(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 108156131 A
(43)申请公布日 2018.06.12

(21)申请号 201711021568.1

(22)申请日 2017.10.27

(71)申请人 上海观安信息技术股份有限公司
地址 200333 上海市浦东新区泥城镇云端路1412弄15号二层1室

(72)发明人 夏玉明 王小东 陈曦 辜乘风

(74)专利代理机构 北京新知远方知识产权代理事务所(普通合伙) 11397
代理人 刘玲 艾凤英

(51)Int.Cl.
H04L 29/06(2006.01)
H04L 29/08(2006.01)
G06F 17/30(2006.01)

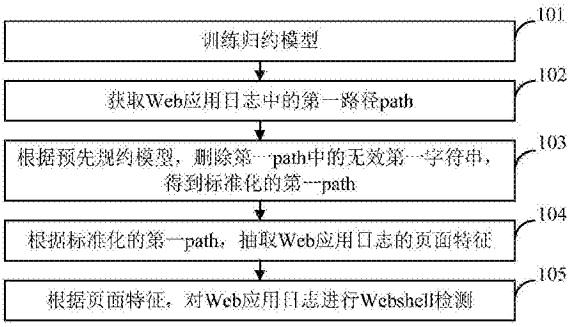
权利要求书2页 说明书15页 附图3页

(54)发明名称

Webshell检测方法、电子设备和计算机存储介质

(57)摘要

本申请提供了一种Webshell检测方法、电子设备和计算机存储介质，属于计算机信息安全技术领域。所述方法包括：根据规约模型，删除path中的无效字符串，得到标准化的path；根据标准化的path，抽取Web应用日志的页面特征，进而对Web应用日志进行Webshell检测。本申请根据规约模型删除path中的无效字符串，得到标准化的path；根据标准化的path，抽取页面特征，进而进行Webshell检测，实现了根据规约模型选择标准字符串，将涉及同一页面的path标准化为相同的path，进而基于path的不同抽取页面特征，进行Webshell检测，不仅具有通用性，而且降低误报率和漏报率。



CN 108156131 A

1.一种Webshell检测方法,其特征在于,所述方法包括:

获取Web应用日志中的第一路径path,所述第一path由至少一个第一字符串组成;

根据预先由字符串聚类得到的规约模型,删除所述第一path中的无效第一字符串,得到标准化的第一path;

根据所述标准化的第一path,抽取所述Web应用日志的页面特征;

根据所述页面特征,对所述Web应用日志进行Webshell检测。

2.根据权利要求1所述的方法,其特征在于,所述根据预先训练的规约模型删除所述第一path中的无效字符串,得到标准化的第一path之前,还包括:

获取多个第二path,所述第二path由至少一个第二字符串组成;

获取各第二path中的第二字符串;

将各第二字符串的各字符转换成ASCII值,形成各第二字符串对应的数组;

确定各数组的长度、平均值和标准差;

根据各数组的长度、平均值和标准差对所有第二path的所有第二字符串进行聚类;

根据预先设定的分类规则,确认聚类结果中的有效聚类;

将有效聚类涉及的第二字符串形成白名单,并将所述白名单确定为训练的规约模型。

3.根据权利要求2所述的方法,其特征在于,所述分类规则包括至少一个第三字符串,所述第三字符串为有效聚类的示例字符串;

所述根据预先设定的分类规则,确认聚类结果中的有效聚类,包括:

计算所述第三字符串的中心点;

计算聚类结果中各聚类中心点距所述第三字符串的中心点的第一距离;

将所述第一距离小于预设第一阈值的聚类确定为有效聚类。

4.根据权利要求2所述的方法,其特征在于,所述分类规则包括至少一个第四字符串,所述第四字符串为无效聚类的示例字符串;

所述根据预先设定的分类规则,确认聚类结果中的有效聚类,包括:

计算所述第四字符串的中心点;

计算聚类结果中各聚类中心点距所述第四字符串的中心点的第二距离;

将所述第二距离大于预设第二阈值的聚类确定为有效聚类。

5.根据权利要求3或4所述的方法,其特征在于,所述根据所述标准化的第一path,抽取所述Web应用日志的页面特征之前,还包括:

选择请求的响应状态为200和500的Web应用日志;

所述根据所述标准化的第一path,抽取所述Web应用日志的页面特征,包括:

根据所述标准化的第一path,抽取选择的Web应用日志的页面特征。

6.根据权利要求5所述的方法,其特征在于,所述根据所述标准化的第一path,抽取选择的Web应用日志的页面特征,包括:

按标准化的第一path的不同,将选择的Web应用日志分类,其中,每类中各选择的Web应用日志对应相同的标准化的第一path;

基于每类中选择的Web应用日志分类,抽取每类的页面特征。

7.根据权利要求6所述的方法,其特征在于,所述页面特征包括:入度、出度、初次访问时间、最新访问时间、页面曝光天数、页面出现过天数、页面get请求次数、页面post请求次

数、页面500次数、页面响应返回最大值、页面响应返回最小值、页面响应返回平均值、来访网络之间互连的协议IP数量、来访IP的C段数量、来访用户代理UA数量、统一资源标识符URI访问有参数访问次数、URI访问无参数访问次数、页面访问量曝光天数比、页面访问量出现过天数比、页面出现不同的时间、访问量与页面出现不同的时间的商、Refer总数、Refer等于path数量、Refer等于path占比、页面对应Session标识ID数量、页面访问量、页面500次数占比、页面200次数、POST请求占比、页面响应值不同数、页面响应值不同占比、UA异常数、Session ID异常数、Session ID异常占比、页面响应最大值与页面响应平均值的商。

8.根据权利要求7所述的方法，其特征在于，所述根据所述页面特征，对所述Web应用日志进行Webshell检测，包括：

分别利用孤立森林模型、高斯混合模型、局部异常因子检测模型判断所述页面特征是否异常；

若孤立森林模型、高斯混合模型、局部异常因子检测模型均判断所述页面特征异常，则确定所述Web应用日志为Webshell访问异常数据；

若孤立森林模型、高斯混合模型、局部异常因子检测模型均判断所述页面特征正常，则确定所述Web应用日志为Webshell访问正常数据；

若孤立森林模型、高斯混合模型、局部异常因子检测模型中2个模型判断所述页面特征异常，1个模型判断所述页面特征正常，则确定所述Web应用日志为Webshell访问较异常数据；

若孤立森林模型、高斯混合模型、局部异常因子检测模型中2个模型判断所述页面特征正常，1个模型判断所述页面特征异常，则确定所述Web应用日志为Webshell访问一般异常数据。

9.一种电子设备，其特征在于，包括存储器、处理器、总线以及存储在存储器上并可在处理器上运行的计算机程序，所述处理器执行所述程序时实现如权利要求1-8任意一项的步骤。

10.一种计算机存储介质，其上存储有计算机程序，其特征在于：所述程序被处理器执行时实现如权利要求1-9任意一项的步骤。

Webshell检测方法、电子设备和计算机存储介质

技术领域

[0001] 本申请涉及计算机信息安全技术领域，尤其涉及Webshell检测方法、电子设备和计算机存储介质。

背景技术

[0002] Webshell是一种网站应用后门程序，是网站应用比较严重的安全威胁，黑客可以利用Webshell长期获得网站应用的访问权限，并且可以利用存在Webshell的应用服务器作为跳板机进一步渗透内网。因此，防护Webshell攻击对于网站应用安全以及内网安全十分重要。

