# *网络安全数据可视化与可视分析案例*

**网络流量特征交互式聚类可视分析**

# 数据说明

## VAST Challenge 2013 MC3 数据集介绍

Big Marketing公司是一家以广告与公关为主营业务的公司，于2013年3月15日收到自称为“Butterfly Warriors”组织的恐吓信，该组织宣称将发起网络攻击，破坏公司的日常运营。从4月1日到4月15日的两个星期，公司的网络管理人员与该恐怖组织展开了一场激烈的网络攻防对抗，且为公众提供了这两个星期内的网络安全监控数据，试图破解黑客组织的攻击手段与攻击规律。

公司提供的网络安全监控数据集包含两周的NetFlow网络流量监控日志、Big Brother主机状态监控日志和一周的入侵防御系统日志，原始数据为CSV格式，涉及内网主机和服务器约1900台，其中NetFlow日志约7000万行，Big Brother日志约550万行，IPS日志约1600万行，原始日志量近10G。

**（1）Netflow网络流量监控日志数据集：**

Netflow日志数据集记录了一次完整网络连接行为（将一次网络连接的全部过程看作一次网络流），其数据结构如下表所示。该数据集共包含19项元素，分别表示一项网络活动中的时间、所通过的协议类型、源/目的地址、源/目的端口、持续时间、传输的源/目的包字节、源/目的包数目以及该次行为是否持续等。

表1-1 NetFlow流量监控数据的主要数据项与相关说明

|  |  |
| --- | --- |
| **数据项** | **相关说明** |
| TimeSeconds | 标准的UNIX时间值，指定为最后一个数据包流入的时间； |
| parsedDate | 解析后的时间，即更易于理解的时间格式； |
| dateTimeStr | 解析时间的字符串版本； |
| ipLayerProtocol | 流量所用协议的数值型编号，例如：TCP编号为6；UDP编号为17； |
| ipLayerProtocolCode | 流量所用协议的编号的文本型编号； |
| firstSeenSrcIp | 数据流中出现的第一个源地址； |
| firstSeenDestIp | 数据流中出现的第一个目的地址； |
| firstSeenSrcPort | 数据流中出现的第一个源端口； |
| firstSeenDestPort | 数据流中出现的第一个目的端口； |
| moreFragments | 标记该数据流是否是一个长数据流的一部分。当该字段不为0时说明该记录存在后续记录（持续发生），直到该字段的值为0为止； |
| contFragments | 标记该数据流是否是一个长数据流的一部分。当该字段不为0时说明该行数据不是从这里开始的； |
| durationSeconds | 该数据流传输时持续的时间，单位为秒； |
| firstSeenSrcPayloadBytes | TCP和UDP报文头所标识的有效载荷字节总和；TCP或UPD协议源地址发出的有效载荷字节数，该字段数据的计算方式为：表头长度（约55bt）\*包总数＋源流量＝源流量总数（加表头）； |
| firstSeenDestPayloadBytes | 协议目的地址接收的有效载荷字节数，该字段数据的计算方式为：表头长度（约55bt）\*包总数＋源流量＝源流量总数（加表头）； |
| firstSeenSrcTotalBytes | 源地址传输的数据流字节总数； |
| firstSeenDestTotalBytes | 目的地址接收到的数据流字节总数； |
| firstSeenSrcPacketCount | 从数据流中记录的第一个源地址传来的数据包数； |
| firstSeenDestPacketCount | 从数据流中记录的第一个目的地址接收的数据包数； |
| recordForceOut | 在传输有效时间截至前被存入数据文件的记录，通常用来表示程序关闭时未被正确传输的数据。 |

**（2）Big Brother主机状态监控日志数据集：**

Big Brother 是一个主机状态监控软件，通过运行该客户端软件，可以向服务端发送和记录重要服务器和主机的状态。数据中包括了对主机与服务器的网络连接能力、CPU和磁盘的使用率、内存和虚拟内存的使用情况以及SMTP邮件服务器的状态信息，如表1-2所示。表1-3列出了本文使用的Big Brother状态监控数据的主要数据项与相关说明。

表1-2 Big Brother状态监控数据的状态监控代码与相关说明表

|  |  |  |
| --- | --- | --- |
| **状态监控代码** | **状态监控代码说明** | **监控对象** |
| conn | 监测系统的网络连接情况 | 服务器和主机 |
| cpu | 检测系统的cpu使用率 | 服务器 |
| disk | 监测系统的硬盘使用率 | 服务器和主机 |
| mem | 监测系统内存的使用情况 | 服务器和主机 |
| pagefile | 监测系统的页面文件使用情况 | 服务器 |
| smtp | 监测SMTP邮件服务器的状态 | 邮件服务器 |

表1-3 Big Brother状态监控数据的主要数据项与相关说明表

|  |  |
| --- | --- |
| **数据项** | **相关说明** |
| ID | 数据的编号； |
| Hostname | 被监测主机的名称； |
| Servicename | 被监测主机的服务类型； |
| Currenttime | 日志数据的监测时间，时间为标准的UNIX时间； |
| statusVal | 被监测主机的状态值 |
| Bbcontent | 被监测主机的详细信息。根据不同的服务类型，记录的详细信息也不同。 |
| Receivedfrom | 被监测主机的IP地址。 |
| distUsagePercent | 当服务类型（Servicename）为‘disk’时，表示被监控主机硬盘的使用率。 |
| pageFileUsagePercent | 当服务类型（Servicename）为‘pagefile’时，表示被监控服务器的页面文件使用情况。 |
| numProcs | 当服务类型（Servicename）为‘cpu’时，表示被监控服务器的处理器数目。 |
| loadAveragePercent | 当服务类型（Servicename）为‘cpu’时，表示被监控服务器的平均装载率。 |
| physicalMemoryUsagePercent | 当服务类型（Servicename）为‘cpu’时，表示被监控服务器物理内存使用率。 |
| connMade | 当服务类型（Servicename）为‘conn’时，表示被监控的服务器和主机是否连接成功。 |
| parsedDate | 日志数据的监测时间，精确到秒 |

**（3）入侵防御系统日志数据集：**

入侵防御系统IPS（类似入侵检测系统IDS）通过预先设定的安全策略，以网络状况获取的数据为基础，通过规则匹配尽可能发现各种攻击行为，以保护网络安全。与IDS所不同的是，IPS可以主动的执行某些操作，比如对恶意报文进行丢弃，对恶意流量进行限流等等，并且通常安装在网络边际，与防火墙的位置类似，而IDS以检测与报警为主要功能。本文使用的IPS日志数据的主要数据项说明如下表所示。

表1-4 IPS日志数据的主要数据项与相关说明

|  |  |
| --- | --- |
| **字段** | **说明** |
| Date/time | 记录的产生时间 |
| Priority | 优先级，比如Info、Warning、Error、Alert等等 |
| Operation | 操作类型，有Built、Teardown、Deny等 |
| Message code | 相应的操作对应的消息码 |
| Protocol | 协议类型，有TCP、UDP、ICMP等 |
| Source IP | 源IP |
| Destination IP | 目的IP |
| Source port | 源端口 |
| Destination port | 目的端口 |
| Destination service | 目的端口所提供的服务 |
| Direction | 通常有inbound和outbound，表示连接方向 |
| Flags | 标记位信息 |

## 数据使用

本案例数据集使用情况如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **数据集来源** | **子数据集** | **是否使用** | **说明** |
| VAST Challenge 2013 MC3 | **（NetFlow）网络流量监控日志** | **√** | **仅使用4月1日到4月6日期间数据** |
| （Big Brother）主机状态监控日志 |  |  |
| （IPS）入侵防御系统日志 |  |  |

# 可视分析任务

本案例希望实现对Netflow网络流量日志进行可视化，了解网络流量活动的变化规律，借此分析网络活动所处状态，以此挖掘网络攻击行为、判断网络安全态势。具体包括如下几个分析需求：

**R1：查看网络流量时变情况数据。**用户可以关注网络连接数和网络总流量随时间变化的情况。

**R2：了解网络流量随机特性的时变模式。**网络流量数据对象主要分为源IP、源端口、目的IP、目的端口。对一个特定时间段和一种特定对象，比如某5分钟内的源IP，可能出现许多不同源IP参与网络活动并且每个源IP产生的流量相当（随机性大），也可能只出现一个源IP参与网络活动（随机性小），这反映了网络流量数据对象出现的随机性。用户希望能从四个角度（上述四类对象）查看网络流量随机性随着时间变化的特点。比如，用户能查看不同源IP的随机性随时间变化的特点。

