(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 105306475 A
(43)申请公布日 2016.02.03

(21)申请号 201510753693.6

(22)申请日 2015.11.05

(71)申请人 天津理工大学
地址 300384 天津市西青区宾水西道 391 号 天津理工大学主校区

(72)发明人 王劲松 莫敬涛 黄玮 杨传印

(74)专利代理机构 天津佳盟知识产权代理有限公司 12002
代理人 侯力

(51)Int.Cl.
H04L 29/06(2006.01)

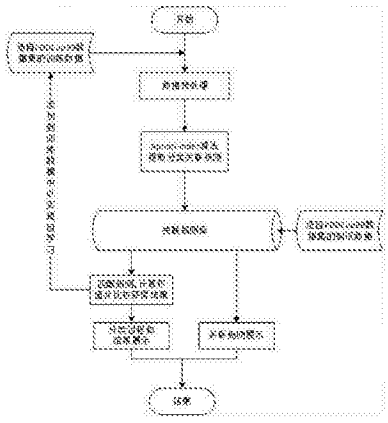
权利要求书3页 说明书10页 附图1页

(54)发明名称

一种基于关联规则分类的网络入侵检测方法

(57)摘要

一种基于关联规则分类的网络入侵检测方法，包括网络数据预处理，关联规则提取，网络连接数据分类和分类结果展示。本发明以改进的 Apriori 算法 (Apriori-index) 为基础，以国际标准数据集 KDDCup99 网络连接数据集为例，首先对选自其中的网络连接数据提取关联规则，然后根据关联规则实现对测试网络连接数据的分类，从而判断出当前网络连接是否为攻击连接以及具体攻击类型，并将相关统计数据展示出来。Apriori-index 算法更适用于 KDDCup99 数据集，大大提高了关联规则提取和网络连接分类的速度，检测结果的准确度也有提升，一定程度上改善了传统入侵检测系统分类慢，误报率高的缺陷。



CN 105306475 A

1.一种基于关联规则分类的网络入侵检测方法包括以下步骤：

第 1 步、对国际标准数据集 10％ KDDCup99 预处理，并将预处理后的数据集分成训练集和测试集两部分数据；

第 2 步、采用改进的 Apriori 算法 (Apriori-index) 对选取的训练集中的网络连接数据进行训练，提取到关联规则，将关联规则存放到关联规则库中，同时将关联规则库中的关联规则展示出来；

第 3 步、测试集中的每条网络连接数据逐条匹配关联规则库中关联规则，根据不同关联规则的条件长度和网络连接类型分别计算权值，找出最大权值所对应的网络连接类型即为最终分类得到的结果；

第 4 步、保存第 3 步中分类结果，将上述分类过程和分类得到的结果展示出来；同时为保证该方法良好的自学习特性，测试集的数据在根据关联规则分类得到具体的网络连接类型后，训练集数据连同对应的网络连接类型重新加入到训练集数据中，为后续关联规则提取提供新的训练集数据源，保证关联规则的动态更新。

2.根据权利要求 1 所述的基于关联规则分类的网络入侵检测方法，其特征在于：第 1 步中数据集预处理的方法是：

第 1.1 步、为每列数据添加位置参数；因为 10％ KDDCup99 数据集中有大量相同的数据，数据集中处于不同列的数据有不同的含义，而原始的 Apriori 算法在处理数据集中不同列的相同数据项时将他们视为同样的数据，因此直接使用原始的 Apriori 算法处理数据集会影响提取规则速度和分类结果的准确度；为避免出现以上问题，需要在数据预处理阶段为每条网络连接数据的每个数据项添加位置参数；

第 1.2 步、采用交叉验证的方法选取经过第 1.1 步预处理后的 10％ KDDCup99 数据集中 60％的连接数据作为训练集，剩余的 40％的连接数据作为测试集；由于改进的 Apriori 算法能够处理字符类型数据，同时数值类型的数据也能够视为字符类型数据，所以无需对网络连接数据中的字符类型数据进行数值化和归一化处理。

3.根据权利要求 1 所述的基于关联规则分类的网络入侵检测方法，其特征在于：第 2 步所述采用 Apriori-index 算法提取关联规则的方法是：

第 2.1 步、初始化最小支持度阈值 Min_Support，最小置信度阈值 Min_Confidence；通过查阅文献资料和实验验证，最小支持度阈值和最小置信度阈值分别设定为 25％和 78.5％能够获得较高的分类准确度；初始化最小支持度阈值 Min_Support ＝ 25％，最小置信度阈值 Min_Confidence ＝ 78.5％；

第 2.2 步、找出所有的频繁项集；遍历训练集中的所有的网络连接数据，统计每个属性值对应的连接类型及其出现的频度，形成候选项集合 C_1；在此基础上，根据支持度公式

$$Support(X)=\frac{Occur(X)}{Count(D)};$$

计算支持度；其中 Occur(X) 表示训练集中所有网络连接数据中包含频繁项 {X} 的数量，Count(D) 表示训练集 {D} 中所有网络连接的数量；在候选项集合 C_1 中删除支持度低于最小支持度阈值 Min_Support 的候选项，剩余的候选项形成频繁 1- 项集 L_1；然后对于每种网络连接类型，连接 L_1 中的不同元素构成候选项集合 C_2，再次遍历训练集数据，根据支持度

公式计算 C_2 中的每个候选项的支持度，删除候选项集合 C_2 中支持度低于最小支持度阈值 Min_Support 的候选项，剩余候选项形成频繁 2- 项集 L_2；按照网络连接类型，再连接 L_2 中的不同元素构成候选项集合 C_3，再次遍历训练集数据，计算 C_3 中的每个候选项的支持度，删除支持度低于最小支持度阈值 Min_Support 的候选项，剩余的候选项形成频繁 3- 项集 L_3；重复进行以上的遍历、删除和连接的步骤，直到没有新的候选项产生，所有的频繁项集 $(L_1, L_2, \ldots, L_n)$ 都已搜寻得到；其中，连接步骤和删除步骤分别严格满足连接定理和频繁子集定理，即：若两个 (k-1)- 项集的前 (k-2) 个项相同，而最后一个项不同，则证明它们可连接得到 k- 项集；若 k- 项集任意一个子集不是频繁项集，则该 k- 项集也不是频繁项集；