[0003] Webshell传统检测方法主要分为两种：

[0004] 1.静态检测：静态特征检测是指针对Webshell中可能使用的关键词、高危函数、文件修改的时间、文件之间的关联关系等静态属性维度进行规则匹配。检测过程需要建立一个包含各种规则的Webshell特征库，如：“关键字：大马|小马|木马|webshell”，“高危函数：eval()、excute()、run()、exec()、phpinfo()”，“WEB文件修改时间”，“文件与文件之间的关联关系”等特征。通过将文件自身的关键字、高危函数、修改时间、文件与文件之间是否有关联等特征与webshell特征库中的规则进行匹配，若命中特征则视为该文件为Webshell。

[0005] 2.动态检测：黑客为了避免Webshell被静态检测方法直接识别到，开发了采用加密方式绕过部分静态特征检测的变种Webshell。为了弥补静态检测方法对此类Webshell检测能力的不足，便有了Webshell动态检测方法。动态检测首先提取镜像流量，然后解析出流量中的Web应用脚本文件(jsp、php、asp、cgi、jspx、aspx等)，最后检测Web应用请求访问的Web应用脚本文件是否采用了加密混淆技术、Web应用请求是否命中Webshell请求特征库等规则，从而判断该Web脚本文件是否为Webshell。

[0006] 现有的静态检测、动态检测对Webshell检测都具有一定的效果，但由于技术原因又存在了诸多不足之处。例如：静态检测过于依赖规则，容易出现误报的情况，且无法检测出加密或伪装过的Webshell，会出现漏报的情况。动态检测也过多的依赖于规则，存在误报率高的问题，同时动态检测时需要获取网络镜像流量及解析出流量中的Web请求，这就需要服务器具有较高的运算性能，同时Web请求解析也需要准确，所以动态检测虽然相对于静态检测在漏报率上效果会好一些，但检测成本会成倍提高。

发明内容

[0007] 为解决上述问题，本申请实施例提出了一种Webshell检测方法、电子设备和计算机存储介质。

[0008] 第一方面，本申请实施例提供了Webshell检测方法，所述方法包括：

[0009] 获取Web应用日志中的第一路径path，所述第一path由至少一个第一字符串组成；

[0010] 根据预先由字符串聚类得到的规约模型，删除所述第一path中的无效第一字符

串，得到标准化的第一path；

[0011] 根据所述标准化的第一path，抽取所述Web应用日志的页面特征；

[0012] 根据所述页面特征，对所述Web应用日志进行Webshell检测。

[0013] 可选地，所述根据预先训练的规约模型删除所述第一path中的无效字符串，得到标准化的第一path之前，还包括：

[0014] 获取多个第二path，所述第二path由至少一个第二字符串组成；

[0015] 获取各第二path中的第二字符串；

[0016] 将各第二字符串的各字符转换成ASCII值，形成各第二字符串对应的数组；

[0017] 确定各数组的长度、平均值和标准差；

[0018] 根据各数组的长度、平均值和标准差对所有第二path的所有第二字符串进行聚类；

[0019] 根据预先设定的分类规则，确认聚类结果中的有效聚类；

[0020] 将有效聚类涉及的第二字符串形成白名单，并将所述白名单确定为训练的规约模型。

[0021] 可选地，所述分类规则包括至少一个第三字符串，所述第三字符串为有效聚类的示例字符串；

[0022] 所述根据预先设定的分类规则，确认聚类结果中的有效聚类，包括：

[0023] 计算所述第三字符串的中心点；

[0024] 计算聚类结果中各聚类中心点距所述第三字符串的中心点的第一距离；

[0025] 将所述第一距离小于预设第一阈值的聚类确定为有效聚类。

[0026] 可选地，所述分类规则包括至少一个第四字符串，所述第四字符串为无效聚类的示例字符串；

[0027] 所述根据预先设定的分类规则，确认聚类结果中的有效聚类，包括：

[0028] 计算所述第四字符串的中心点；

[0029] 计算聚类结果中各聚类中心点距所述第四字符串的中心点的第二距离；

[0030] 将所述第二距离大于预设第二阈值的聚类确定为有效聚类。

[0031] 可选地，所述根据所述标准化的第一path，抽取所述Web应用日志的页面特征之前，还包括：

[0032] 选择请求的响应状态为200和500的Web应用日志；

[0033] 所述根据所述标准化的第一path，抽取所述Web应用日志的页面特征，包括：

[0034] 根据所述标准化的第一path，抽取选择的Web应用日志的页面特征。

[0035] 可选地，所述根据所述标准化的第一path，抽取选择的Web应用日志的页面特征，包括：

[0036] 按标准化的第一path的不同，将选择的Web应用日志分类，其中，每类中各选择的Web应用日志对应相同的标准化的第一path；

[0037] 基于每类中选择的Web应用日志分类，抽取每类的页面特征。

[0038] 可选地，所述页面特征包括：入度、出度、初次访问时间、最新访问时间、页面曝光天数、页面出现过天数、页面get请求次数、页面post请求次数、页面500次数、页面响应返回最大值、页面响应返回最小值、页面响应返回平均值、来访网络之间互连的协议IP数量、来

访IP的C段数量、来访用户代理UA数量、统一资源标识符URI访问有参数访问次数、URI访问无参数访问次数、页面访问量曝光天数比、页面访问量出现过天数比、页面出现不同的时间、访问量与页面出现不同的时间的商、Refer总数、Refer等于path数量、Refer等于path占比、页面对应Session标识ID数量、页面访问量、页面500次数占比、页面200次数、POST请求占比、页面响应值不同数、页面响应值不同占比、UA异常数、Session ID异常数、Session ID异常占比、页面响应最大值与页面响应平均值的商。

[0039] 可选地，所述根据所述页面特征，对所述Web应用日志进行Webshell检测，包括：

[0040] 分别利用孤立森林模型、高斯混合模型、局部异常因子检测模型判断所述页面特征是否异常；

[0041] 若孤立森林模型、高斯混合模型、局部异常因子检测模型均判断所述页面特征异常，则确定所述Web应用日志为Webshell访问异常数据；

[0042] 若孤立森林模型、高斯混合模型、局部异常因子检测模型均判断所述页面特征正常，则确定所述Web应用日志为Webshell访问正常数据；

[0043] 若孤立森林模型、高斯混合模型、局部异常因子检测模型中2个模型判断所述页面特征异常，1个模型判断所述页面特征正常，则确定所述Web应用日志为Webshell访问较异常数据；

[0044] 若孤立森林模型、高斯混合模型、局部异常因子检测模型中2个模型判断所述页面特征正常，1个模型判断所述页面特征异常，则确定所述Web应用日志为Webshell访问一般异常数据。

[0045] 第二方面，本申请实施例提供了一种电子设备，包括存储器、处理器、总线以及存储在存储器上并可在处理器上运行的计算机程序，所述处理器执行所述程序时实现如上述第一方面的步骤。

[0046] 第三方面，本申请实施例提供了一种计算机存储介质，其上存储有计算机程序所述程序被处理器执行时实现如上述第一方面的步骤。

[0047] 有益效果如下：

[0048] 根据预先由字符串聚类得到的规约模型，删除path中的无效字符串，得到标准化的path；根据标准化的path，抽取Web应用日志的页面特征；根据页面特征，对Web应用日志进行Webshell检测，实现了根据规约模型选择标准字符串，进而将涉及同一页面的Web应用日志中的path标准化为相同的path，进而基于path的不同抽取页面特征，进行Webshell检测，不仅具有通用性，降低检测成本，而且降低误报率和漏报率。