**R3：掌握具有相似流量特征的时间片**。假设以一小时为间隔来分析网络流量，那么在一段时间内很可能出现某几个甚至多个小时流量相似，而且这些具有相似性的时间片可能连续分布也可能不连续分布。从时间角度，掌握具有相似流量特征的时间片，对网络攻击分析具有重要意义，因为同一种网络攻击手段，可能在不同时间多次使用，每次使用产生的流量特征很可能会相近。

**R4：展示网络活动在IP和端口上的分布状态**。在网络空间中，IP地址是主机的表示，端口是网络应用的表示，恶意的网络活动最终目标一般都是具体的主机或主机上的端口，而且许多网络攻击都会在主机分布和端口分布上形成鲜明的特征。用户可以查看网络活动在IP（包括源IP和目的IP）和端口（源端口和目的端口）上的分布特征，借此分析流量特征和检测异常。

# 数据处理

针对R1、R2需求，本案例需要对网络流量数据进行相应的处理。

针对两种网络流量统计维度：连接数和流量总数，本案例提供两种时间统计粒度（5分钟、1小时）对其进行统计；针对网络流量随机特性，我们定义了不同类型的信息熵，由此表示网络活动随机特性。

**（1）网络总连接数统计**

以5分钟为统计粒度得到的数据形式如下所示：

|  |  |
| --- | --- |
| **字段类型** | **取值说明** |
| 时间节点 | Ti（表示统计的开始时间节点） |
| 记录数 | N（表示从Ti到Ti+5min时间段内数据源中的总记录数，表现为连接数） |

**（2）网络总流量统计**

以5分钟为统计粒度得到的数据形式如下所示：

|  |  |
| --- | --- |
| **字段类型** | **取值说明** |
| 时间节点 | Ti（表示统计的开始时间节点） |
| 流量总数 | M（表示从Ti到Ti+5min时间段内数据源中的总流量值） |

**（3）信息熵的定义与计算**

在信息论中，信息熵常用于度量随机变量取值的不确定性程度，取值越是稳定（随机性低），信息熵就越低；反之，取值越是不稳定（随机性高），信息熵就越高。因而，在网络流量数据分析中，常将多种熵值进行组合分析，从而达到检测异常和识别特征的目的。

对于网络流量数据来说，主要分析对象为如下所示：

|  |  |
| --- | --- |
| **数据类型** | **相关说明** |
| 源IP | 数据流量传输过程中的源IP，即数据传输发起方 |
| 目的IP | 数据流量传输过程中的目的IP，即数据传输接收方 |
| 源端口 | 数据流量传输过程中的源端口，即数据传输发起应用 |
| 目的端口 | 数据流量传输过程中的目的端口，即数据传输接收应用 |

故本案例主要定义与网络流量活动相关的源IP地址熵、目的IP地址熵、源端口熵和目的端口熵。

目的端口熵值计算方法如下所示：

假设流量日志中出现的目的端口号为离散随机变量***A***，那么***A***的取值空间为***A=***｛*ai*， *i*=1..*m*｝，*m*因为是端口号，所以实际取值范围从0到65535，如果在某时间段内，每一种端口出现的次数为*ni*,*i*=1~m，目的端口号*ai*出现的概率表示为*pi* ，即：

（公式3-1）

则目的端口在该时段中的**信息熵**和**相对熵**为：

（公式3-2）

越大，表示活动的目的端口越随机，也就是说有更多的端口以较为相近的概率出现；相反，熵值越小，表示活动的目的端口越有序，也就是说只有少量的活动端口。

通常情况下，表明目的端口分布差异越大，而则表示端口活动分布越均匀，特别当所有的目的端口出现的概率都一样，则随机性最大，当某个目的端口出现的概率为100%，则随机性最小。

假设变量B、C、D分别表示源IP地址、目的IP地址、源端口，则这些信息熵计算定义如下所示：

源IP地址熵：

（公式3-3）

目的IP地址熵：

（公式3-4）

源端口熵：

（公式3-5）

以5分钟为采样时间间隔计算得到的信息熵数据形式如下所示：

|  |  |
| --- | --- |
| **字段类型** | **取值说明** |
| 时间节点 | Ti（表示信息熵计算的开始时间节点） |
| 不同信息熵类型 | A（表示待计算的信息熵类型） |
| 信息熵值 | H(A)（表示Ti到Ti+5min时间段内信息熵A的计算结果） |

# 可视化设计

## 总体设计

本案例可视分析方法引入基于熵值的网络流量特征分析，首先，将同一时刻的多个熵值看做多维数据点，结合RadViz可视聚类方法对多维熵值数据进行可视聚类，实现熵值在时序空间和聚类空间的交互可视分析；然后，提出基于矩阵图的网络活动可视化方法，以源IP熵、目的IP熵、源端口熵和目的端口熵的组合分析为例，设计了四个相互关联的矩阵视图，实现基于熵值的随机特性和基于矩阵图的分布特性的交互可视分析。

通过将熵值分析、聚类分析和可视分析相结合，有助于用户更好地理解和利用熵值传达的随机特性，而且可以借助网络活动分布特征定位网络异常的具体分布，并且快速有效地寻找其它类似特征的异常时刻，从而带给用户更强大的洞察力以掌握网络流量日志中隐含的复杂信息。

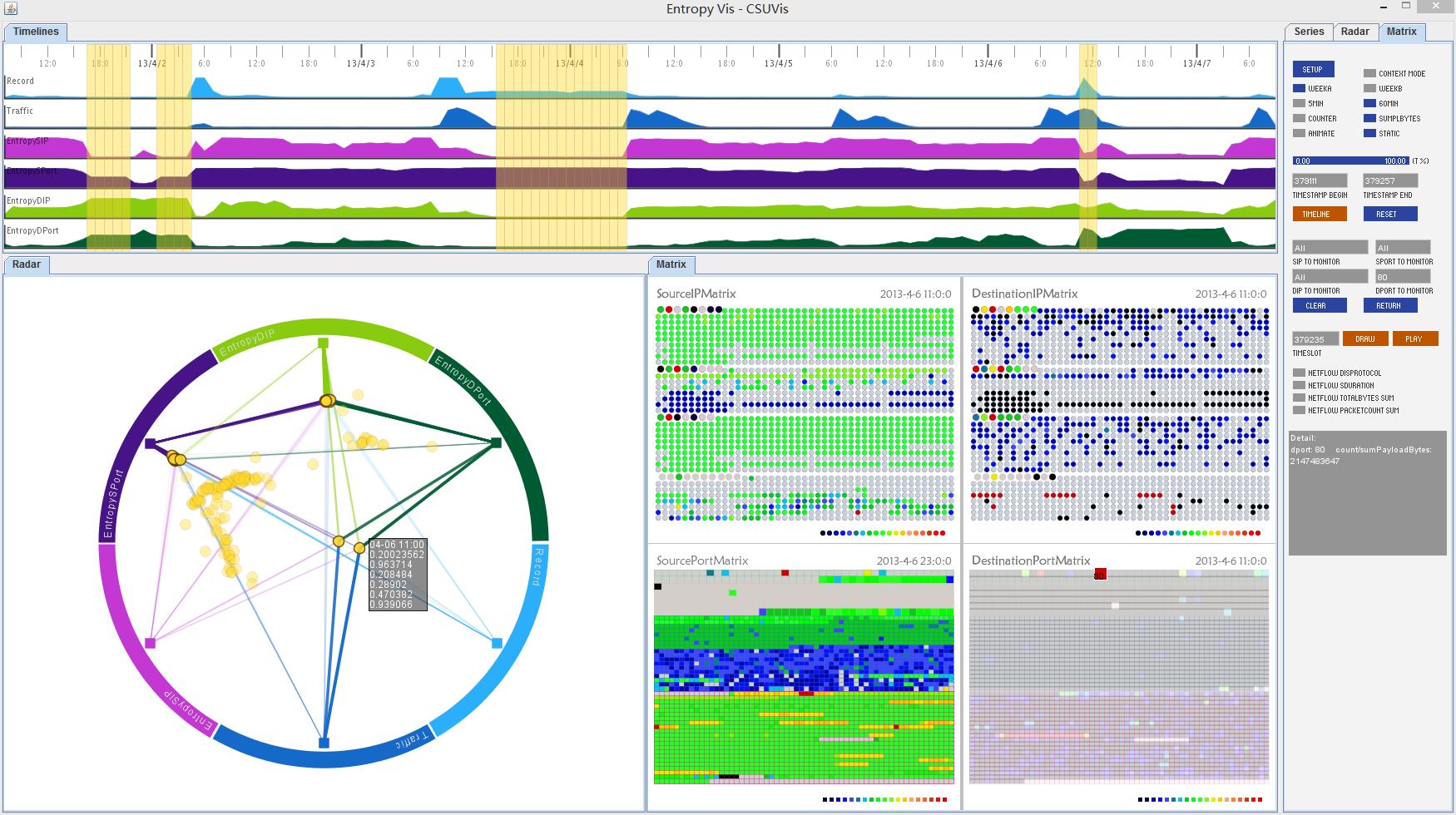


图4-1 基于信息熵的流量特征聚类与可视化的原型系统总体效果图

## 时序视图设计

根据R1、R2需求，本案例使用面积图对网络流量时序数据进行可视化，并使用通过不同的颜色表示不同的时序对象，且提供两种时序统计粒度视图。

本案例的可视化效果如下图所示：时序视图同时对一组时序同时进行可视化，从上至下分别是表示网络总连接数和总流量，以及表示熵值的源IP熵、源端口熵、目的IP熵和目的端口熵，每一条时间线：X轴表示时间，Y轴表示具体分析的属性值。为了便于区分，本案例针对不同的时序对象使用不同的颜色进行展示，即连接数是绿色、总流量是蓝色、源IP熵是品红、源端口熵是紫色、目的IP熵是草绿、目的端口熵是深绿。为了形成多粒度的分析，系统提供了1小时和5分钟两种时序采样间隔，1小时适合以天为单位的中长期分析，5分钟适合以小时为单位的短期分析。