第 2.3 步、由频繁项集产生关联规则；对于第 2.2 步中得到的频繁项集 $(L_1, L_2, \ldots, L_n)$，假设频繁项集 L_i 中每个频繁项 l_i 的网络连接类型用 t_j 表示；如果 $(l_i-t_j) \rightarrow t_j$ 的置信度大于最小置信度阈值 Min_Confidence，则输出 $(l_i-t_j) \rightarrow t_j$；置信度的计算根据置信度计算公式：

$$Confidence(A \rightarrow B) = P(B \mid A) = \frac{Support(A \cup B)}{Support(A)};$$

其中 Support(A ∪ B) 和 Support(A) 分别表示频繁项 {A ∪ B} 和 {A} 的支持度；即置信度公式可化为：

$$Confidence\{(l_i - t_j) \rightarrow t_j\} = \frac{Support(l_i)}{Support(l_i - t_j)};$$

找到的所有满足要求的 $(l_i-t_j) \rightarrow t_j$，即为关联规则；

第 2.4 步、将第 2.3 步中得到的关联规则添加到关联规则库中，作为对测试集中未知类型的网络连接数据测试分类的判断依据；

第 2.5 步、将关联规则库中的关联规则展示出来；10％ KDDCup99 数据集的数据量较大，经过 Apriori-index 算法训练得到的关联规则非常多，关联规则在页面内显示会比较混乱，所以关联规则展示页面进行适当的缩放，使关联规则清晰展示。

4. 根据权利要求 1 所述的基于关联规则分类的网络入侵检测方法，其特征在于：第 3 步所述的确定最终网络连接类型的方法是：

第 3.1 步、读取测试集数据，对测试集中的每条网络连接数据按照关联规则分类，统计分类结果；10％ KDDCup99 数据集中每条网络连接数据有 41 个属性数据项和 1 个连接类型数据项，第 2 步中提取到的关联规则的条件部分包含有多个属性数据项，测试集中的每条未知类型的网络连接数据按照提取的关联规则分类时，会有多条关联规则与之对应，所以按关联规则分类需经过以下过程：

第 3.1.1 步、对读取到的测试集中的一条连接数据，遍历整个关联规则库，统计并记录匹配的关联规则 R_i（例如：$(l_i-t_j) \rightarrow t_j$）中条件部分 (l_i-t_j) 的长度，即统计关联规则 R_i 条件部分 (l_i-t_j) 包含的属性数据项的数量 $Length_i$；

第 3.1.2 步、分别计算匹配的 n 条规则中对应的网络连接类型部分 t_j 的权值；按照 Apriori-index 算法权值计算公式：

$$Weight(t_j) = \sum_{i=1}^{n} (Length_i * \log_2 Length_i + Length_i);$$

计算该条未知网络连接类型的测试数据在经过关联规则库中所有关联规则比对后匹配的第 j 种网络连接类型 t_j 的权值；这主要是由于关联规则条件部分 (l_i-t_j) 的长度 $Length_i$ 越大，分类准确度越高，这样做能够同其他关联规则加以区分，提高分类结果的准确度；

第 3.2 步、输出分类结果：网络连接类型 t；比较所有的权值，从中找出权值最高的 Weight(t)，将分类结果即网络连接类型 t 输出。

5. 根据权利要求 1 所述的基于关联规则分类的网络入侵检测方法，其特征在于：第 4 步所述的展示分类过程和分类后的网络连接类型以及向训练集中添加分类后的测试数据方法是：

第 4.1 步、测试数据展示；为将每条测试数据从读取直至分类完成过程展示出来，将每条测试数据用运动的图形代表，图形的运动轨迹和颜色变化代表测试数据的分类过程和分类得到的连接类型；

第 4.2 步、将测试过的网络连接数据与对应的网络连接类型添加到训练集中，保证该方法能够自学习；考虑到实际网络状况的动态特性，一次训练所得的关联规则不能始终代表网络的当前状况，在本方法中将每条分类后的测试数据连同其网络连接类型加入到训练集中并再次训练，实时训练产生新规则并更新到关联规则库中。

一种基于关联规则分类的网络入侵检测方法

技术领域

[0001] 本方法涉及网络入侵检测系统领域，尤其涉及到一种基于关联规则分类的网络入侵检测方法。

背景技术

[0002] 入侵检测通过收集和分析网络行为、安全日志、审计数据、其它网络上可以获得的信息以及计算机系统中若干关键点的信息，检查网络或系统中是否存在违反安全策略的行为和被攻击的迹象。它对于网络系统安全发挥着非常重要的作用，是防火墙的重要补充，入侵检测能够在不影响网络系统性能指标的情况下完成对网络系统的保护。

[0003] 将数据挖掘技术应用于网络入侵检测已成为一个研究的热点，国内外已出现不少这方面的研究成果，但仍存在如下一些不足和难点：多数数据挖掘的入侵检测系统集中于异常检测或误用检测，而异常检测具有较高的误报率，误用检测具有较高的漏报率；目前，多数系统属于准实时系统，不能及时对入侵做出检测并响应；面对不同的网络环境，以及不断改变的入侵类型，当前的网络入侵检测系统缺乏自适应性。

[0004] 将数据挖掘技术中的 Apriori 算法应用到入侵检测领域具有很强的理论基础，在技术上具有可行性。Apriori 算法提取的关联规则由频繁项集生成，规则具有很强的置信度，分类结果精确度较高，很好地避免了异常检测高误报率和误用检测高漏报率的缺陷。

发明内容

[0005] 本发明针对传统入侵检测系统的缺陷，提出了一种基于关联规则分类的网络入侵检测方法，通过采用 Apriori-index 算法来处理大量网络连接数据，提高入侵检测的及时性和准确性。通过在 10％ KDDCup99 实验数据集上测试，对比其他入侵检测算法，该算法的整体检测效果较优。

[0006] 本发明技术方案：

[0007] 一种基于关联规则分类的网络入侵检测方法，该方法包括以下步骤：

[0008] 第 1 步、对国际标准数据集 10％ KDDCup99 预处理，将预处理后的数据集分成训练集和测试集两部分数据。

[0009] 第 2 步、采用改进的 Apriori 算法 (Apriori-index) 对选取的训练集中的网络连接数据进行训练，提取到关联规则，将关联规则存放到关联规则库中，同时将关联规则库中的关联规则展示出来。

[0010] 第 3 步、测试集中的每条网络连接数据逐条匹配关联规则库中关联规则，根据不同关联规则的条件长度和网络连接类型分别计算权值，找出最大权值所对应的网络连接类型即为最终分类得到的结果。

