(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 103577756 A
(43)申请公布日 2014.02.12

(21)申请号 201310544226.3

(22)申请日 2013.11.05

(71)申请人 北京奇虎科技有限公司
地址 100088 北京市西城区新街口外大街28号D座112室(德胜园区)
申请人 奇智软件(北京)有限公司

(72)发明人 陈卓 范纪鎤 杨康 唐海

(74)专利代理机构 北京市浩天知识产权代理事务所 11276
代理人 宋菲 刘兰兰

(51)Int.Cl.
G06F 21/56(2013.01)

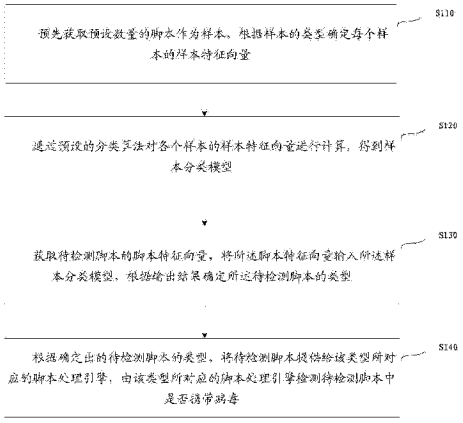
权利要求书2页 说明书14页 附图2页

(54)发明名称

基于脚本类型判断的病毒检测方法及装置

(57)摘要

本发明涉及一种基于脚本类型判断的病毒检测方法及装置。该方法包括:预先获取预设数量的脚本作为样本,根据样本的类型确定每个样本的样本特征向量,并通过预设的分类算法对各个样本的样本特征向量进行计算,得到样本分类模型;获取待检测脚本的脚本特征向量,将脚本特征向量输入样本分类模型,根据输出结果确定待检测脚本的类型;根据确定出的待检测脚本的类型,将待检测脚本提供给该类型所对应的脚本处理引擎,由脚本处理引擎检测待检测脚本中是否携带病毒。由此解决了现有技术中由人工分析脚本类型所导致的耗费时间和精力,以及由此所导致的不便于将脚本类型判断的方式应用于脚本病毒检测领域的技术问题。



CN 103577756 A

1. 一种基于脚本类型判断的病毒检测方法，包括：

预先获取预设数量的脚本作为样本，根据样本的类型确定每个样本的样本特征向量，并通过预设的分类算法对各个样本的样本特征向量进行计算，得到样本分类模型；

获取待检测脚本的脚本特征向量，将所述脚本特征向量输入所述样本分类模型，根据输出结果确定所述待检测脚本的类型；

根据确定出的待检测脚本的类型，将所述待检测脚本提供给该类型所对应的脚本处理引擎，由该类型所对应的脚本处理引擎检测所述待检测脚本中是否携带病毒。

2. 如权利要求 1 所述的方法，其中，所述根据样本的类型确定每个样本的样本特征向量的步骤包括：

根据样本的类型，分别设定各个类型的样本所对应的至少一个样本目标特征；

对于每个样本，根据该样本的类型确定该样本所对应的各个样本目标特征，并在该样本中查找并计算每个样本目标特征的出现频率；

将每个样本中的各个样本目标特征及其出现频率对应存储为一个样本特征向量。

3. 如权利要求 2 所述的方法，其中，所述样本目标特征包括：字符串、字词、语句和 / 或标点。

4. 如权利要求 2 或 3 所述的方法，其中，所述获取待检测脚本的脚本特征向量的步骤包括：

预先设定所有待检测脚本所对应的至少一个脚本目标特征，其中，每个待检测脚本所对应的脚本目标特征相同；

对于每个待检测脚本，在该待检测脚本中查找并计算每个脚本目标特征的出现频率；

将每个待检测脚本中的各个脚本目标特征及其出现频率对应存储为一个脚本特征向量。

5. 如权利要求 4 所述的方法，其中，所述脚本目标特征包括：各个类型的样本所对应的样本目标特征。

6. 如权利要求 1-5 任一所述的方法，其中，当所述分类算法为决策树算法时，所述样本分类模型为决策树模型；

当所述分类算法为支持向量机 SVM 算法时，所述样本分类模型为 SVM 模型；或者，

当所述分类算法为贝叶斯算法时，所述样本分类模型为贝叶斯模型。

7. 如权利要求 6 所述的方法，其中，当所述分类算法为决策树算法，所述样本分类模型为决策树模型时，所述通过预设的分类算法对各个样本的样本特征向量进行计算，得到样本分类模型的步骤包括：

先对部分样本的样本特征向量进行训练，得到待修正的决策树模型；

当判断出所述待修正的决策树模型不满足预设精度时，继续对剩余样本的样本特征向量进行训练，直到训练后得到的决策树模型满足预设精度。

8. 如权利要求 1 所述的方法，其中，所述样本的类型以及待检测脚本的类型根据脚本格式和 / 或脚本功能划分。

9. 一种基于脚本类型判断的病毒检测装置，包括：

获取单元，适于预先获取预设数量的脚本作为样本；

模型生成单元，适于根据样本的类型确定每个样本的样本特征向量，并通过预设的分

类算法对各个样本的样本特征向量进行计算，得到样本分类模型；

脚本判断单元，适于获取待检测脚本的脚本特征向量，将所述脚本特征向量输入所述样本分类模型，根据输出结果确定所述待检测脚本的类型；

病毒检测单元，适于根据确定出的待检测脚本的类型，将所述待检测脚本提供给该类型所对应的脚本处理引擎，由所述脚本处理引擎检测所述待检测脚本中是否携带病毒。

10. 如权利要求 9 所述的装置，其中，所述模型生成单元进一步包括：

第一设定子单元，适于根据样本的类型，分别设定各个类型的样本所对应的至少一个样本目标特征；

第一查找子单元，适于对于每个样本，根据该样本的类型确定该样本所对应的各个样本目标特征，并在该样本中查找并计算每个样本目标特征的出现频率；

第一存储子单元，适于将每个样本中的各个样本目标特征及其出现频率对应存储为一个样本特征向量。

基于脚本类型判断的病毒检测方法及装置

技术领域

[0001] 本发明涉及网络通信技术领域，具体涉及一种基于脚本类型判断的病毒检测方法及装置。

背景技术

[0002] 计算机病毒是指编制者在计算机程序中插入的破坏计算机功能或者破坏数据，影响计算机使用并且能够自我复制的一组计算机指令或者程序代码。计算机一旦染上病毒，通常表现为其文件被增加、删除、改变名称或属性、移动到其它目录下，病毒对计算机文件的这些操作，可能会导致正常的程序无法运行、计算机操作系统崩溃、计算机被远程控制、用户信息被盗用等一系列的问题。