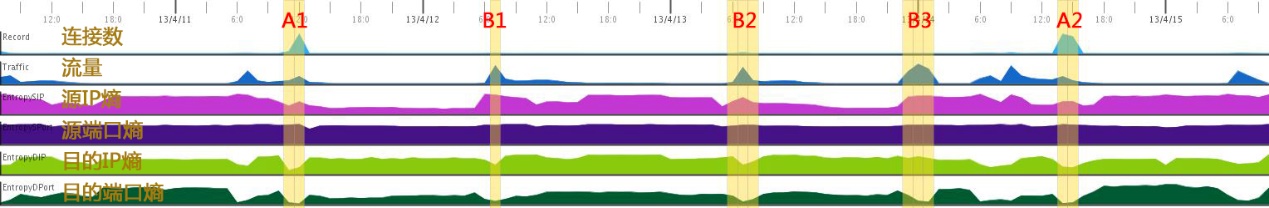
**D3实现：**

实际上，时序图的本质是面积图，D3提供d3.area()方法来定义面积生成器，从而绘制面积图。通常一个区域由两条边界线进行定义，可以是直线、折现或曲线。面积图通常由一条基线和曲线包围而成。由于时序数据的x轴是时间，故可视化时需使用d3.scaleTime()时间比例尺对其进行映射。

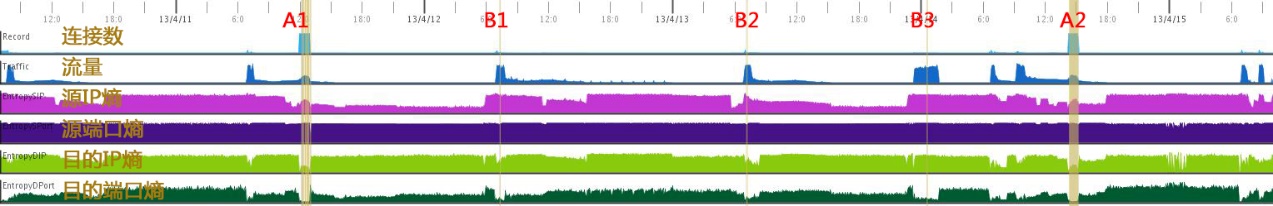
参考文档：

<https://github.com/d3/d3-scale/blob/v4.0.2/README.md#scaleTime>

[https://github.com/d3/d3-shape/blob/v3.0.1/README.md#**areas**](https://github.com/d3/d3-shape/blob/v3.0.1/README.md#areas)



（a）一小时采样间隔的时序图



（b）五分钟采样间隔的时序图

图4-2 时序示意图

## RadViz视图设计

根据R3需求，本案例使用RadViz对网络流量数据多维特征进行聚类分析，并使用D3进行实现。

**案例可视化设计：**

时序图中的时序数据可看作多维数据点集合*S*={*S*1,*S*2…*S*n}，每个数据点具有6个维度的属性。为了对该多维数据进行聚类，本案例使用RadViz圆环聚类方法将多维数据映射到二维的圆形区域内。

其中RadViz的数据点为时序图中每个采样点，RadViz的维度平均分布在圆环上，按顺时针角度维度分别为*V*={*V*网络连接数, *V*流量字节数, *V*源IP熵, *V*源端口熵, *V*目的IP熵, *V*目的端口熵}。我们使用不同的颜色来对不同维度进行标记，并在每个维度对应的圆环内写上对应圆环的名称。

数据点的位置由弹簧力模型确定。如果设一个*n*维数据记录为（*V*1，*V*2，···，*V*n），则该数据记录在RadViz中的位置可计算如下：

公式（4-3）

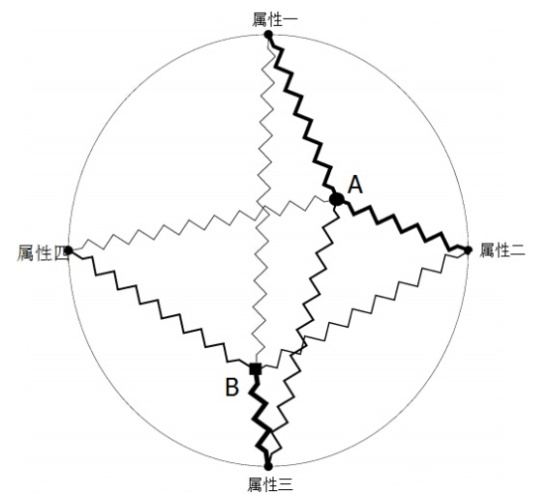


图4-3 RadViz基本原理示意图

**案例交互设计：**

RadViz视图提供了丰富的交互功能，这些交互功能可以帮助用户自由的探索数据类聚特征，主要交互手段包括：单选、多选、查看详细信息、关联更新、增加和减少维度、调整维度排序。选择功能可以让用户单选或多选圆内的数据点，查看数据点各维度的值，值的大小可以从弹出的信息框中显示，也可以通过查看与维度锚点连线的粗细程度而获知；关联更新功能可以让RadViz视图中的数据点和时序视图中的采样时刻点进行绑定显示，即用户在某个视图中任意点选，在另外一个视图中对应的将会高亮起来；任意维度组合功能可以让用户选择时序视图中任意的维度进行聚类分析，不同的维度组合将带来不同的聚类结果。

**D3实现：**

针对RadViz视图中的圆环部分，本案例使用D3.js中的d3.pie()与d3.arc()方法来进行环形图的绘制，其中pie进行布局的定义，arc进行元素的绘制。首先，选择需要展示的维度，且由于圆环上的维度是均匀分布，故需将其设置为相同的数据值，并通过d3.pie()方法可将对应数据转为如下格式：

*{"data": x, "value": x, "index": x "startAngle": x, "endAngle": x, "padAngle": x}*

分别表示维度名称、维度占比值、维度序号、对应圆弧开始角度，对应圆弧结束角度。值得注意的是，由于使用pie进行布局得到的圆弧开始角度为0，而本案例需将圆弧的中心所对的角度定义为0，故可使用foreach对pie数据中的"startAngle"和"endAngle"进行调整，具体如下所示：

*var data = pie(dataset);*

*data.forEach((element) => {*

*let shift = (2 \* Math.PI) / 6 / 2;*

*element.startAngle = element.startAngle - shift;*

*element.endAngle = element.endAngle - shift;*

*});*

然后，将调整后的pie数据传递给d3.arc()弧形生成器，通过定义arc.innerRadius()和arc.outerRadius()可生成对应弧形的path元素。最后，对各类警报弧形对应的颜色进行映射。

针对RadViz视图中的散点部分，本案例使用标准化后的多维数据，根据公式对各时刻数据点进行降维后，可得到对应时刻的x、y坐标值。然后使用合适的比例尺，如d3.linearScale()即可对计算后的x、y坐标进行可视化。

针对RadViz视图中的多选功能，本案例需对svg进行鼠标事件监听，分别对"mousedown"，"mousemove"和"mouseup"事件定义对应的行为。其中矩形选框的绘制使用svg中的rect元素进行可视化，rect元素的x、y、width、height属性分别使用d3.event.layerX和d3.event.layerY值进行动态更新。特别的，针对被框选的数据点，可在"mouseup"事件中根据数据点的位置进行过滤，同时添加.attr("class","node selected")属性对数据点进行区分。