[0011] 第 4 步、保存第 3 步中分类结果，将上述分类过程和分类得到的结果展示出来；同时为保证该方法良好的自学习特性，测试集的数据在根据关联规则分类得到具体的网络连接类型后，训练集数据连同对应的网络连接类型重新加入到训练集数据中，为后续关联规

则提取提供新的训练集数据源，保证关联规则的动态更新。

[0012] 第 1 步中所述的数据集预处理包括以下步骤：

[0013] 将关联规则算法应用到入侵检测方法中，主要是一种以数据为中心的观点，对于网络连接数据的采集处理不在本发明的考虑范围之内。本发明中以国际标准网络连接数据集 10％ KDDCup99 为例，以数据挖掘的思想为理论依据对入侵网络连接进行分类。

[0014] 第 1.1 步、为每列数据添加位置参数。因为 10％ KDDCup99 数据集中有大量相同的数据，比如，“0”和“1”，数据集中处于不同列的数据有不同的含义，而原始的 Apriori 算法在处理数据集中不同列的相同数据项时将他们视为同样的数据，因此直接使用原始的 Apriori 算法处理数据集会影响提取规则速度和分类结果的准确度。为避免出现以上问题，需要在数据预处理阶段为每条网络连接数据的每个数据项添加位置参数。

[0015] 第 1.2 步、采用交叉验证的方法选取经过第 1.1 步预处理后的 10％ KDDCup99 数据集中 60％的连接数据作为训练集，剩余的 40％的连接数据作为测试集。由于改进后的 Apriori 算法能够处理字符类型数据，同时数值类型的数据也能够视为字符类型数据，所以无需对网络连接数据中的字符类型数据进行数值化和归一化处理。

[0016] 第 2 步所述采用 Apriori-index 算法提取关联规则需要经过以下步骤：

[0017] 第 2.1 步、初始化最小支持度阈值 Min_Support，最小置信度阈值 Min_Confidence。通过查阅文献资料和实验验证，最小支持度阈值和最小置信度阈值分别设定为 25％和 78.5％可以获得较高的分类准确度。初始化最小支持度阈值 Min_Support ＝ 25％，最小置信度阈值 Min_Confidence ＝ 78.5％。

[0018] 第 2.2 步、找出所有的频繁项集。遍历训练集中的所有的网络连接数据，统计每个属性值对应的连接类型及其出现的频度，形成候选项集合 C_1。在此基础上，根据支持度公式

[0019] $$Support(X)=\frac{Occur(X)}{Count(D)};$$

[0020] 计算支持度。其中 Occur(X) 表示训练集中所有网络连接数据中包含频繁项 {X} 的数量， Count(D) 表示训练集 {D} 中所有网络连接的数量。在候选项集合 C_1 中删除支持度低于最小支持度阈值 Min_Support 的候选项，剩余的候选项形成频繁 1- 项集 L_1；然后对于每种网络连接类型，连接 L_1 中的不同元素构成候选项集合 C_2，再次遍历训练集数据，根据支持度公式计算 C_2 中的每个候选项的支持度，删除候选项集合 C_2 中支持度低于最小支持度阈值 Min_Support 的候选项，剩余候选项形成频繁 2- 项集 L_2；按照网络连接类型，再连接 L_2 中的不同元素构成候选项集合 C_3，再次遍历训练集数据，计算 C_3 中的每个候选项的支持度，删除支持度低于最小支持度阈值 Min_Support 的候选项，剩余的候选项形成频繁 3- 项集 L_3；重复进行以上的遍历、删除和连接的步骤，直到没有新的候选项产生，所有的频繁项集 $(L_1, L_2, \ldots, L_n)$ 都已搜寻得到。其中，连接步骤和删除步骤分别严格满足连接定理和频繁子集定理，即：若两个 (k-1)- 项集的前 (k-2) 个项相同，而最后一个项不同，则证明它们可连接得到 k- 项集；若 k- 项集任意一个子集不是频繁项集，则该 k- 项集也不是频繁项集。

[0021] 第 2.3 步、由频繁项集产生关联规则。对于第 2.2 步中得到的频繁项集 $(L_1, L_2, \ldots, L_n)$，频繁项集 L_i 中每个频繁项 l_i 中的连接类型用 t_j 表示。如果 $(l_i-t_j) \rightarrow t_j$ 的置信度大于最小置信度阈值 Min_Confidence，则输出 $(l_i-t_j) \rightarrow t_j$。置信度的计算根据

置信度计算公式：

[0022] $$Confidence(A \rightarrow B) = P(B|A) = \frac{Support(A \cup B)}{Support(A)};$$

[0023] 其中 Support(A ∪ B) 和 Support(A) 分别表示频繁项 {A ∪ B} 和 {A} 的支持度；即置信度公式可化为：

[0024] $$Confidence\{(l_i - t_j) \rightarrow t_j\} = \frac{Support(l_i)}{Support(l_i - t_j)};$$

[0025] 找到的所有满足要求的 $(l_i-t_j) \rightarrow t_j$，即为关联规则。

[0026] 第 2.4 步、将第 2.3 步中得到的关联规则添加到关联规则库中，作为对测试集中未知类型的网络连接数据测试分类的判断依据。

[0027] 第 2.5 步、将关联规则库中的关联规则展示出来。10％ KDDCup99 数据集的数据量较大，经过 Apriori-index 算法训练得到的关联规则非常多，关联规则在页面内显示会比较混乱，所以关联规则展示页面进行适当的缩放，使关联规则清晰展示。

[0028] 第 3 步所述的分类过程需经过以下步骤：

[0029] 第 3.1 步、读取测试集数据，对测试集中的每条网络连接数据按照关联规则分类，统 计分类结果。10％ KDDCup99 数据集中每条网络连接数据有 41 个属性数据项和 1 个连接类型数据项，第 2 步中提取到的关联规则的条件部分包含有多个属性数据项，测试集中的每条未知类型的网络连接数据按照提取的规则分类时，会有多条关联规则与之对应，所以按关联规则分类需经过以下过程：

[0030] 第 3.1.1 步、对读取到的测试集中的一条连接数据，遍历整个关联规则库，统计并记录匹配的关联规则 R_i（例如：$(l_i-t_j) \rightarrow t_j$）中条件部分 (l_i-t_j) 的长度，即统计关联规则 R_i 条件部分 (l_i-t_j) 包含的属性数据项的数量 $Length_i$；