[0003] 脚本（script）是使用一种特定的描述性语言，依据一定的格式编写的可执行文件，又称作宏或批处理文件。通常，脚本可以由应用程序临时调用并执行，并且，因为脚本不仅可以减小网页的规模和提高网页浏览速度，而且可以丰富网页的表现，如动画、声音等，所以，各类脚本经常被广泛地应用于网页设计中。例如，当点击网页上的 Email 地址时能自动调用 Outlook Express 或 Foxmail 这类邮箱软件的功能就是通过脚本来实现的。

[0004] 正是由于脚本使用方便且应用广泛的特点，往往被黑客等别有用心的人加以利用，使脚本成为上述的计算机病毒传播的载体，例如，在脚本中加入一些破坏计算机系统的命令，这样当用户浏览网页时，一旦调用该脚本，便会使用户的系统受到攻击。其中，脚本所携带的病毒往往与脚本的类型有很大关联，不同类型的脚本容易感染的病毒也各不相同，因此，明确了脚本类型之后，就能根据脚本的类型，对该类型的脚本所容易感染的病毒进行有针对性的检测。所以，在脚本病毒检测领域，预先识别出脚本类型对于提高病毒检测的效率和准确度有着很大帮助。

[0005] 但是，脚本的类型往往不能直观地通过后缀名进行判断，目前，为了准确识别出这些脚本的类型，通常的做法是由人工分析的方式，对脚本中的可执行代码逐行进行分析，根据分析得到的语法等特征来判断脚本的类型。但是，这种人工分析的方式非常耗费时间和精力，无法快速识别出脚本的类型，因而不便于应用到脚本病毒检测领域。

发明内容

[0006] 鉴于上述问题，提出了本发明以便提供一种克服上述问题或者至少部分地解决上述问题的基于脚本类型判断的病毒检测方法及装置。

[0007] 依据本发明的一个方面，提供了一种基于脚本类型判断的病毒检测方法，包括：预先获取预设数量的脚本作为样本，根据样本的类型确定每个样本的样本特征向量，并通过预设的分类算法对各个样本的样本特征向量进行计算，得到样本分类模型；获取待检测脚本的脚本特征向量，将脚本特征向量输入样本分类模型，根据输出结果确定待检测脚本的类型；根据确定出的待检测脚本的类型，将待检测脚本提供给该类型所对应的脚本处理引擎，由该类型所对应的脚本处理引擎检测待检测脚本中是否携带病毒。

[0008] 可选地，根据样本的类型确定每个样本的样本特征向量的步骤包括：根据样本的类型，分别设定各个类型的样本所对应的至少一个样本目标特征；对于每个样本，根据该样本的类型确定该样本所对应的各个样本目标特征，并在该样本中查找并计算每个样本目标特征的出现频率；将每个样本中的各个样本目标特征及其出现频率对应存储为一个样本特征向量。

[0009] 可选地，样本目标特征包括：字符串、字词、语句和/或标点。

[0010] 可选地，获取待检测脚本的脚本特征向量的步骤包括：预先设定所有待检测脚本所对应的至少一个脚本目标特征，其中，每个待检测脚本所对应的脚本目标特征相同；对于每个待检测脚本，在该待检测脚本中查找并计算每个脚本目标特征的出现频率；将每个待检测脚本中的各个脚本目标特征及其出现频率对应存储为一个脚本特征向量。

[0011] 可选地，脚本目标特征包括：各个类型的样本所对应的样本目标特征。

[0012] 可选地，当分类算法为决策树算法时，样本分类模型为决策树模型；当分类算法为支持向量机SVM算法时，样本分类模型为SVM模型；或者，当分类算法为贝叶斯算法时，样本分类模型为贝叶斯模型。

[0013] 可选地，当分类算法为决策树算法，样本分类模型为决策树模型时，通过预设的分类算法对各个样本的样本特征向量进行计算，得到样本分类模型的步骤包括：先对部分样本的样本特征向量进行训练，得到待修正的决策树模型；当判断出待修正的决策树模型不满足预设精度时，继续对剩余样本的样本特征向量进行训练，直到训练后得到的决策树模型满足预设精度。

[0014] 可选地，样本的类型以及待检测脚本的类型根据脚本格式和/或脚本功能划分。

[0015] 可选地，脚本处理引擎的数量为多个，每个脚本处理引擎对应至少一个脚本类型，用于根据预设的该至少一个脚本类型对应的病毒样本特征来检测属于该至少一个脚本类型的待检测脚本中是否携带病毒，其中，所述多个脚本处理引擎之间相互并行工作。

[0016] 可选地，所述脚本处理引擎检测所述待检测脚本中是否携带病毒的步骤包括：获取待检测脚本的特征，判断所述待检测脚本的特征是否与预设的病毒特征库中的病毒特征匹配，若判断结果为是，则确定所述待检测脚本携带有病毒；其中，所述病毒特征库设置在客户端本地或云端服务器上，用于存储病毒文件的病毒特征，其中，所述病毒特征包括：md5值。

[0017] 依据本发明的一个方面，提供了一种基于脚本类型判断的病毒检测装置，包括：获取单元，适于预先获取预设数量的脚本作为样本；模型生成单元，适于根据样本的类型确定每个样本的样本特征向量，并通过预设的分类算法对各个样本的样本特征向量进行计算，得到样本分类模型；脚本判断单元，适于获取待检测脚本的脚本特征向量，将脚本特征向量输入样本分类模型，根据输出结果确定待检测脚本的类型；病毒检测单元，适于根据确定出的待检测脚本的类型，将待检测脚本提供给该类型所对应的脚本处理引擎，由脚本处理引擎检测待检测脚本中是否携带病毒。

[0018] 可选地，模型生成单元进一步包括：第一设定子单元，适于根据样本的类型，分别设定各个类型的样本所对应的至少一个样本目标特征；第一查找子单元，适于对于每个样本，根据该样本的类型确定该样本所对应的各个样本目标特征，并在该样本中查找并计算每个样本目标特征的出现频率；第一存储子单元，适于将每个样本中的各个样本目标特征

及其出现频率对应存储为一个样本特征向量。

[0019] 可选地，样本目标特征包括：字符串、字词、语句和 / 或标点。

[0020] 可选地，脚本判断单元进一步包括：第二设定子单元，适于预先设定所有待检测脚本所对应的至少一个脚本目标特征，其中，每个待检测脚本所对应的脚本目标特征相同；第二查找子单元，适于对于每个待检测脚本，在该待检测脚本中查找并计算每个脚本目标特征的出现频率；第二存储子单元，适于将每个待检测脚本中的各个脚本目标特征及其出现频率对应存储为一个脚本特征向量。