针对RadViz视图中的关联更新功能，可依靠多选功能设定的"node selected" class类名来获取对应数据点的时刻，然后在时序视图上，利用事先设定好的时间比例尺，在对应时刻的位置绘制黄色的长矩形，矩形的中心对应该时刻，矩形的宽度为5min对应的宽度。

针对RadViz视图中的单选功能，可对选中的数据点与各个锚点进行连线。其中锚点的位置可用圆环中间所对应的arc.centroid()方法获取，并使用d3.line()方法对两点进行连线。线段的颜色根据对应维度的颜色进行映射，线段的粗细根据对应弹簧力的大小（或距离）进行映射，且距离越近，线段越粗，距离越远，线段越细。可专门设置一个d3.linearScale()作为弹簧力比例尺，对线段的粗细进行映射。

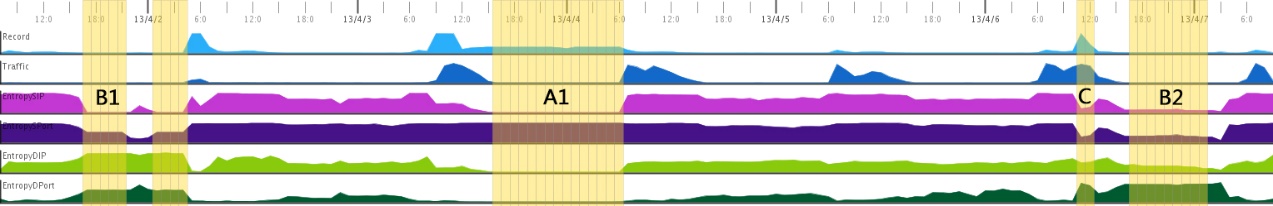
针对RadViz视图中的交互提示框功能，本案例通过监听数据点元素的点击事件来进行提示框的显示与隐藏。提示框本身可使用dom元素进行绘制，比如<div></div>；提示框的位置通常是用 d3.event 的 pageX 和 pageY进行定位；提示框的样式可使用style属性进行修改；提示框的内容包括选中数据点的时刻，及当前时刻标准化后的各维度：连接数、流量值、源IP地址熵、目的IP地址熵、源端口熵和目的端口熵。

针对RadViz视图中维度重组功能，只需用公式对散点坐标进行重新计算，再对圆环进行重新绘制即可实现。

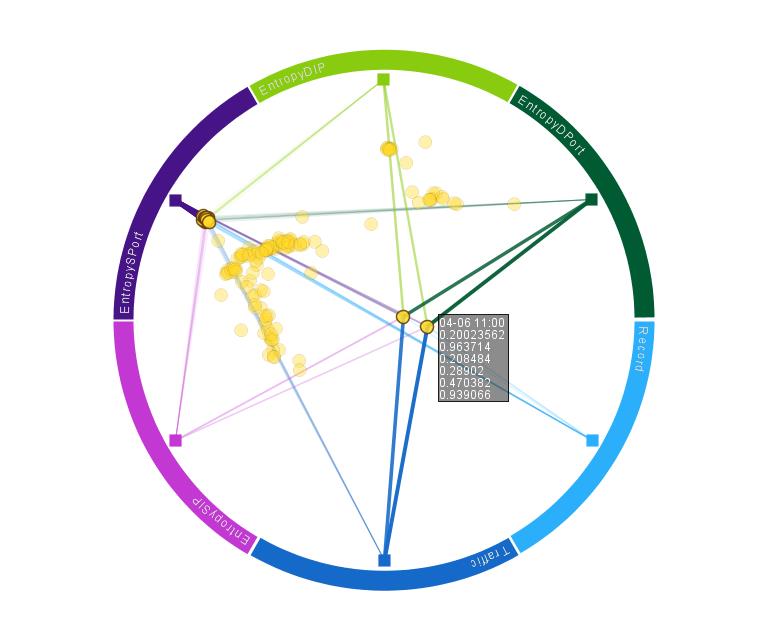
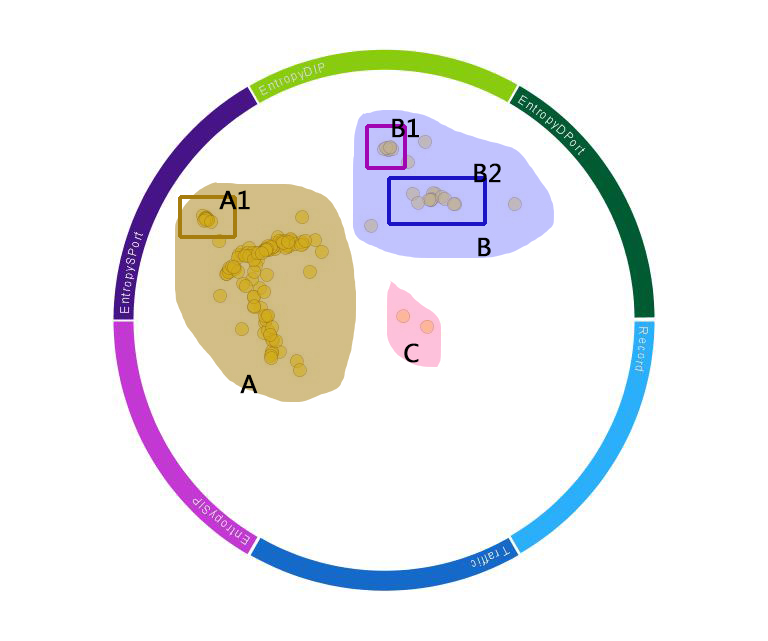
参考案例：

<https://blog.csdn.net/xyphf/article/details/83822192>

<https://blog.csdn.net/dlwbill/article/details/79169295>



（a）



（b） （c）

图4-4 基于RadViz的网络流量与熵值特征聚类. (a) 时序图. (b) 聚类效果示意图. (c) 交互选择效果示意图.

## 矩阵视图设计

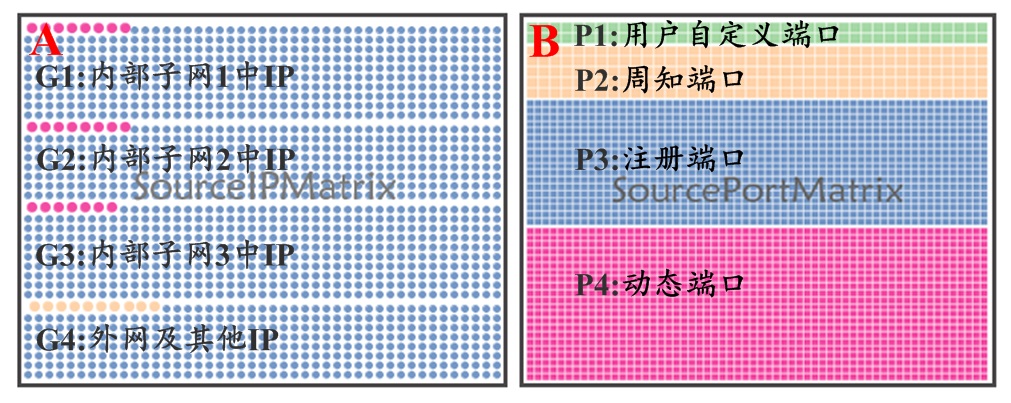
根据R4需求，本案例通过使用矩阵图来对IP和端口的分布进行可视化，并使用D3中的方法进行实现，并通过视图关联更新实现信息熵的随机性与流量的分布特征的融合分析。

**案例可视化设计：**

针对IP矩阵，本案例将被监控网络中的IP地址按照IP前缀进行分组，比如：实验中采用的Big Marketing公司的网络结构由三个分支网络组成，因此，在IP地址布局中三个不同子网的内网主机分为三组，外网主机和其余的特殊主机分为一组，位于同一组中的IP地址按从左到右、从上到下的顺序依次排列。

针对端口矩阵，本案例按优先级分组的方式将65536个端口分为了四组：（1）用户自定义端口：主要包括最常用的端口，这些端口都与重要网络服务或应用相关；（2）周知端口：主要包括0到1023号端口中除第一组以外的所有端口，这些端口也是网络服务或应用的常用端口；（3）注册端口：主要包括从1024到49511的被一些著名网络应用所注册的端口；（4）动态端口：包括前三组未申明的所有端口，与前面三组不同，考虑到这组端口的数量大和使用率低，将这一组端口进行再次分组，每100个端口作为一组被部署到一个网格单元中。

本案例共有四个矩阵视图，分别表示源IP和目的IP矩阵，源端口和目的端口矩阵。在IP矩阵中，主机和服务器用不同大小的圆点表示，每一组最前面较大的圆点表示各组的服务器，而较小的点则表示普通工作站。在端口矩阵中，每一个端口（第四组中为每100个端口）由一个方格表示，前面两组较大方格表示用户自定义端口和周知端口，后面两组较小方格表示注册端口和动态端口。每一个圆点或者方格的颜色用来编码流量大小，流量大小将按照从暖色调向冷色调进行编码。比如，黑色和蓝色等冷色调表示流量较低，橙色和红色等暖色调则表示流量较高。



(a) IP布局. (b) 端口布局.