[0031] 第 3.1.2 步、分别计算匹配的 n 条规则中对应的结果部分 t_j 的权值。按照 Apriori-index 算法权值计算公式：

[0032] $$Weight(t_j) = \sum_{i=1}^{n} (Length_i * \log_2 Length_i + Length_i);$$

[0033] 计算该条未知结果类型的测试数据在经过关联规则库中所有规则比对后匹配的第 j 种网络连接类型 t_j 的权值。这主要是由于关联规则条件部分 (l_i-t_j) 的长度 $Length_i$ 越大，分类准确度越高，这样做能够同其他关联规则加以区分，提高分类结果的准确度。

[0034] 第 3.2 步、输出分类结果：网络连接类型 t。比较所有的权值，从中找出权值最高的 Weight(t)，将分类结果（即网络连接类型 t）输出。

[0035] 第 4 步所述的展示分类过程和分类得到的结果以及向训练集中添加分类后的测试数据方法是：

[0036] 第 4.1 步、测试数据展示。为将每条测试数据从读取直至分类完成过程展示出来，将每条测试数据用运动的图形代表，图形的运动轨迹和颜色变化代表测试数据的分类过程和分类得到的连接类型；

[0037] 第 4.2 步、将测试过的网络连接数据与对应的网络连接类型添加到训练集中，保证该方法能够自学习。考虑到实际网络状况的动态特性，一次训练所得的关联规则不能始

终代表网络的当前状况，在本方法中将每条分类后的测试数据连同其网络连接类型加入到训练集中并再次训练，实时训练产生新规则并更新到关联规则库中。

[0038] 本发明具有以下优点：

[0039] 本发明首先为 10％ KDDCup99 的 41 个属性数据项添加位置参数，按照交叉检验的方法将其分成训练集和测试集。然后通过 Apriori-index 算法对训练集中的网络连接数据训练提取关联规则。最后通过 Apriori-index 算法中的测试方法对测试集中未知类型的网络连接数据分类，获得分类结果，并展示分类过程和分类结果，同时将测试集中的数据和对应的 分类结果导入到训练集中以实时更新训练集数据，产生新的关联规则，使本方法具有良好的自适应性和自学习特性。该发明采用 Apriori-index 算法，有效避免了经典 Apriori 算法处理 10％ KDDCup99 数据集时的大量重复计算，大大加快了分类的效率，提高了网络连接数据分类检测结果的准确度，降低了算法的时间复杂度，使本方法具有较强的稳定性。

附图说明

[0040] 图 1是本发明基于关联规则分类的网络入侵检测方法的流程图。

具体实施方式

[0041] 下面结合附图对本发明的具体实施方式作进一步的详细说明。

[0042] 将关联规则算法应用到入侵检测中，主要是一种以数据为中心的观点，对于网络连接数据的采集处理不在本发明的考虑范围之内。本发明中以国际标准网络连接数据集 10％ KDDCup99 为例，以数据挖掘的思想为理论依据对入侵网络连接进行分类。

[0043] 图 1对一种基于关联规则分类的网络入侵检测方法进行了详细的步骤说明。本发明提供的方法包括以下步骤：

[0044] 第 1 步、对国际标准数据集 10％ KDDCup99 预处理，将预处理后的数据集分成训练集和测试集两部分数据。

[0045] 第 1.1 步、为每列数据添加位置参数。因为 10％ KDDCup99 数据集中的数据中有大量相同的数据，比如：“0”和“1”。10％ KDDCup99 数据集中处于不同列的数据有不同的含义，而原始的 Apriori 算法在处理数据集中不同列的相同数据项时将他们视为同样的数据，因此直接使用原始的 Apriori 算法处理数据集会影响提取规则速度和分类结果的准确度。为避免出现以上问题，需要在数据预处理阶段为每个数据项添加位置参数 position，即数据项所在的列。这样数据集中的每个数据项可用结构体 Item{int positon,string data} 表示，例如位于第 2 列的 tcp 表示为 (2,tcp)。

[0046] 第 1.2 步、采用交叉验证的方法选取经过第 1.1 步预处理后的 10％ KDDCup99 数据集中 60％的连接数据作为训练集，40％的连接数据作为测试集。将 10％ KDDCup99 数据集中每 10 条网络连接数据归为一组，然后从每组中任意选取其中 6 条加入到训练集，剩余的 4 条数据加入到测试集。由于改进后的 Apriori 算法能够处理字符类型数据，同时数值类型的数据也可视为字符类型数据，所以无需对网络连接数据中的字符类型数据进行数值化和归一化处理。

[0047] 第 2 步、采用 Apriori-index 算法对选取的训练集中的网络连接数据进行训练，提取到关联规则，并将关联规则存放到关联规则库中。将关联规则库中的关联规则展示出来。

[0048] 首先进行以下定义和定理的说明：

[0049] • 项：设 $I = \{i_1, i_2, i_3, \ldots, i_m\}$ 是 m 个不同项的集合，每个 $i_k(k = 1,2,3,\ldots,m)$ 称为一个项 (Item)。

[0050] • 项集：项目的集合 I 称为项目集合 (Itemset)，简称为项集。

[0051] • k- 项集：项集中的每个项中包含有 k 个元素，k 也称为项集的长度。

[0052] • 候选项集 C_i：通过项集中各项的连接，用来获取频繁项集 L_i 的候选项目集合，下标 i 表示候选项集中每个候选项长度为 i。候选项集中满足支持度大于最小支持度阈值条件的候选项保留成为频繁项，不满足条件的将被删除。

[0053] • 频繁项集 L_i：候选项集 C_i 中支持度大于最小支持度 Min_Support 的候选项组成的集合称为频繁项集。

[0054] • 支持度 Support(Item)：指的是在训练集的所有网络连接数据中，包含项 Item 比例。例如：在 10 条训练集的网络连接数据中有 5 条包含项 (2,tcp)，则 Support((2,tcp)) = 0.5

[0055] • 置信度 Confidence(R)：对于规则 R:A → B，其置信度指的是在训练集中包含属性项 A 的网络连接数据中网络连接类型为 B 的比例。例如：在之前的 5 条包含项 (2,tcp) 的网络连接数据中连接类型为 Normal 的有 4 条，则 Confidence((2,tcp) → Normal) = 0.8。

[0056] 两个定理：

[0057] • 连接定理：若有两个 k-1- 项集，如果两个 k-1- 项集的前 k-2 个项相同，而最后一个项不同，则证明它们是可连接的，即这个 k-1- 项集可以连接生成 k- 项集。例如有两个 3- 项集：{a,b,c} {a,b,d}，这两个 3- 项集就是可连接的，它们可以连接生成 4- 项集 {a,b,c,d}。又如两个 3- 项集 {a,b,c}，{a,d,e}，这两个 3- 项集显示不能连接生成 4- 项集。