附图说明

[0049] 下面将参照附图描述本申请的具体实施例，其中：

[0050] 图1示出了本申请一实施例提供的一种Webshell检测方法的流程示意图；

[0051] 图2示出了本申请一实施例提供的字符串聚类示意图；

[0052] 图3示出了本申请一实施例提供的字符串长度、数组均值和数组标准差分类示意图；

[0053] 图4示出了本申请一实施例提供的kmeans和GMM示意图；

[0054] 图5示出了本申请一实施例提供的Isolation Forest结果示意图；

[0055] 图6示出了本申请一实施例提供的一种电子设备的结构示意图。

具体实施方式

[0056] 为了使本申请的技术方案及优点更加清楚明白，以下结合附图对本申请的示例性实施例进行进一步详细的说明，显然，所描述的实施例仅是本申请的一部分实施例，而不是所有实施例的穷举。并且在不冲突的情况下，本说明中的实施例及实施例中的特征可以互相结合。

[0057] 目前常用的Webshell检测方法主要分为两种：

[0058] 1.静态检测：针对Webshell中可能使用的关键词、高危函数、文件修改的时间、文件之间的关联关系等静态属性维度进行规则匹配。检测过程需要建立一个包含各种规则的Webshell特征库，通过将文件自身的关键字、高危函数、修改时间、文件与文件之间是否有关联等特征与webshell特征库中的规则进行匹配，若命中特征则视为该文件为Webshell。

[0059] 2.动态检测：首先提取镜像流量，然后解析出流量中的Web应用脚本文件，最后检测Web应用请求访问的Web应用脚本文件是否采用了加密混淆技术、Web应用请求是否命中Webshell请求特征库等规则，从而判断该Web脚本文件是否为Webshell。

[0060] 无论何种检测方法，都无法做到即降低检测成本又降低漏报率。

[0061] 本申请提出了一种Webshell检测方法、电子设备和计算机存储介质，根据预先由字符串聚类得到的规约模型，删除path中的无效字符串，得到标准化的path；根据标准化的path，抽取Web应用日志的页面特征；根据页面特征，对Web应用日志进行Webshell检测，实现了根据规约模型选择标准字符串，进而将涉及同一页面的Web应用日志中的path标准化为相同的path，进而基于path的不同抽取页面特征，进行Webshell检测，不仅具有通用性，降低检测成本，而且降低误报率和漏报率。

[0062] 结合上述实施环境，参见图1所示的实施例，本实施例提供了一种Webshell检测方法，本实施例提供的方法流程具体如下：

[0063] 101，训练归约模型。

[0064] 本步骤中的归约模型是由字符串聚类得到的。

[0065] 因为Web应用可能使用不同的框架(J2EE、.NET、PHP)开发的，所以URI(Uniform Resource Identifier，统一资源标识符)存在哈希hash、随机数等情况，这样会导致生成的目录树相对庞大，通过归约模型对路径path做统一规约，可以有效合并Web应用中的相似的URI，使得最终生成的URI目录树简洁、合理。

[0066] 因此，本提案在进行正式path规约前，需要执行本步骤训练归约模型。

[0067] 归约模型训练方案为：对path路径采用正则表达式字符串匹配的方式进行处理。采用这种方式处理path需要要求Web应用在开发时约定好随机数的格式及出现位置。这种处理方式对于不同的Web应用不具备通用性，后期维护灵活性也相对较差。

[0068] 为了更加提升本提案的效果，本申请提供一种更优的训练方案。在处理path规约时采用k-means方式来实现path聚类，对request字段及http_referer字段中的path进行数据标准化规约。

[0069] 该更优的训练流程如下：

[0070] 1，获取多个第二path。

[0071] 其中，第二path由至少一个第二字符串组成。

[0072] 如：第二path为：/test/login/da853b0d3f88d99b，或者，第二path为：/test/login/89c86e2691cfc444。

[0073] 2，获取各第二path中的第二字符串。

[0074] 在实际应用时，可以形成一个包括所有第二path的所有第二字符串的字符集合。

[0075] 如：第二path：/test/login/da853b0d3f88d99b中的第二字符串：test，login和da853b0d3f88d99b。

[0076] 3，将各第二字符串的各字符转换成ASCII值，形成各第二字符串对应的数组。

[0077] 如：第二字符串test可以转换成[116,101,115,116]。

[0078] 在具体实施时，还可以形成一个包括所有第二字符串对应的数组的数组集合。

[0079] 另外，还可以根据不同公司的path特点，对生成的ASCII值进行调整。

[0080] 4，确定各数组的长度、平均值和标准差。

[0081] 如：确定test的数组[116,101,115,116]的长度为4，平均值为112.0，标准差为6.364，可表示为(4,112.0,6.364)。

[0082] 5，根据各数组的长度、平均值和标准差对所有第二path的所有第二字符串进行聚类。

[0083] 本实施例不对聚类方法进行限定。下面以k-means聚类方法为例，其聚类过程如图2所述。

[0084] 其中，图2(a)为需要聚类的数据，图2(b)“x”点为初始化中心点，2(c)为初始点的邻近点，图2(d)“x”点为c中邻近点计算的中心点，通过不断计算邻近点和中心点的方式最终模型收敛如图2(f)所示。

[0085] 根据字符串长度、数组均值和数组标准差进行聚类如图3所示。

[0086] 经过此步骤，会将第二字符串聚类，如test，login为一类，da853b0d3f88d99b，89c86e2691cfc444为一类。

[0087] 6，根据预先设定的分类规则，确认聚类结果中的有效聚类。

[0088] 聚类的时候是聚成多类，会有浮点数集、字母集、带参数字符串集和中文集等，需要加入分类规则的处理才能确定最终的聚类结果。

[0089] 本步骤的执行主体可以为人，也可以为本方法所在的设备。

[0090] 1)用户观察聚类结果，根据每类中包括的第二字符串的内容，根据该内容确定分类规则，并基于该规则确定得到的各个聚类是否有效。

[0091] 2)本方法所在的设备读取预先设定的分类规则，确定每类中包括的第二字符串的内容是否满足该规则，根据满足规则的聚类确定为有效。

[0092] 对于2)中的分类规则，可以包括至少一个第三字符串，第三字符串为有效聚类的示例字符串。

[0093] 本方法所在的设备可以读取分类规则，计算第三字符串的中心点；计算聚类结果中各聚类中心点距第三字符串的中心点的第一距离；将第一距离小于预设第一阈值的聚类确定为有效聚类。

[0094] 对于2)中的分类规则，可以包括至少一个第四字符串，第四字符串为无效聚类的示例字符串。

[0095] 本方法所在的设备可以读取分类规则，计算第四字符串的中心点；计算聚类结果中各聚类中心点距第四字符串的中心点的第二距离；将第二距离大于预设第二阈值的聚类确定为有效聚类。

[0096] 除此之外，对于2)中的分类规则，可以即包括至少一个第三字符串，也至少一个第四字符串。

[0097] 本方法所在的设备可以读取分类规则，同时计算第三字符串的中心点和第四字符串的中心点；计算聚类结果中各聚类中心点距第三字符串的中心点的第一距离，并计算聚类结果中各聚类中心点距第四字符串的中心点的第二距离；将第一距离小于预设第三阈值且第二距离大于预设第四阈值的聚类确定为有效聚类。