图4-5 矩阵分组和布局示意图.

**案例交互设计：**

在交互手段方面，主要包括矩阵视图与其它视图以及矩阵视图内的关联更新。比如，矩阵视图可以与时序视图和RadViz视图进行关联更新，当用户在时序视图选中某个时刻，或者在RadViz视图中选中某个数据点，则矩阵视图将会显示该时刻的网络活动在IP和端口上的活动分布情况。如下图所示，用户在RadViz中选中一个数据点，在矩阵视图中可以看到该时刻的流量分布。

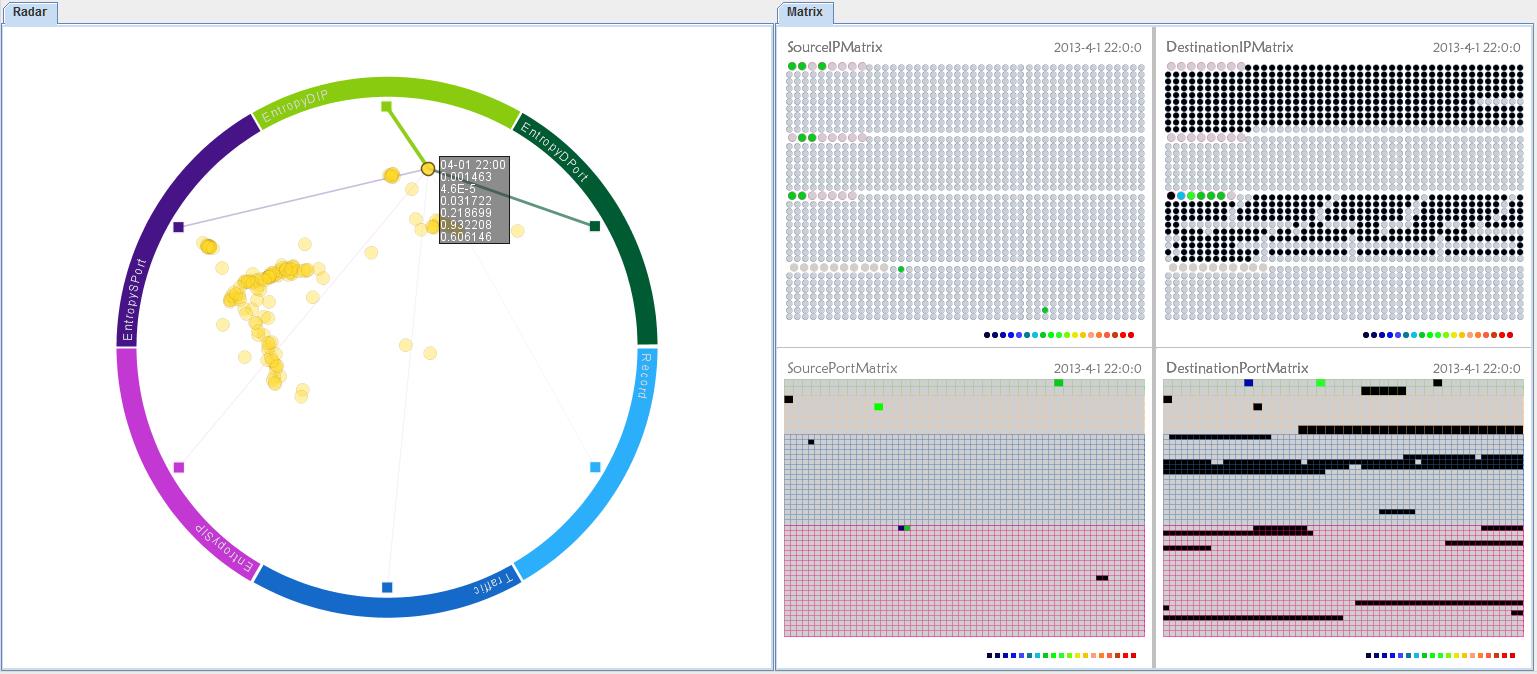


图4-6 RadViz视图与矩阵视图的关联更新示意图.

又比如，用户直接点击任意矩阵中的某一个圆点或者方块，其他的矩阵将相应地显示过滤后的数据集，这样的交互让数据流的方向也变得显而易见。

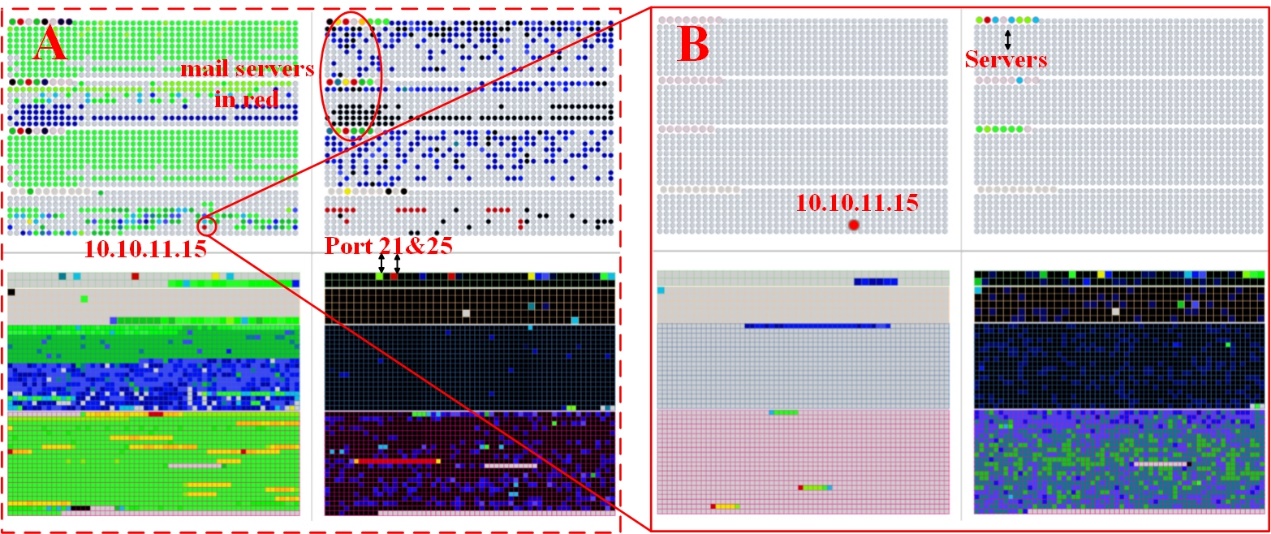


图4-7 矩阵视图内部的关联更新示意图.

**D3实现：**

针对矩阵视图的可视化分组部分，由于矩阵中各IP和端口的分组与条形图类似，只不过组成条形的不再是一个rect元素，而是特定数量的小圆点或小方块，故本案例使用与条形图类似的方法对其进行可视化。D3通常使用d3.scaleBand()条形比例尺来作为条形图的x轴比例尺，故本案例也使用d3.scaleBand()条形比例尺来对不同组的svg小元素进行不同条形的映射。其中小圆点使用circle元素进行绘制，小方块使用rect元素进行绘制，元素的位置可依靠各个元素的序号进行调整。以circle元素为例，通过使用绑定数据的序号值i可实现各个小圆点按顺序间隔排列的功能：

*.attr("cx", (d, i) => d3.scaleBand(name) + (i % 8) \* 10)*

*.attr("cy", (d, i) => height / 2 - parseInt(i / 8) \* 10);*

且针对IP视图，本案例通过调整circle元素的r属性来对主机进行重点显示。

针对矩阵视图的流量可视化部分，本案例使用d3.interpolateTurbo(t)方法对流量值进行颜色映射映射，数值越高则圆点或方格的颜色越暖，反之亦然。

针对矩阵视图的视图间关联功能，本案例需对时序视图中时刻点和RadViz视图中数据点的鼠标点击事件进行监听，即*element.on（"click"，function（d）{click函数}）*。根据this关键词和d3.event方法可获取时序视图或RadViz视图中选中的时刻；根据这个给出的时刻，可在初始数据中找出与之相关的目的IP节点和源IP节点、目的端口和源端口；再根据找到的端口号和IP号可对矩阵视图中对应的圆点和方块进行显示和隐藏。