[0021] 可选地，脚本目标特征包括：各个类型的样本所对应的样本目标特征。

[0022] 可选地，当分类算法为决策树算法时，样本分类模型为决策树模型；当分类算法为支持向量机 SVM 算法时，样本分类模型为 SVM 模型；或者，当分类算法为贝叶斯算法时，样本分类模型为贝叶斯模型。

[0023] 可选地，当分类算法为决策树算法，样本分类模型为决策树模型时，模型生成单元用于：先对部分样本的样本特征向量进行训练，得到待修正的决策树模型；当判断出待修正的决策树模型不满足预设精度时，继续对剩余样本的样本特征向量进行训练，直到训练后得到的决策树模型满足预设精度。

[0024] 可选地，样本的类型以及待检测脚本的类型根据脚本格式和 / 或脚本功能划分。

[0025] 可选地，脚本处理引擎的数量为多个，每个脚本处理引擎对应至少一个脚本类型，用于根据预设的该至少一个脚本类型对应的病毒样本特征来检测属于该至少一个脚本类型的待检测脚本中是否携带病毒，其中，所述多个脚本处理引擎之间相互并行工作。

[0026] 可选地，所述脚本处理引擎用于获取待检测脚本的特征，判断所述待检测脚本的特征是否与预设的病毒特征库中的病毒特征匹配，若判断结果为是，则确定所述待检测脚本携带有病毒；其中，所述病毒特征库设置在客户端本地或云端服务器上，用于存储病毒文件的病毒特征，其中，所述病毒特征包括：md5 值。

[0027] 在本发明实施例提供的基于脚本类型判断的病毒检测方法及装置中，预先根据样本类型确定出各个样本的样本特征向量，并据此得到样本分类模型，该模型用于对脚本进行分类，由此，在需要判断脚本类型时，只需获取脚本的脚本特征向量，并根据预先得到的样本分类模型就可以完成脚本类型的判断。由此解决了现有技术中由人工分析脚本类型所导致的耗费时间和精力，以及由此所导致的不便于将脚本类型判断的方式应用于脚本病毒检测领域的技术问题，实现了能够便捷高效地识别脚本类型，并根据脚本的类型进行有针对性的病毒检测，以便提高检测效率和准确度的技术效果。

[0028] 上述说明仅是本发明技术方案的概述，为了能够更清楚了解本发明的技术手段，而可依照说明书的内容予以实施，并且为了让本发明的上述和其它目的、特征和优点能够更明显易懂，以下特举本发明的具体实施方式。

附图说明

[0029] 通过阅读下文优选实施方式的详细描述，各种其他的优点和益处对于本领域普通技术人员将变得清楚明了。附图仅用于示出优选实施方式的目的，而并不认为是对本发明的限制。而且在整个附图中，用相同的参考符号表示相同的部件。在附图中：

[0030] 图 1 示出了本发明实施例提供的基于脚本类型判断的病毒检测方法的流程图；以

及

[0031] 图 2 示出了本发明实施例提供的基于脚本类型判断的病毒检测装置的结构图。

具体实施方式

[0032] 下面将参照附图更详细地描述本公开的示例性实施例。虽然附图中显示了本公开的示例性实施例，然而应当理解，可以以各种形式实现本公开而不应被这里阐述的实施例所限制。相反，提供这些实施例是为了能够更透彻地理解本公开，并且能够将本公开的范围完整的传达给本领域的技术人员。

[0033] 本发明实施例提供了一种基于脚本类型判断的病毒检测方法及装置，用以解决现有技术中由人工分析脚本类型所导致的耗费时间和精力，以及由此所导致的不便于将脚本类型判断的方式应用于脚本病毒检测领域的技术问题。

[0034] 图 1 示出了本发明实施例提供的基于脚本类型判断的病毒检测方法流程图。如图 1 所示，该方法起始于步骤 S110，在步骤 S110 中，预先获取预设数量的脚本作为样本，根据样本的类型确定每个样本的样本特征向量。

[0035] 在步骤 S110 中，可以通过多种方式来获取作为脚本的样本。例如，可以预先通过多台虚拟机并行运行的方式来获取样本。其中，每台虚拟机可以运行多台 xp 对应的实体机，由此可以提升样本收集的效率。样本的数量可以根据实际情况来确定，例如，精确要求越高的情况下所需的样本数量也越多，反之样本数量则越低。另外，在本实施例中，可以通过预设的动态库，如 qex.dll，来实现获取样本的操作。当然，除了动态库的实现方式之外，本领域技术人员也可以采用其它的软、硬件编程方式或硬件设备来实现本实施例中的各个步骤。

[0036] 在步骤 S110 中，获取到预设数量的样本之后，还需要进一步确定每个样本的类型。其中，样本的类型可以通过多种方式进行划分：例如，可以根据脚本的格式、编写的语言和 / 或脚本的功能进行划分。在本实施例中，以通过脚本格式划分样本类型为例进行介绍，此时，确定样本类型的操作实质上就是分析脚本格式的步骤。在具体分析脚本格式时，可以通过多种方式来分析，本发明对具体的分析方式不做限定。例如，可以通过人工分析的方式逐行分析样本中的可执行代码；或者，也可以通过预先编写的程序来逐行分析样本中的可执行代码，以便确定样本的格式。

[0037] 在确定出各个样本的类型之后，还需要根据样本的类型来确定每个样本的样本特征向量。其中，样本特征向量的作用在于标识某一类型的样本的共同特征，因此，只要是能够反映出该类型的样本的共同特征的向量都可以作为样本特征向量，本发明对样本特征向量的具体选取方式不做限定。

[0038] 下面给出样本特征向量的一种可能的确定方式：

[0039] 首先，根据样本的类型，分别设定各个类型的样本所对应的至少一个样本目标特征。也就是说，每个类型的样本分别对应着一组或多组样本目标特征。

[0040] 例如，JS 格式的样本所对应的样本目标特征可以通过如下数据结构进行定义：

[0041]

static char const g_javascript_words [] =*

{

"break", "case", "catch", "class", "const", "continue", "debugger", "default", "delete", "do", "else", "enum", "export", "extends", "finally", "for", "function", "if" ,"implements", import", "in", "instanceof", "interface", "let","new", "package", "private", "protected", "public","return", "static", "super", "switch", "this", "throw", "try", "typeof", "var", "void", "while", "with", "yield",

};