[0098] 例如，分类规则为随机数所在类无效，本方法所在的设备读取该分类规则，确定每类中包括的第二字符串的内容，对于包括随机数的聚类，确定其无效，对于未包括随机数的聚类，确定其有效，即确定test，login所在类有效，da853b0d3f88d99b，89c86e2691cfc444所在类无效。

[0099] 7，将有效聚类涉及的第二字符串形成白名单，并将白名单确定为训练的规约模型。

[0100] 经过6得到由第二字符串组成的类，将有效类涉及的第二字符串形成白名单。如白名单为test，login。

[0101] 至此，完成归约模型的训练。

[0102] 需要说明的是，步骤101可以每次均执行，也可以非均执行，即仅当path发生变化，或者，其他需要重新执行的情况发生时，才执行。本实施例不对步骤101的执行方式进行限定。

[0103] 102，获取Web应用日志中的第一路径path。

[0104] 具体包括：

[0105] 1，获取Web应用日志。

[0106] Web应用日志需满足记录了如下字段：客户端来访IP地址(remote_user)、记录访问时间(time_local)、记录请求的URL和HTTP协议(request)、记录请求的响应状态(status)、记录web应用响应大小(body_bytes_sent)、记录从哪个页面链接访问过来的(http_referer)、记录客户端浏览器相关信息(http_user_agent)、用户身份(http_cookie)。

[0107] 如，从各类Web应用代理(nginx、apache等)服务器上，按天将web应用日志存入本地检测服务器相应目录下。

[0108] 2，对Web应用日志进行格式化处理，获取第一路径path。

[0109] 其中，第一path由至少一个第一字符串组成。

[0110] 对Web日志进行统一格式化处理。拆分出request字段中的请求方法(method)、页面(path)、请求参数(para)，拆分出http_referer字段中的主机(host)、页面(path)这些数据信息。

[0111] 将从request字段中获取的path以及将从http_referer字段中获取path均作为第一路径path。

[0112] 1)quest字段拆分：例如将"GET/ab/cd/ef.jsp?id＝123&name＝456HTTP/1.1"拆

分为{“method”:”GET”,”path”:”/ab/cd/ef.jsp”,”para”:”id=123&name=456”},获取path为/ab/cd/ef.jsp。

[0113] 2)http_referer字段拆分:例如将“https://www.abc.com/index.jsp”拆分为{“host”:”www.abc.com”,”path”:”/index.jsp”},获取path为/index.jsp。

[0114] 103,根据预先规约模型,删除第一path中的无效第一字符串,得到标准化的第一path。

[0115] 将第一path中的各第一字符串分别与步骤101确定的白名单中的第二字符串进行对比,相同则确定为有效字符串,不同则确定为无效字符串。删除无效字符串,得到标准化的第一path。

[0116] 经过此步骤,可以实现利用k-means来寻找有效字符串的功能,进而免去过于复杂的Webshell特征规则库维护工作,减少运维成本,也无需识别Web应用开发语言,无需区分jsp、php、asp、cgi等具体Webshell分类,对所有Webshell具有通用性。

[0117] 另外,利用k-means来寻找有效字符串这种分析Web服务器日志的旁路方式,无需在应用服务器上部署检测脚本,也无需高成本的解析网络镜像流量,降低Webshell检测成本。

[0118] 例如,path中存在hash值的情况,如表1所示,将/test/login/da853b0d3f88d99b、/test/login/89c86e2691cfc444通过归约模型得到标准化的第一path为/test/login。

[0119] 表1

[0120]

原始 path	标准化的第一 path
/test/login/da853b0d3f88d99b	/test/login
/test/login/89c86e2691cfc444	
/test/login/…	

[0121] 再例如:path中存在随机数的情况,如表2所示,将/test/user/0.633545824953、/test/user/0.249536335458通过归约模型得到标准化的第一path为/test/user。

[0122] 表2

[0123]

原始 path	标准化的第一 path
/test/user/0.633545824953	/test/user
/test/user/0. 249536335458	
/test/user/…	

[0124] 104,根据标准化的第一path,抽取Web应用日志的页面特征。

[0125] 本步骤可以抽取所有Web应用日志的页面特征。

[0126] 但是对于不存在页面等的Web应用日志,无需再抽取页面特征,因此,本步骤的优

选实现方式为:过滤无效页面的Web应用日志,抽取过滤后Web应用日志的页面特征。

[0127] 具体为:1)选择请求的响应状态为200和500的Web应用日志。2)根据标准化的第一path,抽取选择的Web应用日志的页面特征。

[0128] 其中,1)可以通过选择status为200和500的Web应用日志实现。

[0129] 2)在实现时,因Webshell文件与正常页面文件访问特征会有诸多不同,例如:Webshell出度、入度一般小于或等于1,正常页面出入度一般大于3;Webshell来访IP相对单一,正常页面会有很多IP访问等等特征。

[0130] 现有技术方案提取的特征比较少,例如有的方案只是考虑页面的出入度或页面曝光时间这种Webshell主要特征,而忽略了其他一些次要特征。Webshell特征维度提取的尽量全面,对于Webshell的发现效果会更准确,因此本提案抽取如表3所示的页面特征:

[0131] 表3

[0132]

序号	页面特征	序号	页面特征
1	入度	2	出度
3	初次访问时间	4	最新访问时间
5	页面曝光天数	6	页面出现过天数
7	页面 get 请求次数	8	页面 post 请求次数
9	页面 500 次数	10	页面响应返回最大值
11	页面响应返回最小值	12	页面响应返回平均值
13	来访 IP（Internet Protocol，网络之间互连的协议）数量	14	来访 IP 的 C 段数量
15	来访 UA（User Agent，用户代理）数量	16	URI 访问有参数访问次数
17	URI 访问无参数访问次数	18	页面访问量曝光天数比
19	页面访问量出现过天数比	20	页面出现不同的时间
21	访问量与页面出现不同的时间的商	22	Refer 总数
23	Refer 等于 path 数量	24	Refer 等于 path 占比
25	页面对应 Session ID（Identity，标识）数量	26	页面访问量

[0133]

27	页面 500 次数占比	28	页面 200 次数
29	POST 请求占比	30	页面响应值不同数
31	页面响应值不同占比	32	UA 异常数
33	Session ID 异常数	34	Session ID 异常占比
35	页面响应最大值与页面响应平均值的商		

[0134] 上述35个特征并非Webshell检测的常用特征，本提案创新的将上述35个特征用于Webshell检测，获得了较佳的检查效果，因此将上述35个特征用于Webshell检测是本提案的创新之一。

[0135] 例如，特征24：Refer等于path占比。很多类型的webshell都是直接通过浏览器访问、操作，这种访问webshell的习惯，在web服务器日志中记录的refer字段中的path与页面path是相同的，而访问正常页面绝大部分不具备此种特征，通过使用这一特征，增加对webshell的发现能力。

[0136] 再例如，特征25：页面对应Session ID。正常页面会有很多不同的用户来访问，访问webshell的用户基本上都是攻击者本人，通过这一特征可以增加判断webshell的维度。

[0137] 另外，上述特征13来访IP数量、14来访IP的C段数量、15来访UA数量可以采用去重后的数据。

[0138] 由于在步骤103中，将同一页面对应的Web应用日志中的第一path都归约成相同的标准化的第一path，也即标准化的第一path相同的Web应用日志对应相同的页面。因此，本步骤在具体实施时，可以1)按标准化的第一path的不同，将选择的Web应用日志分类，其中，每类中各选择的Web应用日志对应相同的标准化的第一path；2)基于每类中选择的Web应用日志分类，抽取每类的页面特征。