针对矩阵视图的视图内关联功能，本案例需对四个矩阵视图中的圆点和方块进行鼠标点击事件监听。当用户直接点击任意矩阵中的某一个圆点或者方块，根据this关键词和d3.event方法可获取所选元素的目的（或源）IP（或端口）信息；根据给出的目的（或源）IP（或端口）信息，可在初始数据中找出与之相关的源（或）IP、目的端口和源端口信息，并通过d3修改对应圆点或方块的颜色属性，可实现对信息的过滤显示。

参考案例：

<https://observablehq.com/@irenedelatorre/grouped-dot-bar-chart-with-transitions>

# 案例分析

本案例选用了VAST Challenge 2013提供的Big Marketing公司的Netflow日志，时间是2013年4月1日到4月6日，按1小时的间隔，共计147个数据点。图5-1显示了这147个数据点的聚类结果，整体上可以将这周的网络流量按四个熵值以及连接数和字节数分为三大类，下面我们来分析各类数据点的特点。

（1）A类数据点反映的流量特征可视分析

A类是数据点最多的类，如图5-2（a）所示，所有A类数据共同的特点是源端口熵较大，一般来说，这说明有较多的源端口参与了网络活动，当然也可能是少量端口以相同的概率参与了网络活动，有矩阵图的辅助可以区分这两种情况。

进一步观察A类数据点，可以将A类数据点进一步分类4个子类。图5-2（b）显示了A1子类的时间分布，图5-2（c）显示了A2子类的时间分布，这两个子类的区别在于A1子类对应的时刻网络流量较小，而且A1子类是数据点最多的子类，也是本周网络流量的主要状态。A2子类的字节数较大，而且源IP熵与源端口熵相对较大，目的IP熵与目的端口熵相对较小。

图5-2（c）（d）分别显示了A3和A4子类的时间分布，A3小类的特点是网络连接数与字节数都非常大，源端口熵非常大，源IP熵较大，目的IP熵和目的端口熵较小，这是一个相对异常的时刻。A4小类的特点是字节数、目的IP熵和目的端口熵都非常小，连接数较大，源端口熵非常大，更值得注意的是A4这类数据点出现在连续的时间序列中，并且特征非常一致，没有太多起伏，这是一种不正常的现象。

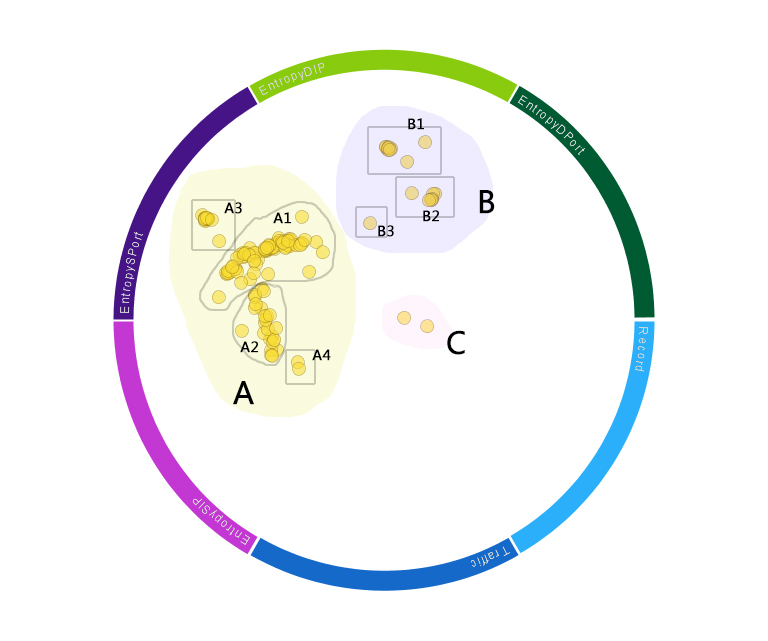
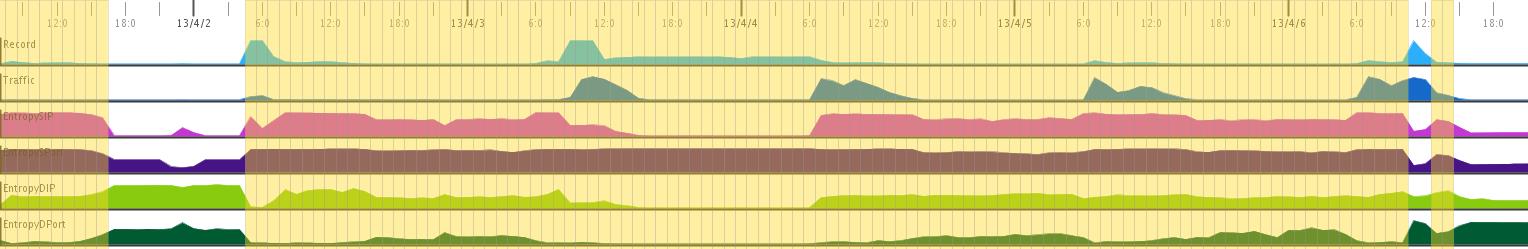
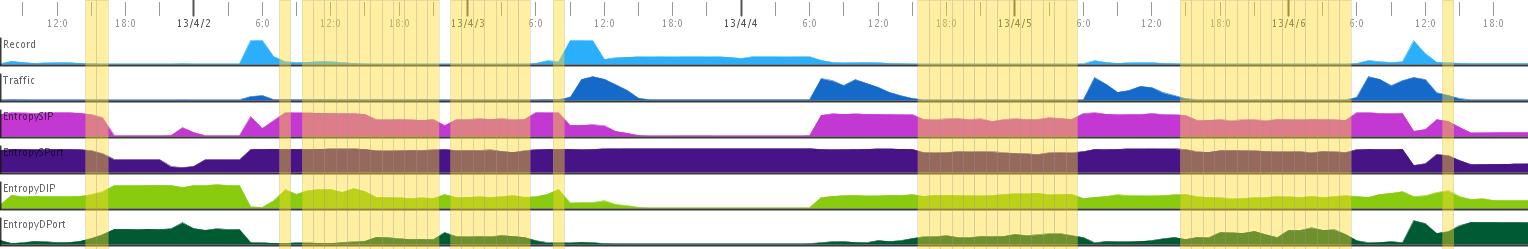