[0042] 其中，大括号内包含的各个字词(例如"break","case","catch"等)均为JS格式的样本所对应的样本目标特征。这些作为样本目标特征的字词通常都是JS格式的脚本中的常用字词。

[0043] HTML格式的样本所对应的样本目标特征可以通过如下数据结构进行定义：

[0044]

static char const g_html_words [] =*

{

"A", "ABBR", "ACRONYM", "ADDRESS", "APPLET", "AREA", "B", "BASE", "BASEFONT", "BDO", "BIG", "BLOCKQUOTE", "BODY", "BR", "BUTTON", "CAPTION", "CENTER", "CITE", "CODE", "COL", "COLGROUP", "DD", "DEL", "DFN", "DIR", "DIV", "DL", "DT", "EM", "FIELDSET", "FONT", "FORM", "FRAME", "FRAMESET", "H1", "H2", "H3", "H4", "H5", "H6", "HEAD", "HR", "HTML", "I", "IFRAME", "IMG", "INPUT", "INS", "ISINDEX", "KBD", "LABEL", "LEGEND", "LI", "LINK", "MAP", "MENU", "META", "NOFRAMES", "NOSCRIPT", "OBJECT", "OL", "OPTGROUP", "OPTION",

[0045]

"P", "PARAM", "PRE", "Q", "S", "SAMP", "SCRIPT", "SELECT", "SMALL", "SPAN", "STRIKE", "STRONG", "STYLE", "SUB", "SUP", "TABLE", "TBODY", "TD", "TEXTAREA", "TFOOT", "TH", "THEAD", "TITLE", "TR", "TT", "U", "UL", "VAR",

};

[0046] 其中，大括号内包含的各个字词(例如"A","ABBR","ACRONYM"等)均为HTML格式的样本所对应的样本目标特征。这些作为样本目标特征的字词通常都是HTML格式的脚本

中的常用字词。

[0047] VBS 格式的样本所对应的样本目标特征可以通过如下数据结构进行定义：

[0048]

```
static char const* g_vbscript_words [] =
{
"Call", "Case", "Class", "Const", "Dim", "Do", "Each", "Else", "Empty",
"Erase", "Error", "Execute", "ExecuteGlobal", "Exit", "Explicit", "False", "For",
"Function", "Get", "If", "Let", "Loop", "Next", "Nothing", "Null", "On", "Option",
"Private", "Property", "Public", "Randomize", "ReDim", "Rem", "Select", "Set",
"Stop", "Sub", "Then", "True", "While" "Wend", "With",
};
```

[0049] 其中，大括号内包含的各个字词（例如 "Call"，"Case"，"Class" 等）均为 VBS 格式的样本所对应的样本目标特征。这些作为样本目标特征的字词通常都是 VBS 格式的脚本中的常用字词。

[0050] 除了上述的字词之外，本领域技术人员还可以根据实际情况增加或删除部分字词，另外，除字词之外，字符串、语句和 / 或标点等其他能够反映脚本特征的信息也可以作为样本目标特征。例如，对于 HTML 类型的样本来说，可以将尖括号作为一个样本目标特征，而对于其他一些类型的样本来说，则可以将小括号作为一个样本目标特征等。

[0051] 然后，对于每个样本，根据该样本的类型确定该样本所对应的各个样本目标特征，并在该样本中（主要是在该样本的可执行代码中）查找并计算每个样本目标特征的出现频率。例如，以 JS 类型的样本为例来说，该样本所对应的样本目标特征即为上述的“static char const*g_javascript_words[]”数据结构中的各个字词，因此，需要在该样本中查找并计算每个样本目标特征的出现频率。例如，假设 "break" 这一字词的出现频率为 3，"case" 这一字词的出现频率为 8，"catch" 这一字词的出现频率为 10 等。其中，如果某一字词在样本中没有出现，则对应的频率为 0。

[0052] 最后，将每个样本中的各个样本目标特征及其出现频率对应存储为一个样本特征向量。也就是说，在每个样本特征向量中，按照各个样本目标特征的顺序存储了每个样本目标特征的出现频率。

[0053] 对于其他类型的各个样本也都可以通过上述方式确定出对应的样本特征向量。通常情况下，一个样本对应一个样本特征向量。在某些特殊情况下，也可能一个样本对应多个样本特征向量：例如，为了更全面地反映出样本特征，因而选取的样本目标特征数量较多且这些样本目标特征分别属于不同的子分类时，可以将每个子分类中的样本目标特征对应存储为一个样本特征向量，比如，可以将由字符串这一子分类构成的样本目标特征存储为一个样本特征向量，将由标点这一子分类构成的样本目标特征存储为另一个样本特征向量等。

[0054] 通过步骤 S110 确定出每个样本的样本特征向量之后，接下来，在步骤 S120 中，通

过预设的分类算法对各个样本的样本特征向量进行计算，得到样本分类模型。

[0055] 其中，分类算法可以灵活选取，只要能够实现分类的目的即可。例如，当预设的分类算法为决策树算法时，得到的样本分类模型为决策树模型；当预设的分类算法为支持向量机 SVM 算法时，得到的样本分类模型为 SVM 模型；或者，当预设的分类算法为贝叶斯算法时，得到的样本分类模型为贝叶斯模型。

[0056] 其中，决策树算法属于机器学习方法，其具有很多优势，例如，构造树的速度快，模式简单便于理解，能够很容易地转换为 SQL 语句以便同数据库进行有效的连接，而且决策树分类模型同其它分类模型相比能获得相似甚至更好的精度。下面以分类算法为决策树算法为例介绍一下决策树模型的一种可能的获取方式：首先，先对部分样本的样本特征向量进行训练，得到待修正的决策树模型；然后，通过模拟测试的方式计算该待修正的决策树模型的精度，如果计算出的精度不满足预设精度，则继续对剩余样本的样本特征向量进行训练，以便在训练过程中继续修正该决策树模型，每次修正决策树模型之后，都重新检查模型精度是否满足预设精度，如果不满足就继续通过训练来修正模型，直到训练后得到的决策树模型满足预设精度为止，将最后得到的满足预设精度的决策树模型作为步骤 S120 中的样本分类模型。

[0057] 其中，由于决策树算法的具体细节以及决策树模型的具体训练方法属于本领域的公知常识，因此不再赘述，本领域技术人员可以根据需要选择各类决策树算法构造各种决策树模型。

[0058] 其中，上述的步骤 S110 和步骤 S120 可以是预先执行的，也就是说，在本发明的方法中，不需要每次都执行上述的步骤 S110 和步骤 S120，而可以预先执行一次步骤 S110 和步骤 S120 并保存得到的样本分类模型，然后利用该模型反复执行步骤 S130 和步骤 S140，以便确定多个待检测脚本的类型，并根据脚本类型来检测病毒。