[0058] • 频繁子集定理：若一个项集的子集不是频繁项集，则该项集也不是频繁项集。例如，存在 4- 项集 {a,b,c,d}，如果它的 3- 项子集 {a,b,c} 的支持度小于最小支持度 Min_Support，则 4- 项集 {a,b,c,d} 的支持度也小于最小支持度 Min_Support。因此，若存在一个项集的子集不是频繁项集，该项集就不能被连接生成。

[0059] 第 2.1 步、初始化最小支持度阈值 Min_Support，最小置信度阈值 Min_Confidence。通过查阅文献资料和实验验证，最小支持度阈值和最小置信度阈值分别设定为 25% 和 78.5% 可以获得较高的分类准确度，初始化指定最小支持度阈值 Min_Support = 25%，最小置信度阈值 Min_Confidence = 78.5%。

[0060] 第 2.2 步、找出所有的频繁项集。遍历训练集中的所有的网络连接数据，统计每个属性值对应的连接类型及其出现的频度，形成候选项集合 C_1。在此基础上，根据支持度公式

[0061] $$Support(X) = \frac{Occur(X)}{Count(D)};$$

[0062] 计算支持度。其中 Occur(X) 表示训练集中所有网络连接数据中包含频繁项 {X} 的数量，Count(D) 表示训练集 {D} 中所有网络连接的数量。在候选选项集合 C_1 中删除支持度低于最小支持度阈值 Min_Support 的候选项，剩余的候选项形成频繁 1- 项目序列集合 L_1；然后对于每种网络连接类型，连接 L_1 中的不同元素构成候选项集合 C_2，再次遍历训练集数据，根据支持度公式计算 C_2 中的每个候选项的支持度，删除候选项集合 C_2 中支持度低于最

小支持度阈值 Min_Support 的候选项，剩余候选项的形成频繁 2- 项目序列集合 L_2；按照网络连接类型，再连接 L_2 中的不同元素构成候选项集合 C_3，再次遍历训练集数据，计算 C_3 中的每个候选项的支持度，删除支持度低于最小支持度阈值 Min_Support 的候选项，剩余的候选项形成频繁 3- 项目序列集合 L_3；重复进行以上的遍历和删除和连接的步骤，直到没有新的候选项产生，所有的频繁项集 $(L_1, L_2, \ldots, L_n)$ 都已搜寻得到。其中，连接步骤和删除步骤分别严格满足连接定理和频繁子集定理，即：若两个 (k-1)- 项集的前 (k-2) 个项相同，而最后一个项不同，则证明它们可连接得到 k- 项集；若 k- 项集任意一个子集不是频繁项集，则该 k- 项集也不是频繁项集。

[0063] 下面举一个例子对上述过程进行说明。从 10％ KDDCup99 随机选取如下数据：

[0064] 表 1 10％ KDDCup99 数据集中随机选取的 5 条数据

[0065]

0	tcp	http	SF	54540	8314	……	0.01	0.01	back
0	tcp	http	SF	54540	8314	……	0.04	0.04	back
2	tcp	time	SF	0	4	……	0.89	0	ipsweep
1	tcp	smtp	SF	1307	367	……	0	0	normal
3	tcp	smtp	SF	1187	329	……	0	0	normal

[0066] 最后一列代表网络连接类型，前面的数据列代表属性数据。通过添加列的位置参数，计算每个数据项的支持度，删除支持度小于最小支持度阈值 Min_Support 的候选项，得到频繁 1- 项目序列集合 L_1:{(1,0)},{(2,tcp)},{(3,http)},{(3,smtp)},{(4,SF)},{(5,54540)},{(6,8314)},{(40,0)},{(41,0)}；将频繁 1- 项目序列集合 L_1 中各项连接，遍历，删除得到频繁 2- 项目序列集合 L_2:{(1,0),(2,tcp)},{(1,0),(3,http)},{(1,0),(4,SF)},{(1,0),(5,54540)},{(1,0),(6,8314)},{(2,tcp),(3,http)},{(2,tcp),(3,smtp)},{(2,tcp),(4,SF)},{(2,tcp),(5,54540)},{(2,tcp),(6,8314)},{(2,tcp),(40,0)},{(2,tcp),(41,0)},{(3,http),(4,SF)},{(3,http),(5,54540)},{(3,http),(6,8314)},{(3,smtp),(4,SF)},{(3,smtp),(40,0)},{(3,smtp),(41,0)},{(4,SF),(5,54540)},{(4,SF),(6,8314)},{(4,SF),(40,0)},{(4,SF),(41,0)},{(5,54540),(6,8314)},{(40,0),(41,0)}；将频繁 2- 项目序列集合 L_2 中各项连接，遍历，删除得到频繁 3- 项目序列集合 L_3:{(1,0),(2,tcp),(3,http)},{(1,0),(2,tcp),(4,SF)},{(1,0),(2,tcp),(5,54540)},{(1,0),(2,tcp),(6,8314)},{(2,tcp),(3,http),(4,SF)},{(2,tcp),(3,http),(5,54540)},{(2,tcp),(3,http),(6,8314)},{(2,tcp),(3,smtp),(4,SF)},{(2,tcp),(3,smtp),(40,0)},{(2,tcp),(3,smtp),(41,0)},{(2,tcp),(4,SF),(5,54540)},{(2,tcp),(4,SF),(6,8314)},{(2,tcp),(5,54540),(6,8314)},{(2,tcp),(4,SF),(40,0)},{(2,tcp),(4,SF),(41,0)},{(2,tcp),(40,0),(41,0)}{(3,http),(4,SF),(5,54540)},{(3,http),(4,SF),(6,8314)},{(3,http),(5,54540),(6,8314)},{(3,smtp),(4,SF),(40,0)},{(3,smtp),(4,SF),(41,0)},{(3,smtp),-(40,0),(41,0)},{(4,SF),(5,54540),(6,8314)},{(4,SF),(40,0),(41,0)}；将频繁 3- 项目序列集合 L_3 中各项连接，遍历，删除得到频繁 4- 项目序列集 L_4:{(1,0),(2,tcp),(3,http),(4,SF)},{(1,0),(2,tcp),(3,http),(5,54540)},{(1,0),(2,tcp),(3,http),(6,8314)},{(2,tcp),(3,http),(4,SF),(5,54540)},{(2,tcp),(3,smtp),(4,SF),(6,8314)},{(3,http),(4,SF),(5,54540),(6,8314)},{(2,tcp),(3,smtp),(4,SF),(40,0)},{(2,tc

