 流量数据聚类检测结果

类簇编号	0	1	2	3	4	5	6
所代表的攻击	normal	neptune	smurf	neptune	normal	smurf	smurf
主要本攻击记录数	5340	2252	548	47221	8109	8691	10411
其他记录数	423	4	0	0	354	102	3
记录总数	5763	2256	548	47221	8463	8793	10414
检测准确率	92.66%	99.82%	100%	100%	95.82%	98.84%	99.97%
类簇编号	7	8	9	10	11	12	
所代表的攻击	teardrop	normal	normal	normal	ipsweep	neptune	
主要本攻击记录数	970	33184	11536	5258	1135	37266	
其他记录数	0	1098	26	63	327	36	
记录总数	970	34282	11562	5321	1462	37302	
检测准确率	100%	96.80%	99.78%	98.82%	77.63%	99.90%	
类簇编号	13	14	15	16	17	18	
所代表的攻击	smurf	normal	normal	neptune	normal	smurf	
主要本攻击记录数	34533	29015	4039	20457	424	193199	
其他记录数	1	1640	390	2267	322	0	
记录总数	34534	30655	4429	22724	746	193199	
检测准确率	100%	94.65%	91.19%	90.02%	56.84%	100%	

从表 1 可以看出，聚类分析后自动将数据分成了 19 类。通过公式（2）可以计算出总准确率 *P* 为 98.47%，准确率在可接受的范围内。

从表 1 也可以看出，通过对使用的算法和选取的步伐数进行优化来提高聚类效果是大有可为的，如第 4 簇中还包含了不少的其他攻击类型。

4 结束语

本文提出了基于 Spark 无指导学习环境的异常流量检测系统，实现了对网络流量的关联分析，在一定程度上提升了互联网企业应对持续性网络攻击的能力。下一步研究的主要方向为：基于 Scala 语言的 Spark 网络流量异常检测分析实现；基于 StreamingKMeans 函数实现对实时增量流量数据的在线分析等。为了加强对未知网络攻击的监测和预警，在现有算法模式