[0059] 在步骤 S130 中，获取待检测脚本的脚本特征向量，将该脚本特征向量输入上述的样本分类模型，根据输出结果确定该待检测脚本的类型。其中，步骤 S130 也可以通过上述预设的动态库（如 qex.dll）来实现。

[0060] 其中，脚本特征向量的作用在于反映待检测脚本的特征，因此，只要是能够反映出待检测脚本的特征的向量都可以作为脚本特征向量，本发明对脚本特征向量的具体选取方式不做限定。具体实现时，脚本特征向量的选取方式一般与步骤 S110 中的样本特征向量的选取方式相对应。

[0061] 下面给出脚本特征向量的一种可能的确定方式：首先，预先设定所有待检测脚本所对应的至少一个脚本目标特征，其中，每个待检测脚本所对应的脚本目标特征相同。也就是说，每个脚本（无论其为何种类型）都对应着一组或多组脚本目标特征。由此可见，脚本目标特征与步骤 S110 中的样本目标特征存在如下区别：由于样本目标特征是根据样本的类型来确定的，因此不同类型的样本所对应的样本目标特征一般不同；而由于确定脚本目标特征时还无法确定待检测脚本的类型，因此，脚本目标特征并不是根据脚本类型来确定的，所以通常情况下，不同类型的脚本所对应的脚本目标特征相同。另外，为了使脚本目标特征能够全面地反映各类脚本的特征，脚本目标特征可以包括各个类型的样本所对应的全部的样本目标特征。也就是说，当脚本类型主要包括 JS 类型、HTML 类型和 VBS 类型时，脚本目标特征由包含上述的 JS 类型、HTML 类型和 VBS 类型各自对应的样本目标特征的集合构

成。然后，对于每个待检测脚本，在该待检测脚本中（主要是在该脚本的可执行代码中）查找并计算每个脚本目标特征的出现频率。最后，将每个待检测脚本中的各个脚本目标特征及其出现频率对应存储为一个脚本特征向量。与样本特征向量类似，通常情况下，一个脚本对应一个脚本特征向量。在某些特殊情况下，也可能一个脚本对应多个脚本特征向量：例如，为了更全面地反映出脚本特征，因而选取的脚本目标特征数量较多且这些脚本目标特征分别属于不同的子分类时，可以将每个子分类中的脚本目标特征对应存储为一个脚本特征向量，比如，可以将由字符串这一子分类构成的脚本目标特征存储为一个脚本特征向量，将由标点这一子分类构成的脚本目标特征存储为另一个脚本特征向量等。

[0062] 当要判断一个脚本的类型时，将该脚本对应的脚本特征向量输入步骤 S120 中得到的样本分类模型，即可根据输出结果确定该待检测脚本的类型。

[0063] 例如，以样本分类模型为决策树模型为例来说，将待检测脚本的脚本特征向量输入到决策树模型之后，首先根据决策树模型的根节点中定义的特征来初步划分待检测脚本的类型，假设共有 10 个类型的脚本，符合根节点中定义的特征的脚本属于前 5 个类型的脚本，而不符合根节点中定义的特征的脚本属于后 5 个类型的脚本，接下来，进一步在相应的类型范围内根据叶子节中定义的特征来逐步缩小待检测脚本的类型范围，直至判断出待检测脚本的准确类型为止。

[0064] 通过上述步骤就可以判断出待检测脚本的类型。另外，在上文描述的获取样本特征向量（或脚本特征向量）的实现过程中，主要是通过在样本（或脚本）的可执行代码中查找样本目标特征（或脚本目标特征）来实现的。可选地，为了提高在可执行代码中查找样本目标特征（或脚本目标特征）的效率，可以预先对样本（或脚本）的可执行代码进行预处理：例如，可以对可执行代码进行大小写分析，将其中的大小写字母统一转换为大写（或小写）字母，以方便后续过程的处理（此方式尤其适用于一些不区分大小写的语言所编写的脚本）；可以预先去掉可执行代码中重复的路径符号和参数等，以简化处理过程。

[0065] 通过上述步骤 S130 判断出待检测脚本的类型之后，在步骤 S140 中，根据确定出的待检测脚本的类型，将待检测脚本提供给该类型所对应的脚本处理引擎，由该类型所对应的脚本处理引擎检测所述待检测脚本中是否携带病毒。在本实施例中，步骤 S140 也可以通过上述预设的动态库（如 qex.dll）来实现。

[0066] 其中，脚本处理引擎的数量为多个，每个脚本处理引擎对应至少一个脚本类型，用于根据预设的至少一个脚本类型对应的病毒样本特征来检测属于该至少一个脚本类型的待检测脚本中是否携带病毒。

[0067] 通常情况下，脚本类型与脚本处理引擎是一一对应的关系，即：一个类型的脚本对应着一个脚本处理引擎。例如，JS 类型的脚本对应着 JS 脚本处理引擎，VBS 类型的脚本对应着 VBS 脚本处理引擎，HTML 类型的脚本对应着 HTML 脚本处理引擎等。

[0068] 举例来说，在步骤 S140 中，假设判断出的待检测脚本为 JS 类型，则将该脚本提供给 JS 脚本处理引擎进行处理。其中，脚本在 JS 脚本处理引擎中的处理逻辑是预先根据 JS 类型的脚本的病毒特点进行设定的：例如，可以预先对多个携带病毒的 JS 脚本样本进行分析，确定出 JS 类型的脚本的病毒样本特征，其中，JS 类型的病毒样本特征可以包括任何能够反映 JS 类型的脚本的病毒特点的特征，例如字符特征和 / 或行为特征等。然后，对待检测脚本进行词法分析及语法分析，得到待检测脚本的虚拟脚本集合，并据此对该脚本进行

虚拟执行，并判断虚拟执行的结果是否符合预设的病毒样本特征，如果符合，则确定脚本带有病毒，反之，则确定脚本没有病毒。例如，当病毒样本特征为字符特征时，可以将虚拟执行的脚本语句转换为字符串形式，并判断字符串形式的脚本语句是否包括该字符特征，当判断结果为是时，确定脚本带有病毒；当病毒样本特征为行为特征时，可以对虚拟执行的执行行为进行分析，并判断该执行行为是否包括该行为特征，当判断结果为是时，确定脚本带有病毒。