p),(3,smtp),(4,SF),(41,0)},{(2,tcp)(3,smtp)(40,0)(41,0)}{(2,tcp)(4,SF)(40,0)(41,0)},{(3,smtp),(4,SF),(40,0)(41,0)};将频繁4-项目序列集合L_4中各项连接,遍历,删除得到频繁5-项目序列集L_5:{(1,0),(2,tcp),(3,http),(4,SF),(5,54540)},{(1,0),(2,tcp),(3,http),(4,SF),(6,8314)},{(2,tcp),(3,ht tp),(4,SF),(5,54540),(6,8314)},{(2,tcp),(3,smtp),(4,SF),(40,0),(41,0)};将频繁5-项目序列集合L_5中各项连接,遍历,删除得到频繁6-项目序列集L_6:{(1,0),(2,tcp),(3,http),(4,SF),(5,54540),(6,8314)},至此所有的频繁项都已找到。

[0067] 第2.3步、由频繁项集产生关联规则。对于第2.2步中得到的频繁项集$(L_1,L_2,\ldots,L_n)$,假设频繁项集L_i中每个频繁项l_i的网络连接类型用t_j表示;如果$(l_i-t_j)\rightarrow t_j$的置信度大于最小置信度阈值Min_Confidence,则输出$(l_i-t_j)\rightarrow t_j$。置信度的计算根据置信度计算公式:

[0068] $$Confidence(A\rightarrow B)=P(B|A)=\frac{Support(A\cup B)}{Support(A)};$$

[0069] 其中Support(A ∪ B)和Support(A)分别表示频繁项{A ∪ B}和{A}的支持度;即置信度公式 可化为:

[0070] $$Confidence\{(l_i-t_j)\rightarrow t_j\}=\frac{Support(l_i)}{Support(l_i-t_j)};$$

[0071] 找到的所有满足要求的$(l_i-t_j)\rightarrow t_j$,即为关联规则;接着第2.2步中的例子对第2.3步由频繁项集产生关联规则进行说明:将第2.2步中得到的频繁项集中的每一项同其连接类型相接,组成$Item_i\rightarrow$ type这种形式的项-连接类型对,分别计算每个项-连接类型对的置信度,删除置信度小于最小置信度阈值的项-连接类型对,得到如下规则:{(1,0)}→back;{(2,tcp)}→back;{(2,tcp)}→ipsweep;{(2,tcp)}→normal;{(3,http)}→back;{(3,smtp)}→normal;{(4,SF)}→back;{(4,SF)}→ipsweep;{(4,SF)}→normal;{(5,54540)}→back;{(6,8314)}→back;{(40,0)}→normal;{(41,0)}→normal;{(41,0)}→ipsweep;{(1,0),(2,tcp)}→back;{(1,0),(3,http)}→back;{(1,0),(4,SF)}→back;{(1,0),(5,54540)}→back;{(1,0),(6,8314)}→back;{(2,tcp),(3,http)}→back;{(2,tcp),(3,smtp)}→normal;{(2,tcp),(4,SF)}→back;{(2,tcp),(4,SF)}→normal;{(2,tcp),(4,SF)}→ipsweep;{(2,tcp),(5,54540)}→back;{(2,tcp),(6,8314)}→back;{(2,tcp),(40,0)}→normal;{(2,tcp),(41,0)}→normal;{(2,tcp),(41,0)}→ipsweep;{(3,http),(4,SF)}→back;{(3,http),(5,54540)}→back;{(3,http),(6,8314)}→back;{(3,smtp),(4,SF)}→normal;{(3,smtp),(40,0)}→normal;{(3,smtp),(41,0)}→normal;{(4,SF),(5,54540)}→back;{(4,SF),(6,8314)}→back;{(4,SF),(40,0)}→normal;{(4,SF),(41,0)}→normal;{(4,SF),(41,0)}→ipsweep;{(5,54540),(6,8314)}→back;{(40,0),(41,0)}→normal;{(1,0),(2,tcp),(3,http)}→back;{(1,0),(2,tcp),(4,SF)}→back;{(1,0),(2,tcp),(5,54540)}→back;{(1,0),(2,tcp),(6,8314)}→back;

{(2,tcp),(3,http),(4,SF)} → back ;{(2,tcp),(3,http),(5,54540)} → back ;{(2,tcp),(3,http),(6,8314)} → back ;{(2,tcp),(3,smtp),(4,SF)} → normal ;{(2,tcp),(3,smtp),(40,0)} → normal ;{(2,tcp),(4,SF),(6,8314)} → back ;{(2,tcp),(3,smtp),(41,0)} → normal ;{(2,tcp),(4,SF),(5,54540)} → back ;{(2,tcp),(5,54540),(6,8314)} → back ;{(2,tcp),(4,SF),(40,0)} → normal ;{(2,tcp),(4,SF),(41,0)} → normal ;{(2,tcp),(4,SF),(41,0)} → ipsweep ; {(2,tcp),(40,0),(41,0)} → normal ;{(3,http),(4,SF),(5,54540)} → back ;{(3,http),(4,SF),(6,8314)} → back ;{(3,http),(5,54540),(6,8314)} → back ;{(3,smtp),(4,SF),(40,0)} → normal ;{(3,smtp),(4,SF),(41,0)} → normal ;{(3,smtp),(40,0),(41,0)} → normal ;{(4,SF),(5,54540),(6,8314)} → normal ;{(4,SF),(40,0),(41,0)} → normal ;{(1,0),(2,tcp),(3,http),(4,SF)} → back ;{(1,0),(2,tcp),(3,http),(5,54540)} → back ;{(1,0),(2,tcp),(3,http),(6,8314)} → back ;{(2,tcp),(3,http),(4,SF),(5,54540)} → back ;{(2,tcp),(3,smtp),(4,SF),(6,8314)} → back ;{(3,http),(4,SF),(5,54540),(6,8314)} → back ;{(2,tcp),(3,smtp),(4,SF),(40,0)} → normal ;{(2,tcp),(3,smtp),(4,SF),(41,0)} → normal ;{(2,tcp)(3,smtp)(40,0)(41,0)} → normal ;{(2,tcp)(4,SF)(40,0)(41,0)} → normal ;{(2,tcp)(3,smtp)(40,0)(41,0)} → normal{(3,smtp),(4,SF),(40,0)(41,0)} → normal ;{(1,0),(2,tcp),(3,http),(4,SF),(5,54540)} → back ;{(1,0),(2,tcp),(3,http),(4,SF),(6,8314)} → back ;{(2,tcp),(3,http),(4,SF),(5,54540),(6,8314)} → back ;{(2,tcp),(3,smtp),(4,SF),(40,0),(41,0)} → normal ;{(1,0),(2,tcp),(3,http),(4,SF),(5,54540),(6,8314)} → back