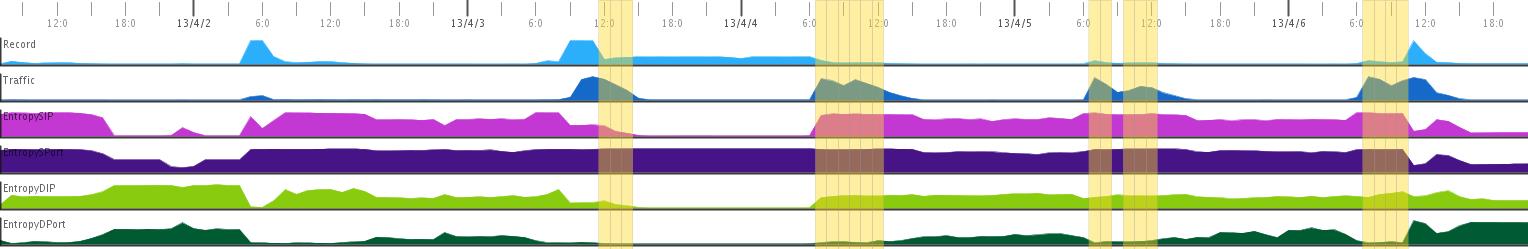
图5-1 Big Marketing公司网络流量特征聚类效果图



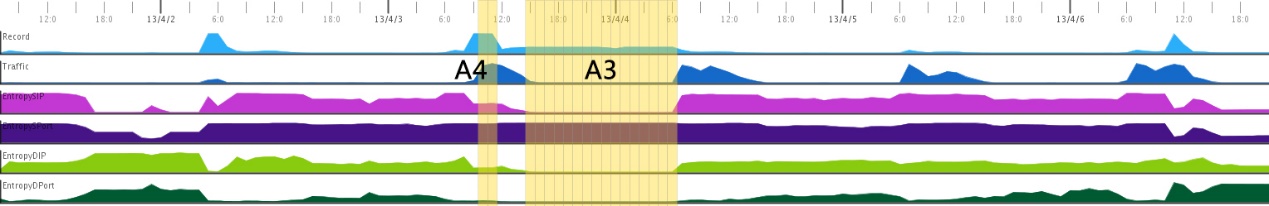
(a) 所有A类数据的时序分布



(b) A1子类数据的时序分布



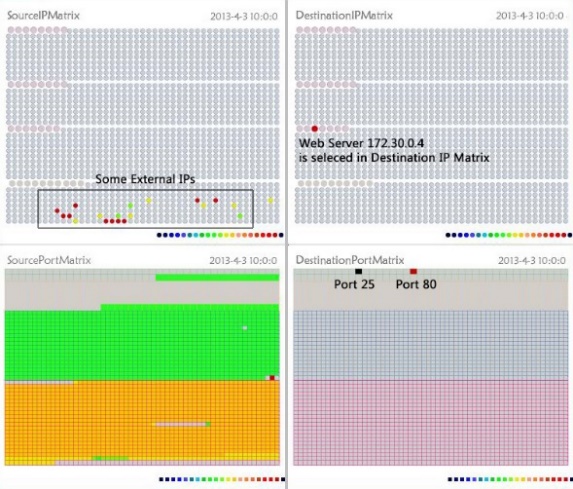
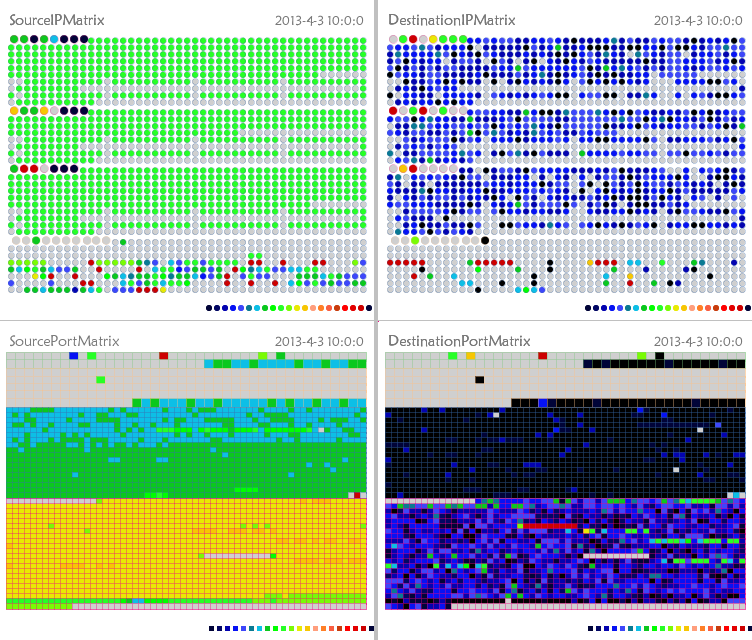
(c) A2子类数据的时序分布



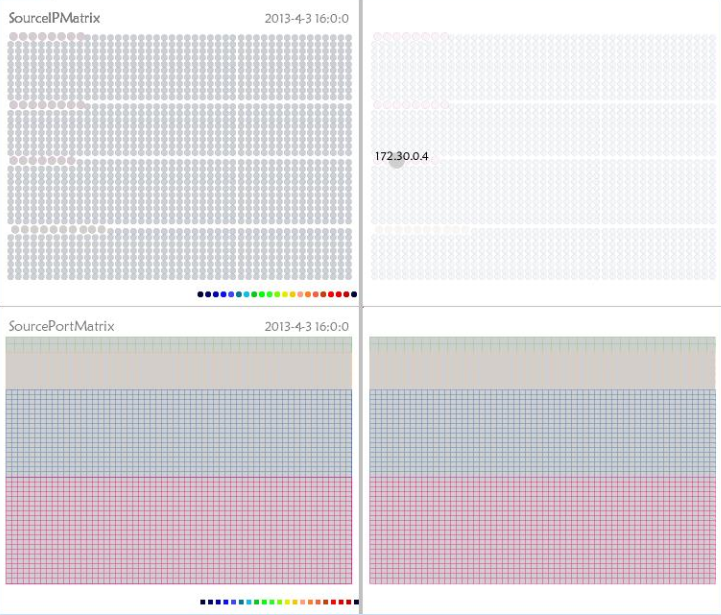
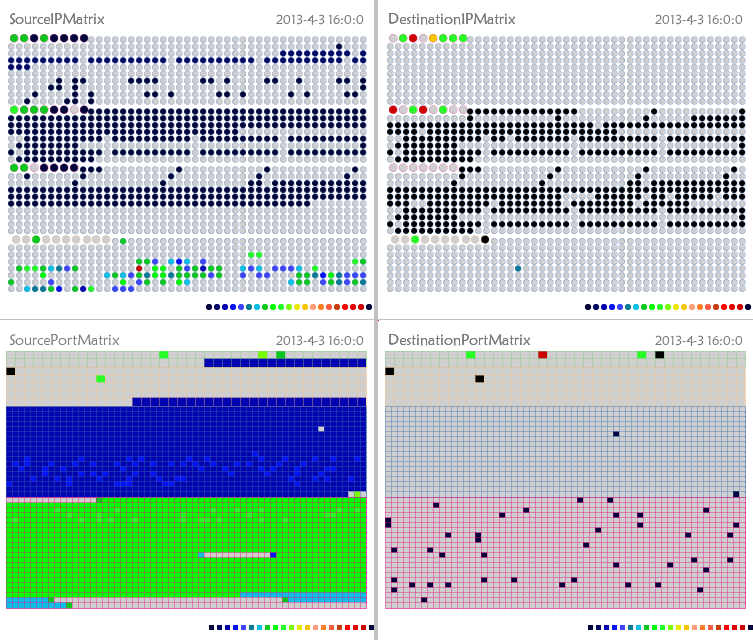
(d) A3和A4子类数据的时序分布

图5-2 A类数据点的时序分布图.

为了进一步分析A3和A4子类数据点的问题，图5-3显示这两子类数据点在矩阵视图中的可视结果。图5-3（a）是A3子类数据点在2013-4-3 10：00时的矩阵图，在两个IP矩阵中都有部分服务器呈现出红色活跃的状态，同时有一些外部IP地址也非常活跃，同时在两个IP矩阵中都有大量活动节点，因此源IP熵和目的IP熵值都处于中等水平。在源端口矩阵中，大量端口都有类似的活跃度，因此源端口熵值非常大，在目的端口矩阵中，虽然也有大量活动端口，但大部分黑色和深蓝色端口的流量非常小，流量都集中在一些红色活跃端口上，所以目的端口熵值并不大。图5-3（b）是以目的IP-172.30.0.4作为过滤条件的流量分布，可以很明显的识别出少量外部IP通过大量源端口攻击了Web服务器172.30.0.4的80号端口和25号端口，这是一次典型的DDoS攻击。



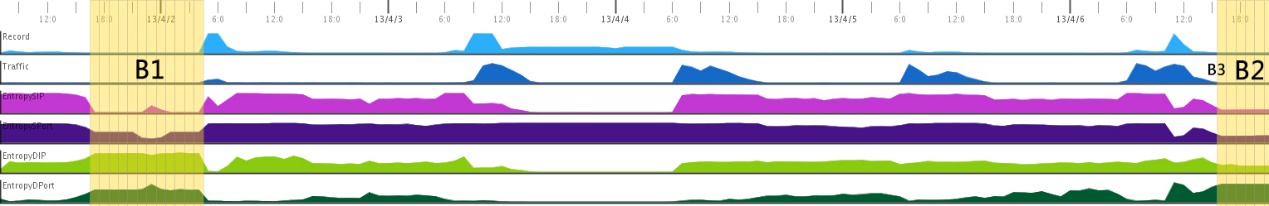
(a) (b)