[0069] 除了上面介绍的通过词法分析、语法分析以及虚拟执行的方式来判断脚本是否带有病毒之外，本领域技术人员还可以采用其它多种方式来判断：例如，可以预先获取到病毒行为特征后，通过 HOOK 机制来监测脚本是否执行了该病毒行为，从而判断脚本是否带有病毒。

[0070] 上面以 JS 脚本处理引擎为例介绍了 JS 类型的脚本的病毒检测过程，对于其他的脚本处理引擎的处理过程与之类似，此处不再赘述。总之，在步骤 S140 的处理过程中，可以根据脚本类型的特点进行有针对性的分析。

[0071] 另外，除了使一个类型的脚本对应一个脚本处理引擎之外，在多个类型的脚本病毒具有相似特征的情况下，还可以使具有相似特征的多个类型的脚本对应同一个脚本处理引擎，例如，使 A、B 两个类型的脚本（A、B 类型脚本的病毒特征相似）对应第一脚本处理引擎，使 C 这一类型的脚本对应第二脚本处理引擎，从而缩减处理引擎的数量，简化处理流程。

[0072] 进一步地，本发明提供的上述基于脚本类型判断的病毒检测方法可以主要应用在杀毒装置中。具体应用时，可以预先对上文提到的病毒样本特征保存到预设的病毒特征库中。该病毒特征库可以设置在客户端本地或云端服务器上（例如保存在云端的数据库中，或者保存在本地的配置文件中），用于存储病毒文件的病毒特征，该病毒文件及其病毒特征例如可以通过 md5 值等进行标识。然后，在步骤 S140 中，杀毒装置（例如安全卫士软件等）通过进程间通信的机制先调用本发明实施例中提供的脚本处理引擎对待检测脚本进行查毒，由脚本处理引擎根据保存的病毒样本特征进行病毒检测：如果检测出了病毒，可以直接将该病毒删除，或者将该病毒提示给用户，由用户来决定是否删除；如果未检测出病毒，可以进一步调用其他杀毒引擎（例如云查杀引擎等）进行查杀。或者，也可以先调用其他杀毒引擎进行查杀，然后再调用本发明实施例中提供的脚本处理引擎进行查杀。

[0073] 下面以后一种方式为例介绍一下杀毒装置的工作过程。在下文中，为了便于描述，将包含云查杀引擎的其他杀毒引擎称作第一杀毒引擎，将本发明实施例中提供的脚本处理引擎称作第二杀毒引擎，则杀毒装置在执行步骤 S140 时可通过如下方式实现：

[0074] 首先，获取待查杀文件中预设数量的文件。其中，预设数量的文件可以是通过上述的步骤 S130 确定出类型的待检测脚本文件。

[0075] 然后，通过至少一个第一杀毒引擎对该预设数量的文件中的第一分类文件进行扫描，得到包括预设数量的文件中的确定文件的第一扫描结果。其中，第一分类文件优选为 PE 类型文件；至少一个第一杀毒引擎包括：用于查杀 PE（Portable Execute，可移植执行体）类型文件的云查杀引擎，和 / 或 QVM（Qihoo Virtual Machine，人工智能引擎）引擎。其中，由于已经通过步骤 S110 至步骤 S130 确定出了预设数量的文件的类型，因而，第一杀毒引擎可以直接利用确定出的文件类型判断其是否属于第一分类文件。在本申请实施例中，可以

利用第一杀毒引擎和第二杀毒引擎实现并行杀毒过程，其中，并行杀毒过程具体是指：当第一杀毒引擎在查杀过程中，可以将已查杀过的文件中的未确定文件输入到第二杀毒引擎中进行查杀，而不必等到第一杀毒引擎查杀完所有待查杀文件，再由第二杀毒引擎进行查杀。同理；如果第一杀毒引擎至少有两个，则至少两个第一杀毒引擎之间的查杀过程也采用前述并行查杀的方式。

[0076] 接下来，将上述预设数量的文件中除确定文件以外的其它文件输入到至少一个第二杀毒引擎，通过第二杀毒引擎对待查杀文件中除第一分类文件中的确定文件以外的其它文件进行扫描，获得第二扫描结果。其中，第二杀毒引擎主要指对除第一分类文件外的其它文件进行扫描的杀毒引擎，需要说明的是，该第二杀毒引擎可以具有对所有分类文件进行查杀的能力，本实施例通过并行查杀的方式减少每一种杀毒引擎的查杀数量，从而提高查杀速度，以便有效利用系统资源。本实施例中第二杀毒引擎可以包括至少一个脚本处理引擎(例如 JS 脚本处理引擎、VBS 脚本处理引擎和 HTML 脚本处理引擎等)。当第二杀毒引擎为多个时，多个第二杀毒引擎之间也是相互并行工作的，由此可以大幅提高查杀效率。

[0077] 具体地，第一杀毒引擎和第二杀毒引擎之间的并行查杀过程可以具体描述如下：顺序获取待查杀文件中预设数量的文件，通过第一杀毒引擎对预设数量的文件中的第一分类文件进行扫描，得到包括预设数量的文件中的确定文件的第一扫描结果，本实施例中的确定文件主要指可以由第一文件确定的恶意文件和 / 或非恶意文件；将预设数量的文件中除确定文件以外的其它文件输入到第二杀毒引擎，由第二杀毒引擎进行扫描；当未扫描完所有待查杀文件时，返回顺序获取待查杀文件中预设数量的文件的步骤，直至通过第一杀毒引擎扫描完所有待查杀文件。

[0078] 最后，将查毒结果通过客户端的电脑反馈给用户。

[0079] 本发明提供的上述杀毒方式通过两种杀毒引擎可以提高查杀的全面性和有效性，该方式尤其适用于通过第二杀毒引擎检测非 PE 类型的文件，并且，可以有效地配合多个杀毒引擎进行并行或者是串行的查杀，从而提高查杀效率。另外，由于采用并行杀毒的方式，可以根据不同杀毒引擎的查杀特点，通过第一杀毒引擎对第一分类文件进行查杀，第一杀毒引擎无法查杀的文件再通过第二杀毒引擎进行查杀，因此可以有效利用系统资源，使得多个杀毒引擎不会重复对同一文件进行扫描。

[0080] 为了便于理解本发明中的杀毒过程的具体实现方式，下面结合一个具体应用的实例对本发明中的杀毒过程加以介绍：

[0081] 步骤 S1，安装在电脑上的监控装置的实时防护功能开启，监控装置的文件访问驱动模块监控对电脑的操作系统中任何文件的访问；

[0082] 步骤 S2，文件访问驱动模块在某一时刻监视发现某程序正在对操作系统中的一文件 A 进行访问。

[0083] 此时，安装在电脑上的其他监控装置也监控到了某程序正在对所述文件 A 进行访问。

[0084] 步骤 S3，文件访问驱动模块首先检查文件 A 是否记录在数据表中，如果没有记录，则说明是杀毒软件第一次发现该文件 A 被访问，将该文件 A 的标识和地址记录到数据表中，放行所述程序对文件 A 的访问行为；