[0072] 第2.4步、将第2.3步中得到的关联规则添加到关联规则库中，作为对测试集中未知类型的网络连接数据测试分类的判断依据。

[0073] 第2.5步、将关联规则库中的规则展示出来。10％ KDDCup99数据量较大，经过Apriori-index算法训练得到的关联规则非常多，关联规则在页面内显示会比较混乱，所以关联规则展示页面进行适当的缩放，使关联规则清晰展示。

[0074] 第3步、测试集中的每条网络连接数据逐条匹配关联规则库中规则，根据不同规则的形式计算分类结果的权值并找出最大权值所对应的结果即为最终分类结果。

[0075] 第3.1步、读取测试集数据，对测试集中的每条网络连接数据按照关联规则分类，统计分类结果。10％ KDDCup99数据集中每条网络连接数据有41个属性数据项和1个连接类型数据项，第2步中提取到的关联规则的条件部分包含有多个属性数据项，测试集中的每条未知类型的网络连接数据按照提取的规则分类时，会有多条关联规则与之对应，所以按关联规则分类需经过以下过程：

[0076] 第3.1.1步、对读取到的测试集中的一条连接数据，遍历整个关联规则库，统计并记录匹配的关联规则 R_i（例如：$(1_i\text{-}t_j) \rightarrow t_j$）中条件部分 $(1_i\text{-}t_j)$ 的长度，即统计关联规则 R_i 条件部分 $(1_i\text{-}t_j)$ 包含的属性数据项的数量 $Length_i$；

[0077] 第3.1.2步、分别计算匹配的n条规则中对应的网络连接类型部分 t_j 的权值。按照Apriori-index算法权值计算公式：

[0078] $$Weight(t_j)=\sum_{i=1}^{n}(Length_i*\log_2 Length_i+Length_i)$$;

[0079] 计算该条未知网络连接类型的测试数据在经过关联规则库中所有关联规则比对后匹配的第 j 种网络连接类型 t_j的权值。这主要是由于关联规则条件部分 (1_i-t_j) 的长度 $Length_i$越大，分类准确度越高，这样做能够同其他关联规则加以区分，提高分类结果的准确度。

[0080] 第 3.2 步、输出分类结果：网络连接类型 t。比较所有的权值，从中找出权值最高的 Weight(t)，将分类结果（即网络连接类型 t）输出。

[0081] 为了展示第 3 步过程，从 10% KDDCup99 数据集中连接类型为 back, ipsweep, normal 的网络连接数据中随机选取一条如下：

[0082] 表 2 10% KDDCup99 数据集中连接类型为以上三种的数据中随机挑选的一条数据

[0083]

1	tcp	smtp	SF	835	377	……	0	0	normal

[0084] 将这条数据与得到的关联规则进行匹配，关联规则库中匹配这条连接的规则有：{(2,tcp)} → back；{(4,SF)} → back；{(2,tcp),(4,SF)} → back；{(2,tcp)} → ipsweep；{(4,SF)} → ipsweep；{(41,0)} → ipsweep；{(2,tcp),(4,SF)} → ipsweep；{(2,tcp),(41,0)} → ipsweep；{(4,SF),(41,0)} → ipsweep；{(2,tcp),(4,SF),(41,0)} → ipsweep；{(2,tcp)} → normal；{(3,smtp)} → normal；{(4,SF)} → normal；{(40,0)} → normal；{(41,0)} → normal；{(2,tcp),(3,smtp)} → normal；{(2,tcp),(4,SF)} → normal；{(2,tcp),(40,0)} → normal；{(2,tcp),(41,0)} → normal；{(3,smtp),(4,SF)} → normal；{(3,smtp),(40,0)} → normal；{(3,smtp),(41,0)} → normal；{(4,SF),(40,0)} → normal；{(4,SF),(41,0)} → normal；{(40,0),(41,0)} → normal；{(2,tcp),(3,smtp),(4,SF)} → normal；{(2,tcp),(3,smtp),(40,0)} → normal；{(2,tcp),(3,smtp),(41,0)} → normal；{(2,tcp),(4,SF),(40,0)} → normal；{(2,tcp),(4,SF),(41,0)} → normal；{(2,tcp),(40,0),(41,0)} → normal；{(3,smtp),(4,SF),(40,0)} → normal；{(3,smtp),(4,SF),(41,0)} → normal；{(3,smtp),(40,0),(41,0)} → normal；{(4,SF),(40,0),(41,0)} → normal；{(2,tcp),(3,smtp),(4,SF),(40,0)} → normal；{(2,tcp),(3,smtp),(4,SF),(41,0)} → normal；{(2,tcp)(3,smtp)(40,0)(41,0)} → normal；{(2,tcp)(4,SF)(40,0)(41,0)} → normal；{(3,smtp),(4,SF),(40,0)(41,0)} → normal；{(2,tcp),(3,smtp),(4,SF),(40,0),(41,0)} → normal；

[0085] 分别计算这些规则对应的三种连接类型的权值：

[0086] $Weight(back)=2*(1*\log_2 1+1)+(2*\log_2 2+2)=6$；

[0087] $Weight(ipsweep)=3*(1*\log_2 1+1)+3*(2*\log_2 2+2)+(3*\log_2 3+3)=22.7549$；

[0088] $Weight(normal)=5*(1*\log_2 1+1)+10*(2*\log_2 2+2)+10*(3*\log_2 3+3)$

[0089] $+5*(4*\log_2 4+4)+(\log_2 5+5)=199.1585$；

[0090]

[0091] 权值最高的是 Weight(normal)，则这条网络连接数据根据关联规则分类得到的结果是 normal，这条数据的最后一列也证明了分类的正确性。

[0092] 第 4 步、保存第 3 步中分类结果，将分类过程和分类结果展示出来，同时为保证该方法良好的自适应性和自学习特性，测试集的数据根据关联规则得到分类结果后，训练集数据连同对应的分类结果重新加入到训练集数据中，为后续关联规则提取提供训练数据源，保证关联规则的实时更新。