下还要加强未知类型监测算法的研究。（责编 潘海洋）

参考文献：
[1] 杨晓君 . 入侵检测报警数据处理技术研究 [D]. 哈尔滨 ：哈尔滨理工大学，2009.
[2] 戚名钰，刘铭，傅彦铭 . 基于 PCA 的 SVM 网络入侵检测研究 [J]. 信息网络安全，2015（2）：15-18.
[3] 陈晓梅 . 入侵检测中的数据预处理问题研究 [J]. 计算机科学，2006，33（1）：81-83.
[4] 王晓晔，张涛，郝亚培 . 网络入侵异常检测中数据预处理的研究 [J]. 天津理工大学学报，2013，29（6）：31-35.
[5] 李凯，薛一波，王春露，等 . 千兆网络入侵防御系统高速数据包处理的研究与实现 [J]. 小型微型计算机系统，2006，27（9）；1677-1681.
[6] 何鹏程，方勇 . 一种基于 Web 日志和网站参数的入侵检测和风险评估模型的研究 [J]. 信息网络安全，2015（1）：61-65.
[7] 李凯 . 千兆网络入侵防御系统包处理技术的研究 [D]. 北京 ：北京邮电大学，2006.
[8] 黄俊，韩玲莉，陈光平 . 基于无指导离群点检测的网络入侵检测技术 [J]. 小型微型计算机系统，2007，28（11）：2007-2009.
[9] 蒋盛益，李庆华 . 无指导的入侵检测方法 [J]. 计算机工程，2005，31（9）：31-33.
[10] 肖海军 . 基于 SVM 和无指导学习的入侵检测研究 [D]. 武汉 ：华中科技大学，2007.
[11] 周思伟 . Spark 大表等值连接的优化及其在网络流量数据分析的应用研究 [D]. 广州 ：华南理工大学，2015.
[12] 吴亚非，李新友，禄凯 . 信息安全风险评估 [M]. 北京 ：清华大学出版社，2007.
[13] 孙科 . 基于 Spark 的机器学习应用框架研究与实现 [D]. 上海 ：上海交通大学，2015.
[14] 尹绪森 . Spark 与 MLlib: 当机器学习遇见分布式系统 [J]. 程序员，2014（7）：112-115.
[15] 陈虹君 . 基于 Spark 框架的聚类算法研究 [J]. 电脑知识与技术，2015（2）：56-57.
[16] 孟浩，王劲松，黄静耘，等 . 基于 TcpFlow 的网络可视分析系统研究与实现 [J]. 信息网络安全，2016（2）：40-46.
[17] RENUKA D S, YOGESH P. A Hybrid Approach to Counter Application Layer DDoS Attacks[J]. International Journal on Cryptography and Information Security, 2012, 2(2): 45-52.
[18] BOLZONI D, CRISPO B, ETALLE S. ATLANTIDES: An Architecture for Alert Verification in Network Intrusion Detection System[C]//USENIX. The 21st Large Installation System Administration Conference, November 11-16, 2007, Dallas, Texas. Berkeley: USENIX, 2007: 141-152.
[19] 张玲，白中英，罗守山，等 . 基于粗糙集和人工免疫的集成入侵检测模型 [J]. 通信学报，2013，34（9）：167-176.
[20] ISO/IEC15408 Common Criteria for Information Technology Security Evaluation[S]. Geneva: IEC, 2004.
[21] 李晓勇，左晓栋 . 信息安全的等级保护体系 [J]. 信息网络安全，2004（1）：18-20.
[22] GB/T 20984-2007 信息安全风险评估规范 [S]. 国家质量监督检验检疫总局，2007.
[23] 李锦玲，汪斌强 . 基于最大频繁序列模式挖掘的 App-DDoS 攻击的异常检测 [J]. 电子与信息学报，2013，35（7）：1739-1745.
[24] CHANDRASEKAR A, VASUDEVAN V, YOGESH P. Evolutionary Approach for Network Anomaly Detection using Effective Classification[J]. IJCSNS Int Journal of Computer Science and Network Security, 2009, 9(1): 296-302.
[25] 陈晓，赵晶玲 . 大数据处理中混合型聚类算法的研究与实现 [J]. 信息网络安全，2015（4）：45-49.
[26] MABU S, CHEN Ci, LU Nannan. An Intrusion-detection Model Based on Fuzzy Class-association-rule Mining using Genetic Programming Network[J]. IEEE Transaction on Systems, Man, and Cybernetics, 2011, 41(1): 130-139.
[27] 陆悠，李伟，罗军舟，等 . 一种基于选择性协同学习的网络用户异常行为检测方法 [J]. 计算机学报，2014，37（1）：28-40.
[28] 王秀利，王永吉 . 基于命令紧密度的用户伪装入侵检测方法 [J]. 电子学报，2014，42（6）：1225-1229.
[29] DASH S K, REDDY K S. Adaptive Naive Bayes Method for Masquerade Detection[J]. Security and Communications Networks, 2011, 4(4): 410-417.
[30] 钱叶魁，陈鸣，叶立新 . 基于多尺度主成分分析的全网络异常检测方法 [J]. 软件学报，2012，23（2）：361-377.
[31] 刘大有，陈慧灵，齐红，等 . 时空数据挖掘研究进展 [J]. 计算机研究与发展，2013，50（2）：225-239.
[32] RINGBERG H, SOULE A, REXFORD J, et al. Sensitivity of PCA for Traffic Anomaly Detection[J]. Acm Sigmetrics Performance Evaluation Review, 2015, 35(1): 109-120.
[33] BRAUCKHOFF D, SALAMATIAN K, MAY M. Applying PCA for Traffic Anomaly Detection: Problems and Solutions[J]. IEEE INFOCOM, 2009, 34(1):2866-2870.
[34] RUBINSTEIN B I P, NELSON B, HUANG L, et al. Stealthy Poisoning Attacks on PCA-based Anomaly Detectors[J]. Acm Sigmetrics Performance Evaluation Review, 2009, 37(2): 73-74.
[35] 郑黎明，邹鹏，韩伟红，等 . 基于多维熵值分类的骨干网流量异常检测研究 [J]. 计算机研究与发展，2012，49（9）：1972-1981.
[36] 王洁松，张小飞 . KDDCup99 网络入侵检测数据的分析和预处理 [J]. 科技信息 : 科学 · 教研，2008（15）：10-17.
[37] RYZA S, LASERSON U, OWEN S, et al. Advanced Analytics with Spark[EB/OL]. http://www.bokus.com/bok/9781491912713/advanced-analytics-with-spark/, 2016-1-22.
[38] HARRINGTON P. Machine Learning in Action[M]. Greenwich: Manning Publications Co., 2012.
[39] 张士豪，顾益军，张俊豪 . 基于用户聚类的热门微博分类研究 [J]. 信息网络安全，2015（7）：84-89.
[40] William Wealey. Python for Data Analysis McKinney[EB/OL]. http://www.bokus.com/bok/9781449323622/python-for-data-analysis/, 2016-1-23.