[0085] 数据表中保存有文件 A 的标识、地址和文件的安全属性；文件访问驱动模块在此

步骤所记录的是该文件 A 的标识和地址，此时安全属性为空白；

[0086] 步骤 S4，扫描模块周期性轮询数据表，发现数据表新增加了文件 A 的标识后，将文件 A 的标识添加到自己的扫描队列中；

[0087] 扫描模块轮询数据表的周期长度的设置要至少保证安装在电脑上的其他监控装置可以对文件 A 进行一次扫描；

[0088] 在实际操作中，往往是在一个周期内或者实时对多个文件的访问行为进行监控，所以扫描模块会通过多个文件被添加的顺序组织扫描队列，根据扫描队列排列管理文件的标识，并依据扫描队列完成对文件的扫描。

[0089] 在上述周期内，电脑上的其他监控装置应该已经对文件 A 完成扫描，可能文件 A 被其他监控装置扫描后认为是恶意程序，在经过用户许可后被其他监控装置清除，也可能文件 A 被其他监控装置扫描后认为安全，而对访问进行放行；此时不考虑其他监控装置的处理结果，监控装置依然对文件 A 进行扫描。

[0090] 在上述实例中，判断文件是否安全时，可通过如下方式实现：分析每个程序文件，从程序文件中抽取预先定义的特征，根据所抽取的特征生成特征向量，以及每个特征向量的黑白属性，根据已知编译器的入口指令序列判定编译生成相应程序的编译器类型。

[0091] 例如，不同的特征分类中包含不同数量的具体特征，以特征分类是编译器为例，其中可以具体包括的编译器特征为：VC4、VC5、VC6、VC7、VC8、Delphi、BC。本申请实施例中，可以为每一个特征分类分配一个分类标识，例如，编译器的分类标识为“1”，对于具体的每个编译器特征，可以为其进一步分配特征标识，例如，VC4 的特征标识为“1”、VC5 的特征标识为“2”、VC6 的特征标识为“3”、VC7 的特征标识为“4”、VC8 的特征标识为“5”、Delphi 的特征标识为“6”、BC 的特征标识为“7”。

[0092] 则在根据所抽取的特征生成特征向量时，特征向量中的每一个特征的数组都用其分类标识和特征标识进行表征，例如，所抽取的特征为编译器特征“VC5”，则其对应的分类标识为“1”，特征标识为“2”，因此特征向量中对应该“VC5”的信息表示为“1:2”；同理，属于其它特征分类的具体特征也用上述形式表示，如下所示，为从某个程序中提取到了 4 个特征的特征向量示例：1:02:121100:123456785000:365。

[0093] 特征向量的黑白属性用于表示包含该特征向量中的特征的程序属于恶意程序还是非恶意程序，其中属性为“白”，则对应非恶意程序，属性为“黑”，则对应恶意程序；进一步，可以为白属性定义标识为“0”，黑属性定义标识为“1”。则在为每个程序生成特征向量后，可以根据特征向量包含的信息为其分配属性标识，例如，为上述特征向量“1:02:121100:123456785000:365”分配属性标识为白属性“0”，则相应的信息可以表示为“01:02:121100:123456785000:365”。其中，非 PE 文件即为所输入的未知程序文件，根据特征分类的不同包含了 k 个决策机，以及对应 k 个决策机的 k 个训练模型。分析非 PE 文件后，抽取相应的特征，将所抽取的特征放入一个相应的特征向量之内，根据已经抽取到的特征，进行特征分类，例如，可以依据类别将特征分成 UPX、NSPack、ASPack、UPack、PECompact 等，或者，据编译器的类型可以分为 VC4、VC5、VC6、VC7、VC8、Delphi、BC 等，根据分类的结果，使用不同的决策机和训练模型进行相应的判断，根据相应决策机和模型得出的判断结果，依分类的权重加权得到评分结果，由评分结果确定该文件是否是恶意程序或正常程序。

[0094] 例如，假设决策机一共有 k 个，分类一共有 m 种，分别为分类 1,2,…,m，第 i 种

分类预先设定的权重是(wi1,wi2,…,wik)，则相应的样本类别 i 的决策机判别的结果是(ri1,ri2,…,rik)，由此得到的综合结果为(wi1,wi2,…,wik)*(ri1,ri2,…,rik)。可以预先设置一个结果判断阈值，当判断结果小于该阈值则确定未知程序为非恶意程序，当判断结果大于该阈值，则可以确定未知程序为恶意程序。

[0095] 综上所述，通过本发明实施例提供的方法，可以直接根据预先确定好的样本分类模型来快速判断出脚本的类型，无需对每个脚本都进行人工分析，由此大大提高了判断脚本类型的效率，而且，由于避免了人工操作的失误因而还大幅提高了判断的准确率；另外，正是由于本发明在判断脚本类型方面有着快速准确等诸多优势，因而才使得该脚本类型的判断方法能够很方便地应用到脚本病毒检测领域，从而能够根据脚本的类型，对该类型的脚本所容易感染的病毒加以识别并防范，由于脚本病毒的具体形式或行为特征往往与脚本类型有关，因此根据脚本类型进行病毒检测能够大幅提高检测效率及准确度。而且，在本发明中，还可以通过多个杀毒引擎并行查杀的方式来大幅提高杀毒效率。

[0096] 图 2 示出了本发明实施例提供的基于脚本类型判断的病毒检测装置的结构图，如图 2 所示，该装置包括：获取单元 21、模型生成单元 22、脚本判断单元 23 以及病毒检测单元 24。

[0097] 其中，获取单元 21 预先获取预设数量的脚本作为样本。具体地，获取单元 21 可以通过多种方式来获取作为脚本的样本。例如，可以预先通过多台虚拟机并行运行的方式来获取样本。其中，每台虚拟机可以运行多台 xp 对应的实体机，由此可以提升样本收集的效率。样本的数量可以根据实际情况来确定，例如，精确要求越高的情况下所需的样本数量也越多，反之样本数量则越低。