[0139] 105，根据页面特征，对Web应用日志进行Webshell检测。

[0140] 根据GMM、Local Outlier Factor、Isolation Forest算法分别计算出的各URI异常分数值，取各算法最异常的URI交集，生成webshell告警。

[0141] 本提案的实现方案为：分别利用孤立森林模型(Isolation Forest)、高斯混合模型(GMM)、局部异常因子检测模型(Local Outlier Factor)判断页面特征是否异常，再通过添加阈值使每一个算法都有决策权，最终通过组合模型管用的投票机制选择异常。

[0142] 如，根据GMM、Local Outlier Factor、Isolation Forest算法分别计算出的各URI异常分数值，取各算法最异常的URI交集，生成webshell告警。

[0143] 若孤立森林模型、高斯混合模型、局部异常因子检测模型均判断页面特征异常，则确定Web应用日志为Webshell访问异常数据；

[0144] 若孤立森林模型、高斯混合模型、局部异常因子检测模型均判断页面特征正常，则确定Web应用日志为Webshell访问正常数据；

[0145] 若孤立森林模型、高斯混合模型、局部异常因子检测模型中2个模型判断页面特征

异常，1个模型判断页面特征正常，则确定Web应用日志为Webshell访问较异常数据；

[0146] 若孤立森林模型、高斯混合模型、局部异常因子检测模型中2个模型判断页面特征正常，1个模型判断页面特征异常，则确定Web应用日志为Webshell访问一般异常数据。

[0147] 其中，

[0148] 1，GMM判断方法为：

[0149] 1)利用k-means对输入的维度进行聚类。

[0150] 2)对聚好的类分别求高斯分布。

[0151] $$f_x(x_1,\cdots,x_k)=\frac{1}{\sqrt{(2\pi)^k|\Sigma|}}\exp(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu))$$

[0152] 其中，$\sqrt{|\Sigma|}$是标准差，μ是平均值，$(x-\mu)^T\Sigma^{-1}(x-\mu)$是计算k维空间中$(x_1,\cdots,x_k)$到μ的n维距离的平方；

[0153] 3)利用加权平均求出联合分布，如图4所示。

[0154] 2，Local Outlier Factor判断方法为：

[0155] 1)设定k值，对指定点A找到最邻近k个点。

[0156] 2)计算A点到这些点的欧式距离记为$N_k(A)$，其中最大距离为A点可探测距离。

[0157] $reachability\_distance_k(A,B)=\max\{k\_distance(B),d(A,B)\}$。

[0158] 其中，k_distance(B)是B点到它k个邻近点的距离的集合。

[0159] $reachability_distance_k(A,B)$是A对B的可达距离，在k_distance(B)和A到B的距离中选择一个最大的。

[0160] 3)求出局部可达密度：

[0161] $$lrd(A):=1/(\frac{\sum_{B\in N_K A}reachability\_distance_k(A,B)}{|N_k(A)|}。$$

[0162] 其中，lrd(A)是A点的局部可达密度，首先对A点的所有可达B点的做一个平均；其中$|N_k(A)|$是$N_k(A)$中的点的个数。

[0163] 4)同样的方法对其他点的进行计算，算出局部异常因子：

[0164] $$LOF_k A:=\frac{\sum_{B\in N_K A}\frac{lrd(B)}{lrd(A)}}{|N_k(A)|}=\frac{\sum_{B\in N_K A}lrd(B)}{|N_k(A)|}/lrd(A)。$$

[0165] 其中，LOF_kA是局部异常因子，所有B点的局部可达密度除以A点的局部可达密度，然后求平均得到。

[0166] 3，Isolation Forest判断方法为：

[0167] Isolation Forest属于一种无参数的非监督算法，他是一种侦测异常十分有效的组合算法，底层用的是决策树。

[0168] 1)建立n个决策树模型。

[0169] 2)随机抽取样本数据输入这n个模型进行训练(随机按最大最小切割)，切到指定异常数据比例为止。

[0170] 3)利用n个决策树进行投票，求出异常系数。

[0171] Isolation Forest结果示意图如图5所示。

[0172] 通过本提案提供的上述Webshell检测方法，可以降低Webshell检测的误报率、低漏报率。

[0173] 另外，免去过于复杂的Webshell特征规则库维护工作，可以减少运维成本。

[0174] 此外，采用分析Web服务器日志的旁路方式，无需在应用服务器上部署检测脚本，也无需高成本的解析网络镜像流量，降低Webshell检测成本。

[0175] 此外，无需识别Web应用开发语言，无需区分jsp、php、asp、cgi等具体Webshell分类，对所有Webshell具有通用性。

[0176] 需要说明的是，本实施例及后续实施例所涉及的第一、第二、第三、第四仅为标识，并无实质意义。如，第一path和第二path可能相同也可能不同。

[0177] 有益效果：

[0178] 根据预先由字符串聚类得到的规约模型，删除path中的无效字符串，得到标准化的path；根据标准化的path，抽取Web应用日志的页面特征；根据页面特征，对Web应用日志进行Webshell检测，实现了根据规约模型选择标准字符串，进而将涉及同一页面的Web应用日志中的path标准化为相同的path，进而基于path的不同抽取页面特征，进行Webshell检测，不仅具有通用性，降低检测成本，而且降低误报率和漏报率。

[0179] 基于同一发明构思，本实施例提供了一种电子设备，参见图6，包括存储器601、处理器602、总线603以及存储在存储器601上并可在处理器602上运行的计算机程序，所述处理器602执行所述程序时实现如下步骤。