[0093] 第 4.1 步、测试数据展示。为将每条测试数据从读取直至分类完成过程展示出来，将每条测试数据用运动的图形代表，图形运动的轨迹和颜色变化代表测试数据的分类过程。每条网络连接对应一个动态 Ellipse 模型，而每个 Ellipse 动画模型对应一个测试线程。每读取一条测试数据，该方法通过规则匹配获取对该连接的分类结果。展示界面中的每个动态 Ellipse 模型，该方法利用其颜色、位置和运动轨迹的变化直观地显示具体分类过程和分类结果。

[0094] 第 4.2 步、将测试过的网络连接数据与对应的分类结果添加到训练集中，保证方法良好的自适应性和字学习特性。考虑到实际网络状况的动态的属性，一次训练所得规则不能一直代表网络当前的网络状况，该方法将每条分类后的测试数据连同其分类结果添加到训练集中并再次训练，实时训练产生的新规则并更新到关联规则库中。

[0095] 为了验证 Apriori-index 算法相比原始的 Apriori 算法应用于网络入侵检测系统的优越性，我们进行以下对比验证实验。实验环境：一台 PC 机。CPU 型号为 InterCorei7-4770 3.4GHz，内存 8G，1T 硬盘，具备 Visual Studio 2013 的软件环境。实验数据：按照 10％ KDDCup99 数据集中网络连接类型的不同比例，从中随机选取，保证每种连接类型的所取数据量最多不超过 4000 条，共选取 36854 条，然后使用交叉检验的方法，选取其中的 60％作为训练集数据，另外 40％作为测试集数据。利用改进前后的 Apriroi 算法进行 5 次实验。实验结果如表 3所示：

[0096] 表 3 利用国际标准数据集 10％ KDDCup99 对改进前后 Apriori 算法验证结果对比

[0097]

	原始Apriori算法		Apripori-index算法	
组号	执行时间	准确度	执行时间	准确度
第1组	76小时35分	89.64%	19.12秒	98.46%
第2组	76小时28分	89.72%	19.10秒	98.84%
第3组	76小时30分	90.12%	19.05秒	99.06%
第4组	76小时45分	90.04%	19.02秒	98.92%
第5组	76小时36分	90.18%	19.07秒	99.43%

[0098] 实验结果表明：本发明的入侵检测方法相比原始的 Apriori 算法在执行结果的准确度上有了很大的提升，在执行时间方面有了极大的改善。

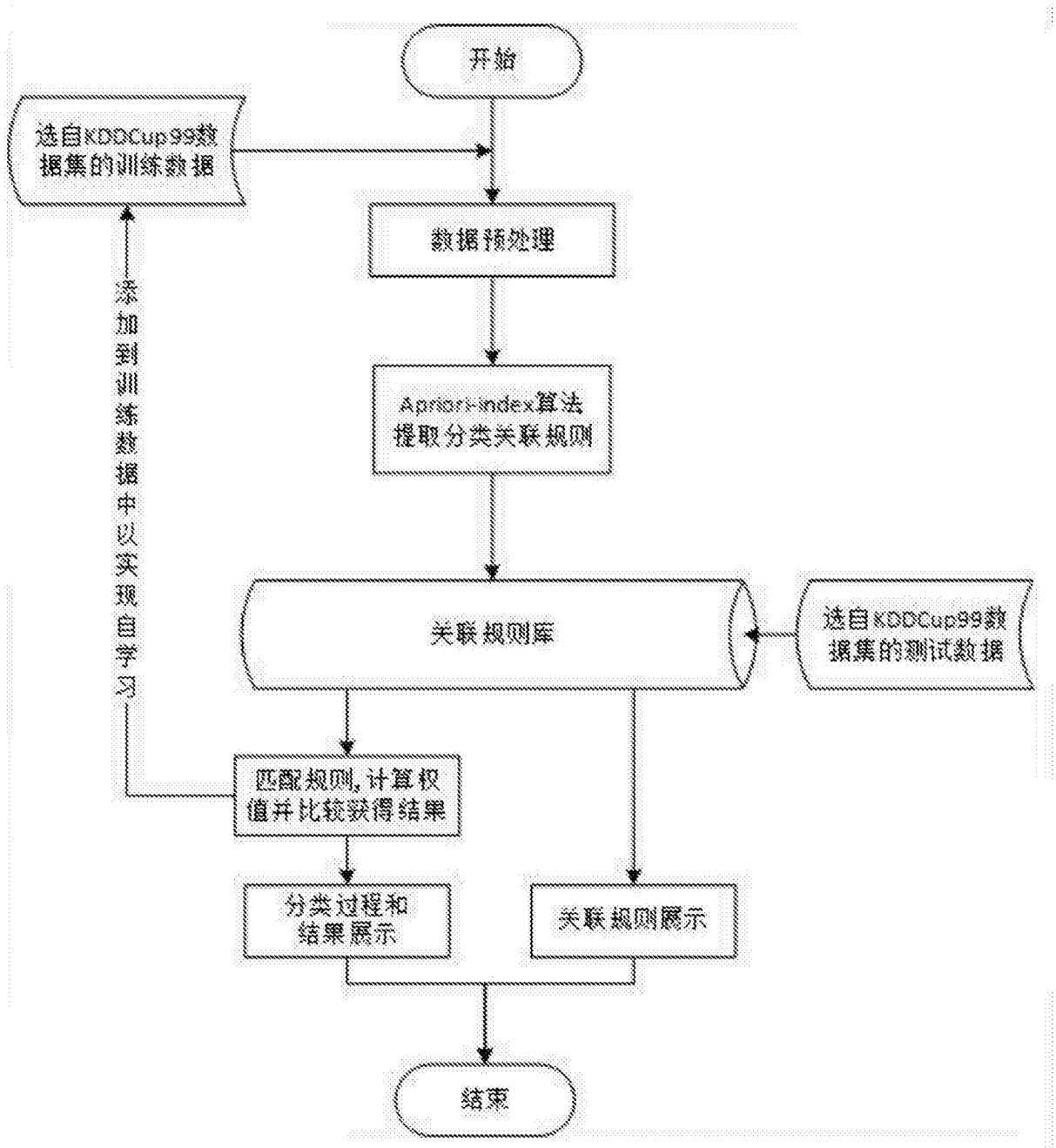

开始
选自KDDCup99数据集的训练数据
数据预处理
Apriori-index算法提取分类关联规则
关联规则库
选自KDDCup99数据集的测试数据
匹配规则,计算权值并比较获得结果
添加到训练数据中以实现自学习
分类过程和结果展示
关联规则展示
结束

图 1