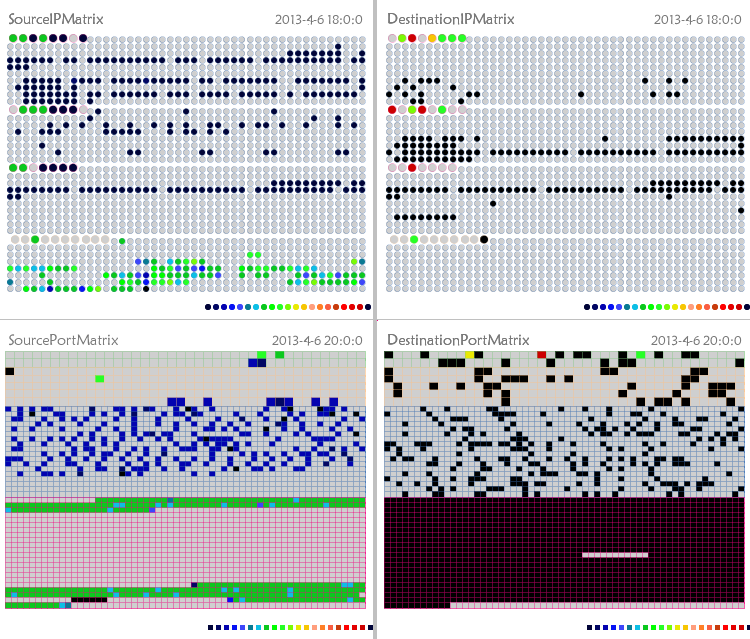
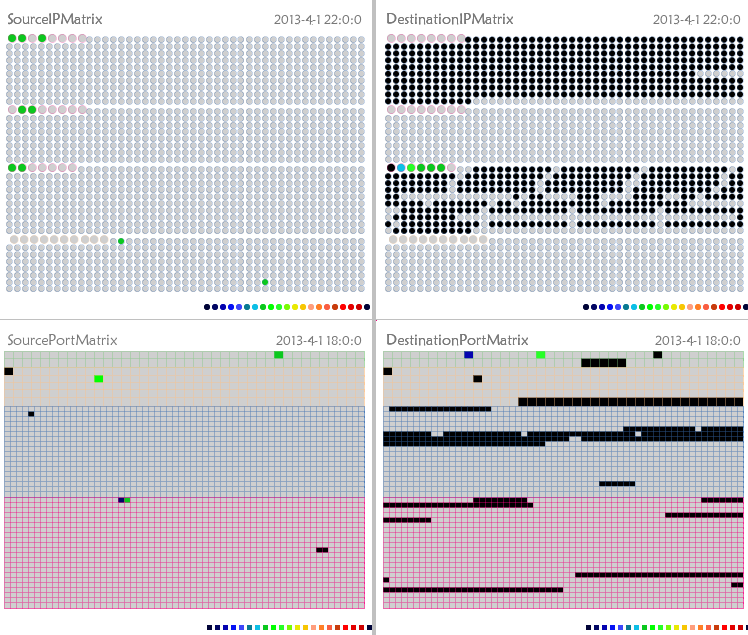
(c) (d)

图5-3 A类数据的两个数据点对应的矩阵视图. (a) 2013-4-3 10:00. (b) 2013-4-3 10:00 选中目的IP-172.30.0.4过滤后结果. (c) 2013-4-3 16:00. (d) 2013-4-3 16:00选中目的IP-172.30.0.4过滤后结果

图5-3（c）是A4子类数据点在2013-4-3 16：00时的矩阵图，源端口矩阵仍然有较明显的异常现象，这也和源端口熵较大而其它熵值都较小相匹配。在这个时刻，仍然选中目的IP-172.30.0.4作为过滤条件，则图5-3（d）显示出几乎没有任何网络活动的迹象，并且从4月3日下午到4月4日上午的所有时刻都有相似特征。如果与A3子类数据点关联起来考虑，则可以怀疑服务器172.30.0.4在遭受到DDoS攻击后陷入了瘫痪状态。



(a)



(b) (c)

图5-4 B类数据点的时序视图和矩阵视图. (a) B类数据点的时间分布. (b) 2013-4-1 22:00. (c) 2013-4-6 18:00.

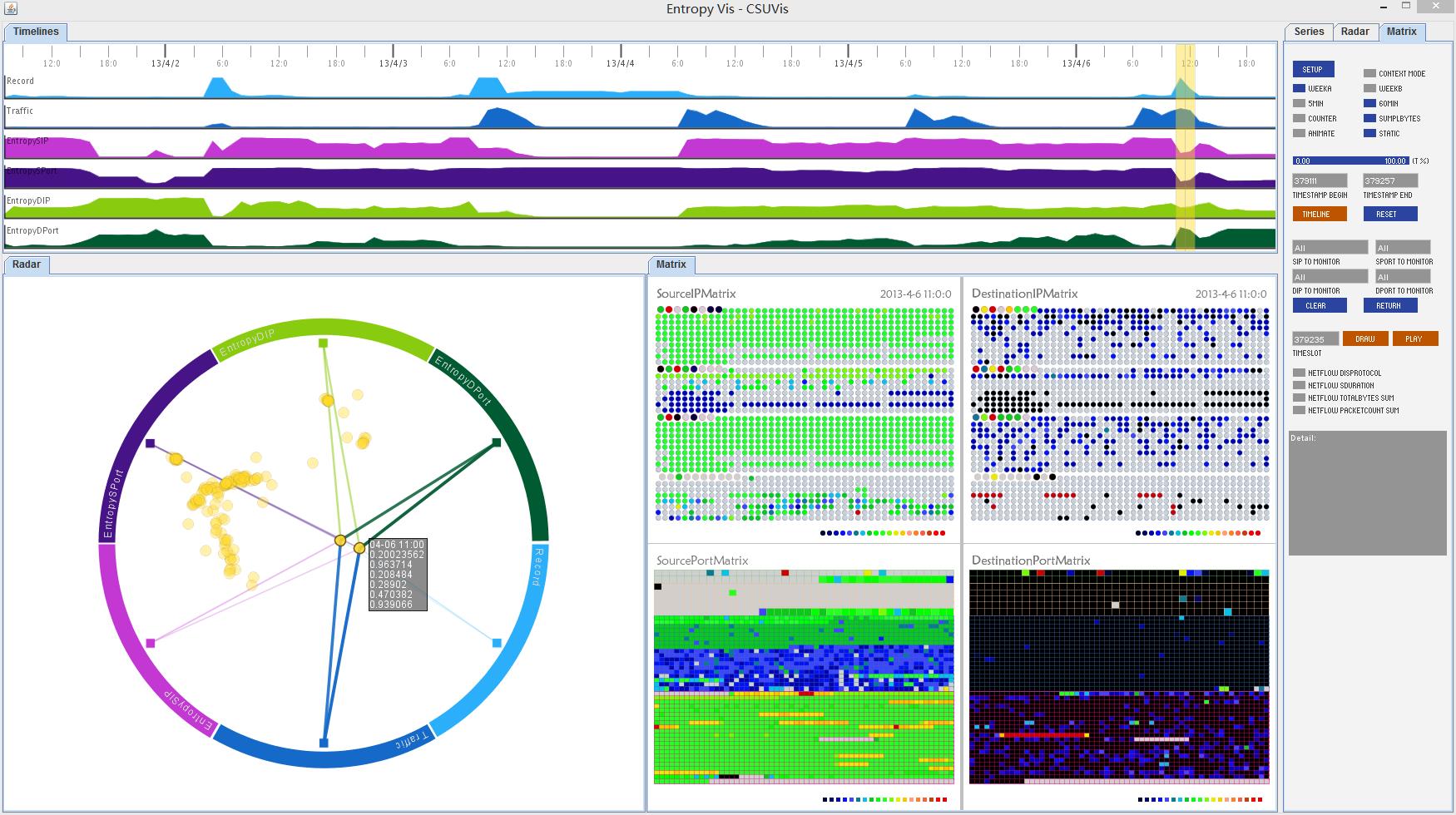


图5-5 C类数据点的流量特征

（2）B类数据点反映的流量特征可视分析

B类数据整体时序分布和数值大小如图5-4（a）所示，该类数据的主要特点是流量小、源IP熵和源端口熵小、目的IP熵大，而其子类B1与B2相比，B1的源IP熵更大。B3子类只有一个数据点，时序上紧挨B2子类，数值上与B2子类相近，但源IP熵和源端口熵要略大。

选取B1子类数据点2013-4-1 22：00进一步观察，如图5-4（b），这个时刻流量很小，活动IP与端口并不多，但是该时刻的源IP熵特别大，这是因为源IP矩阵中有10个浅绿色节点的流量非常相近，而少量活跃度相同的IP造成了较大的信息熵值。选取B2子类数据点2013-4-6 18：00，如图5-4（c），这个时刻流量也不大，但是目的端口矩阵有大量动态端口出现了网络访问，因此该时刻存在目的端口扫描的可能。

（3）C类数据点反映的流量特征可视分析

C类数据只有2个数据点，是三类中数据点最少的。如图5-5所示，首先从时序视图上可以观察到这两个数据点是4月6日11点和12点，反映网络流量的连接数和字节数都非常大，反映网络活动随机性的目的端口熵非常大，而源IP熵、源端口熵以及目的IP熵的值位于中等水平。在RadViz视图中，因为连接数、字节数和目的端口熵三个值较大，所以这2个数据点位置与其它数据点相比更靠近右下方这3个维度的中心位置。在矩阵视图中，可明显的看到在目的端口矩阵中几乎所有目的端口都有网络活动，而源端口矩阵中也有绝大部分端口参与了活动，而且两个IP矩阵也反映出很多活跃的主机。

综合来看，C类数据点是本周较为异常的时刻，流量大并且被访问的目的端口众多，图4-7显示的是4月6日11点与源IP矩阵中的一个流量较高的IP地址“10.10.11.15”相关的流量分布，很显然这台外部IP使用少量的源端口对内网服务器的所有端口进行了目的端口扫描。