[0180] 获取Web应用日志中的第一路径path，第一path由至少一个第一字符串组成；

[0181] 根据预先由字符串聚类得到的规约模型，删除第一path中的无效第一字符串，得到标准化的第一path；

[0182] 根据标准化的第一path，抽取Web应用日志的页面特征；

[0183] 根据页面特征，对Web应用日志进行Webshell检测。

[0184] 可选地，根据预先训练的规约模型删除第一path中的无效字符串，得到标准化的第一path之前，还包括：

[0185] 获取多个第二path，第二path由至少一个第二字符串组成；

[0186] 获取各第二path中的第二字符串；

[0187] 将各第二字符串的各字符转换成ASCII值，形成各第二字符串对应的数组；

[0188] 确定各数组的长度、平均值和标准差；

[0189] 根据各数组的长度、平均值和标准差对所有第二path的所有第二字符串进行聚类；

[0190] 根据预先设定的分类规则，确认聚类结果中的有效聚类；

[0191] 将有效聚类涉及的第二字符串形成白名单，并将白名单确定为训练的规约模型。

[0192] 可选地，分类规则包括至少一个第三字符串，第三字符串为有效聚类的示例字符串；

[0193] 根据预先设定的分类规则，确认聚类结果中的有效聚类，包括：

[0194] 计算第三字符串的中心点；

[0195] 计算聚类结果中各聚类中心点距第三字符串的中心点的第一距离；

[0196] 将第一距离小于预设第一阈值的聚类确定为有效聚类。

[0197] 可选地，分类规则包括至少一个第四字符串，第四字符串为无效聚类的示例字符串；

[0198] 根据预先设定的分类规则，确认聚类结果中的有效聚类，包括：

[0199] 计算第四字符串的中心点；

[0200] 计算聚类结果中各聚类中心点距第四字符串的中心点的第二距离；

[0201] 将第二距离大于预设第二阈值的聚类确定为有效聚类。

[0202] 可选地，根据标准化的第一path，抽取Web应用日志的页面特征之前，还包括：

[0203] 选择请求的响应状态为200和500的Web应用日志；

[0204] 根据标准化的第一path，抽取Web应用日志的页面特征，包括：

[0205] 根据标准化的第一path，抽取选择的Web应用日志的页面特征。

[0206] 可选地，根据标准化的第一path，抽取选择的Web应用日志的页面特征，包括：

[0207] 按标准化的第一path的不同，将选择的Web应用日志分类，其中，每类中各选择的Web应用日志对应相同的标准化的第一path；

[0208] 基于每类中选择的Web应用日志分类，抽取每类的页面特征。

[0209] 可选地，页面特征包括：入度、出度、初次访问时间、最新访问时间、页面曝光天数、页面出现过天数、页面get请求次数、页面post请求次数、页面500次数、页面响应返回最大值、页面响应返回最小值、页面响应返回平均值、来访网络之间互连的协议IP数量、来访IP的C段数量、来访用户代理UA数量、统一资源标识符URI访问有参数访问次数、URI访问无参数访问次数、页面访问量曝光天数比、页面访问量出现过天数比、页面出现不同的时间、访问量与页面出现不同的时间的商、Refer总数、Refer等于path数量、Refer等于path占比、页面对应Session标识ID数量、页面访问量、页面500次数占比、页面200次数、POST请求占比、页面响应值不同数、页面响应值不同占比、UA异常数、Session ID异常数、Session ID异常占比、页面响应最大值与页面响应平均值的商。

[0210] 可选地，根据页面特征，对Web应用日志进行Webshell检测，包括：

[0211] 分别利用孤立森林模型、高斯混合模型、局部异常因子检测模型判断页面特征是否异常；

[0212] 若孤立森林模型、高斯混合模型、局部异常因子检测模型均判断页面特征异常，则确定Web应用日志为Webshell访问异常数据；

[0213] 若孤立森林模型、高斯混合模型、局部异常因子检测模型均判断页面特征正常，则确定Web应用日志为Webshell访问正常数据；

[0214] 若孤立森林模型、高斯混合模型、局部异常因子检测模型中2个模型判断页面特征异常，1个模型判断页面特征正常，则确定Web应用日志为Webshell访问较异常数据；

[0215] 若孤立森林模型、高斯混合模型、局部异常因子检测模型中2个模型判断页面特征正常，1个模型判断页面特征异常，则确定Web应用日志为Webshell访问一般异常数据。

[0216] 有益效果如下：

[0217] 根据预先由字符串聚类得到的规约模型，删除path中的无效字符串，得到标准化的path；根据标准化的path，抽取Web应用日志的页面特征；根据页面特征，对Web应用日志进行Webshell检测，实现了根据规约模型选择标准字符串，进而将涉及同一页面的Web应用

日志中的path标准化为相同的path，进而基于path的不同抽取页面特征，进行Webshell检测，不仅具有通用性，降低检测成本，而且降低误报率和漏报率。

[0218] 基于同一发明构思，本实施例提供了一种计算机存储介质，其上存储有计算机程序所述程序被处理器执行时实现如下步骤。

[0219] 获取Web应用日志中的第一路径path，第一path由至少一个第一字符串组成；

[0220] 根据预先由字符串聚类得到的规约模型，删除第一path中的无效第一字符串，得到标准化的第一path；

[0221] 根据标准化的第一path，抽取Web应用日志的页面特征；

[0222] 根据页面特征，对Web应用日志进行Webshell检测。

[0223] 可选地，根据预先训练的规约模型删除第一path中的无效字符串，得到标准化的第一path之前，还包括：

[0224] 获取多个第二path，第二path由至少一个第二字符串组成；

[0225] 获取各第二path中的第二字符串；

[0226] 将各第二字符串的各字符转换成ASCII值，形成各第二字符串对应的数组；

[0227] 确定各数组的长度、平均值和标准差；

[0228] 根据各数组的长度、平均值和标准差对所有第二path的所有第二字符串进行聚类；

[0229] 根据预先设定的分类规则，确认聚类结果中的有效聚类；

[0230] 将有效聚类涉及的第二字符串形成白名单，并将白名单确定为训练的规约模型。

[0231] 可选地，分类规则包括至少一个第三字符串，第三字符串为有效聚类的示例字符串；

[0232] 根据预先设定的分类规则，确认聚类结果中的有效聚类，包括：

[0233] 计算第三字符串的中心点；

[0234] 计算聚类结果中各聚类中心点距第三字符串的中心点的第一距离；

[0235] 将第一距离小于预设第一阈值的聚类确定为有效聚类。

[0236] 可选地，分类规则包括至少一个第四字符串，第四字符串为无效聚类的示例字符串；

[0237] 根据预先设定的分类规则，确认聚类结果中的有效聚类，包括：

[0238] 计算第四字符串的中心点；

[0239] 计算聚类结果中各聚类中心点距第四字符串的中心点的第二距离；

[0240] 将第二距离大于预设第二阈值的聚类确定为有效聚类。

[0241] 可选地，根据标准化的第一path，抽取Web应用日志的页面特征之前，还包括：

[0242] 选择请求的响应状态为200和500的Web应用日志；

[0243] 根据标准化的第一path，抽取Web应用日志的页面特征，包括：

[0244] 根据标准化的第一path，抽取选择的Web应用日志的页面特征。

[0245] 可选地，根据标准化的第一path，抽取选择的Web应用日志的页面特征，包括：

[0246] 按标准化的第一path的不同，将选择的Web应用日志分类，其中，每类中各选择的Web应用日志对应相同的标准化的第一path；

[0247] 基于每类中选择的Web应用日志分类，抽取每类的页面特征。

[0248] 可选地，页面特征包括：入度、出度、初次访问时间、最新访问时间、页面曝光天数、页面出现过天数、页面get请求次数、页面post请求次数、页面500次数、页面响应返回最大值、页面响应返回最小值、页面响应返回平均值、来访网络之间互连的协议IP数量、来访IP的C段数量、来访用户代理UA数量、统一资源标识符URI访问有参数访问次数、URI访问无参数访问次数、页面访问量曝光天数比、页面访问量出现过天数比、页面出现不同的时间、访问量与页面出现不同的时间的商、Refer总数、Refer等于path数量、Refer等于path占比、页面对应Session标识ID数量、页面访问量、页面500次数占比、页面200次数、POST请求占比、页面响应值不同数、页面响应值不同占比、UA异常数、Session ID异常数、Session ID异常占比、页面响应最大值与页面响应平均值的商。

[0249] 可选地，根据页面特征，对Web应用日志进行Webshell检测，包括：

[0250] 分别利用孤立森林模型、高斯混合模型、局部异常因子检测模型判断页面特征是否异常；

[0251] 若孤立森林模型、高斯混合模型、局部异常因子检测模型均判断页面特征异常，则确定Web应用日志为Webshell访问异常数据；

[0252] 若孤立森林模型、高斯混合模型、局部异常因子检测模型均判断页面特征正常，则确定Web应用日志为Webshell访问正常数据；

[0253] 若孤立森林模型、高斯混合模型、局部异常因子检测模型中2个模型判断页面特征异常，1个模型判断页面特征正常，则确定Web应用日志为Webshell访问较异常数据；

[0254] 若孤立森林模型、高斯混合模型、局部异常因子检测模型中2个模型判断页面特征正常，1个模型判断页面特征异常，则确定Web应用日志为Webshell访问一般异常数据。

[0255] 有益效果如下：

[0256] 根据预先由字符串聚类得到的规约模型，删除path中的无效字符串，得到标准化的path；根据标准化的path，抽取Web应用日志的页面特征；根据页面特征，对Web应用日志进行Webshell检测，实现了根据规约模型选择标准字符串，进而将涉及同一页面的Web应用日志中的path标准化为相同的path，进而基于path的不同抽取页面特征，进行Webshell检测，不仅具有通用性，降低检测成本，而且降低误报率和漏报率。

[0257] 上述实施例中，均可以采用现有的功能元器件模块来实施。例如，处理模块可以采用现有的数据处理元器件，至少，现有定位技术中采用的定位服务器上便具备实现该功能元器件；至于接收模块，则是任意一个具备信号传输功能的设备都具备的元器件；同时，处理模块进行的A、n参数计算、强度调整等采用的都是现有的技术手段，本领域技术人员经过相应的设计开发即可实现。

[0258] 为了描述的方便，以上所述装置的各部分以功能分为各种模块或单元分别描述。当然，在实施本发明时可以把各模块或单元的功能在同一个或多个软件或硬件中实现。

[0259] 本领域内的技术人员应明白，本发明的实施例可提供为方法、系统、或计算机程序产品。因此，本发明可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且，本发明可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器、CD-ROM、光学存储器等)上实施的计算机程序产品的形式。

[0260] 本发明是参照根据本发明实施例的方法、设备(系统)、和计算机程序产品的流程

图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器，使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[0261] 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中，使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品，该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

[0262] 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上，使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理，从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

[0263] 尽管已描述了本发明的优选实施例，但本领域内的技术人员一旦得知了基本创造性概念，则可对这些实施例作出另外的变更和修改。所以，所附权利要求意欲解释为包括优选实施例以及落入本发明范围的所有变更和修改。

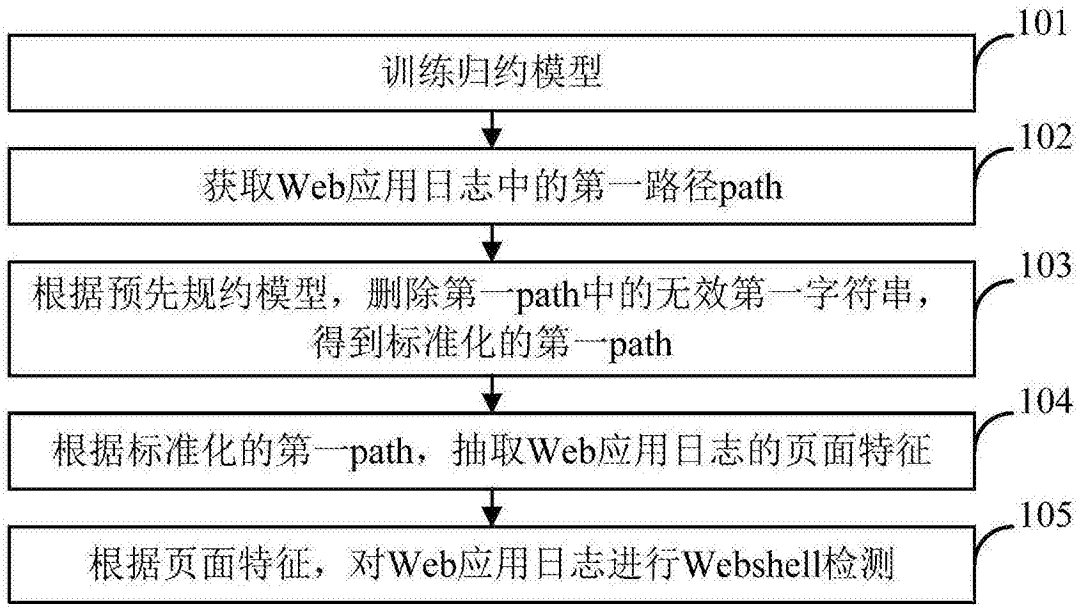

101
训练归约模型
102
获取Web应用日志中的第一路径path
103
根据预先规约模型，删除第一path中的无效第一字符串，得到标准化的第一path
104
根据标准化的第一path，抽取Web应用日志的页面特征
105
根据页面特征，对Web应用日志进行Webshell检测

图1

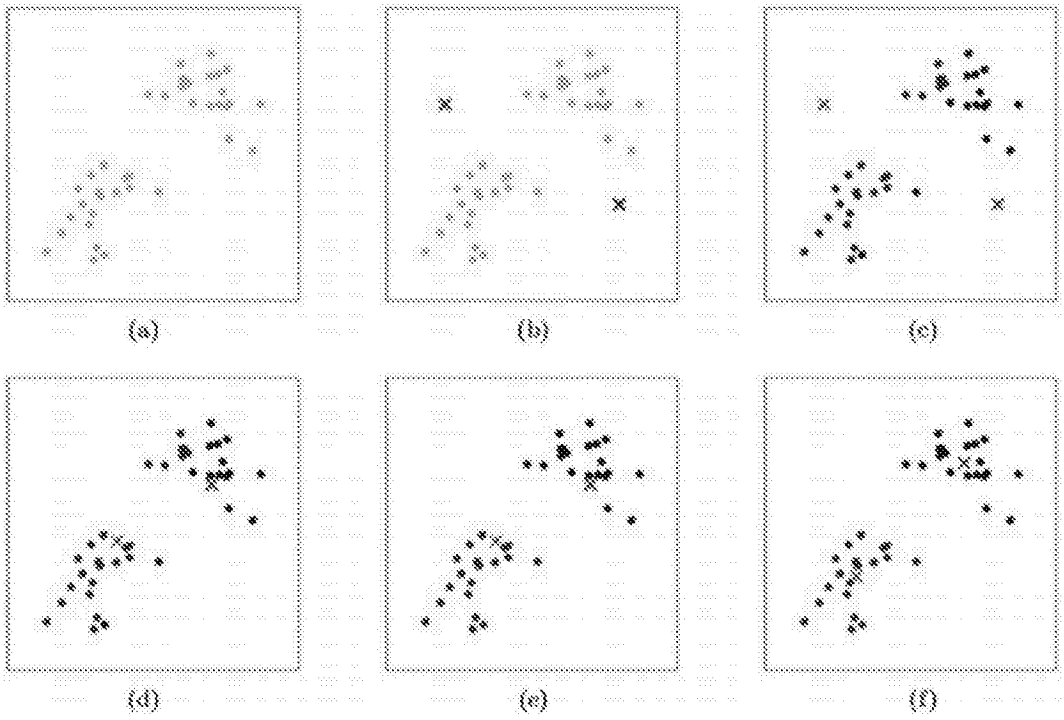

(a)
(b)
(c)
(d)
(e)
(f)

图2

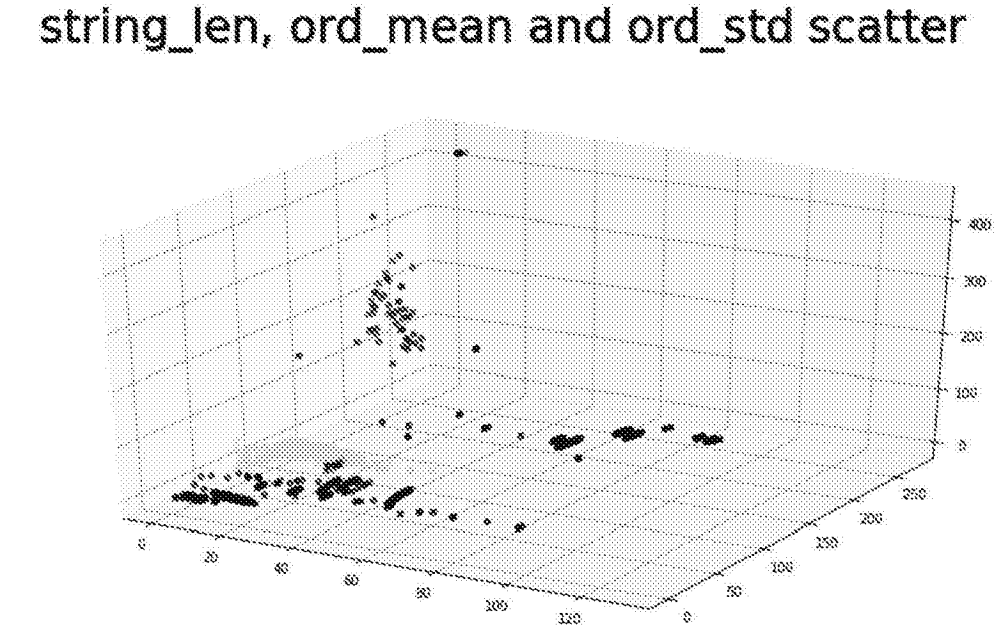

string_len, ord_mean and ord_std scatter

图3

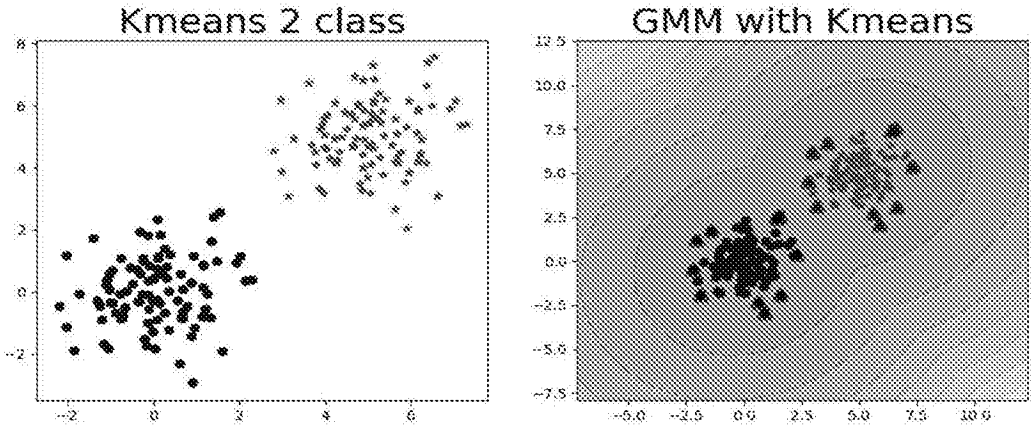

Kmeans 2 class
GMM with Kmeans

图4

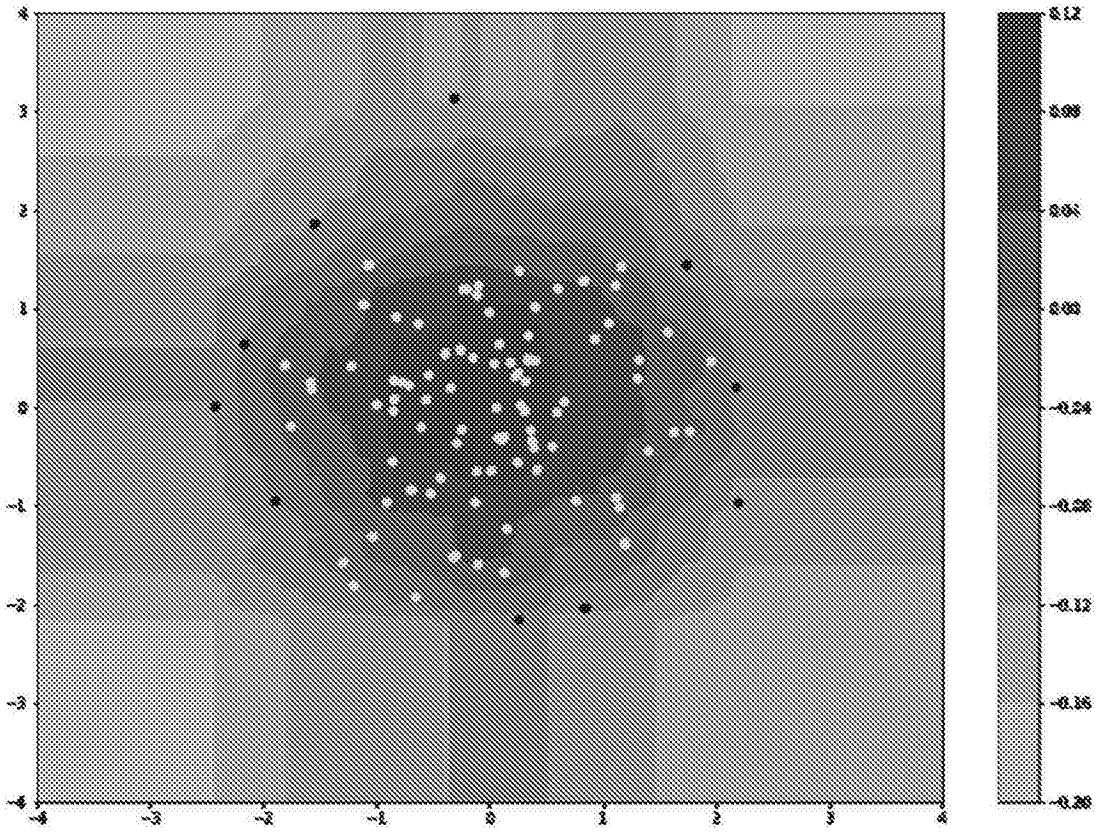


图5

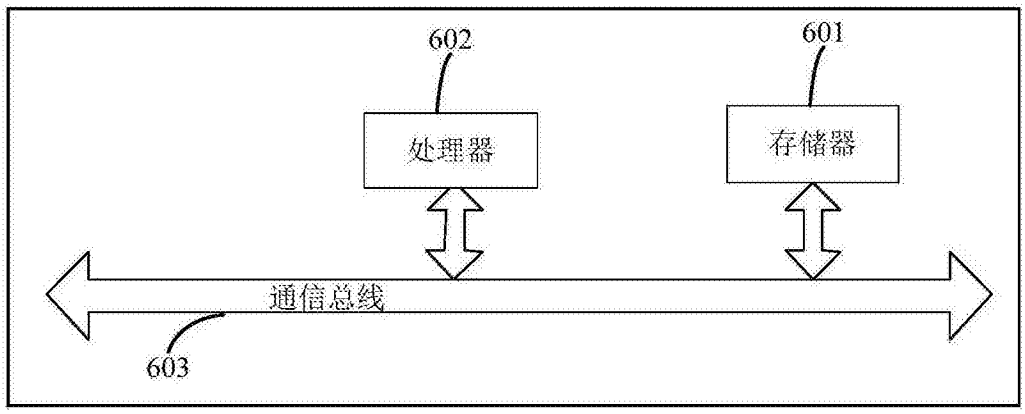


图6