[0098] 模型生成单元 22 根据样本的类型确定每个样本的样本特征向量，并通过预设的分类算法对各个样本的样本特征向量进行计算，得到样本分类模型。

[0099] 可选地，上述模型生成单元 22 进一步包括：第一设定子单元，适于根据样本的类型，分别设定各个类型的样本所对应的至少一个样本目标特征；第一查找子单元，适于对于每个样本，根据该样本的类型确定该样本所对应的各个样本目标特征，并在该样本中查找并计算每个样本目标特征的出现频率；第一存储子单元，适于将每个样本中的各个样本目标特征及其出现频率对应存储为一个样本特征向量。其中，样本目标特征包括：字符串、字词、语句和 / 或标点。

[0100] 脚本判断单元 23 获取待检测脚本的脚本特征向量，将脚本特征向量输入样本分类模型，根据输出结果确定待检测脚本的类型。

[0101] 可选地，上述脚本判断单元进一步包括：第二设定子单元，适于预先设定所有待检测脚本所对应的至少一个脚本目标特征，其中，每个待检测脚本所对应的脚本目标特征相同；第二查找子单元，适于对于每个待检测脚本，在该待检测脚本中查找并计算每个脚本目标特征的出现频率；第二存储子单元，适于将每个待检测脚本中的各个脚本目标特征及其出现频率对应存储为一个脚本特征向量。其中，脚本目标特征包括：各个类型的样本所对应的样本目标特征。

[0102] 其中，当上述分类算法为决策树算法时，样本分类模型为决策树模型；当上述分类算法为支持向量机 SVM 算法时，样本分类模型为 SVM 模型；或者，当上述分类算法为贝叶斯算法时，样本分类模型为贝叶斯模型。

[0103] 例如，当上述分类算法为决策树算法，所述样本分类模型为决策树模型时，所述模型生成单元用于：先对部分样本的样本特征向量进行训练，得到待修正的决策树模型；当判断出所述待修正的决策树模型不满足预设精度时，继续对剩余样本的样本特征向量进行训练，直到训练后得到的决策树模型满足预设精度。

[0104] 病毒检测单元 24 根据确定出的待检测脚本的类型，将待检测脚本提供给该类型所对应的脚本处理引擎，由脚本处理引擎检测待检测脚本中是否携带病毒。其中，脚本处理引擎的数量为多个，每个脚本处理引擎对应至少一个脚本类型，用于根据预设的至少一个脚本类型对应的病毒样本特征来检测属于该至少一个脚本类型的待检测脚本中是否携带病毒。

[0105] 在上述实施例中，模型生成单元 22 及其包含的第一设定子单元、第一查找子单元及第一存储子单元的具体工作方式可参照方法实施例中步骤 S120 的描述；脚本判断单元 23 及其包含的第二设定子单元、第二查找子单元及第二存储子单元的具体工作方式可参照方法实施例中步骤 S130 的描述，病毒检测单元 24 的具体工作方式可参照方法实施例中步骤 S140 的描述，此处不再赘述。

[0106] 在本发明实施例提供的基于脚本类型判断的病毒检测方法及装置中，预先根据样本类型确定出各个样本的样本特征向量，并据此得到样本分类模型，该模型用于对脚本进行分类，由此，在需要判断脚本类型时，只需获取脚本的脚本特征向量，并根据预先得到的样本分类模型就可以完成脚本类型的判断。由此解决了现有技术中由人工分析脚本类型所导致的耗费时间和精力，以及由此所导致的不便于将脚本类型判断的方式应用于脚本病毒检测领域的技术问题，实现了能够便捷高效地识别脚本类型，并根据脚本的类型进行有针对性的病毒检测，以便提高检测效率和准确度的技术效果。

[0107] 在此提供的算法和显示不与任何特定计算机、虚拟系统或者其它设备固有相关。各种通用系统也可以与基于在此的示教一起使用。根据上面的描述，构造这类系统所要求的结构是显而易见的。此外，本发明也不针对任何特定编程语言。应当明白，可以利用各种编程语言实现在此描述的本发明的内容，并且上面对特定语言所做的描述是为了披露本发明的最佳实施方式。

[0108] 在此处所提供的说明书中，说明了大量具体细节。然而，能够理解，本发明的实施例可以在没有这些具体细节的情况下实践。在一些实例中，并未详细示出公知的方法、结构和技术，以便不模糊对本说明书的理解。

[0109] 类似地，应当理解，为了精简本公开并帮助理解各个发明方面中的一个或多个，在上面对本发明的示例性实施例的描述中，本发明的各个特征有时被一起分组到单个实施例、图、或者对其的描述中。然而，并不应将该公开的方法解释成反映如下意图：即所要求保护的本发明要求比在每个权利要求中所明确记载的特征更多的特征。更确切地说，如下面的权利要求书所反映的那样，发明方面在于少于前面公开的单个实施例的所有特征。因此，遵循具体实施方式的权利要求书由此明确地并入该具体实施方式，其中每个权利要求本身都作为本发明的单独实施例。

[0110] 本领域那些技术人员可以理解，可以对实施例中的设备中的模块进行自适应性地改变并且把它们设置在与该实施例不同的一个或多个设备中。可以把实施例中的模块或单元或组件组合成一个模块或单元或组件，以及此外可以把它们分成多个子模块或子单元或

子组件。除了这样的特征和 / 或过程或者单元中的至少一些是相互排斥之外，可以采用任何组合对本说明书（包括伴随的权利要求、摘要和附图）中公开的所有特征以及如此公开的任何方法或者设备的所有过程或单元进行组合。除非另外明确陈述，本说明书（包括伴随的权利要求、摘要和附图）中公开的每个特征可以由提供相同、等同或相似目的的替代特征来代替。

[0111] 此外，本领域的技术人员能够理解，尽管在此所述的一些实施例包括其它实施例中所包括的某些特征而不是其它特征，但是不同实施例的特征的组合意味着处于本发明的范围之内并且形成不同的实施例。例如，在下面的权利要求书中，所要求保护的实施例的任意之一都可以以任意的组合方式来使用。

[0112] 本发明的各个部件实施例可以以硬件实现，或者以在一个或者多个处理器上运行的软件模块实现，或者以它们的组合实现。本领域的技术人员应当理解，可以在实践中使用微处理器或者数字信号处理器（DSP）来实现根据本发明实施例的装置中的一些或者全部部件的一些或者全部功能。本发明还可以实现为用于执行这里所描述的方法的一部分或者全部的设备或者装置程序（例如，计算机程序和计算机程序产品）。这样的实现本发明的程序可以存储在计算机可读介质上，或者可以具有一个或者多个信号的形式。这样的信号可以从因特网网站上下载得到，或者在载体信号上提供，或者以任何其他形式提供。

[0113] 应该注意的是上述实施例对本发明进行说明而不是对本发明进行限制，并且本领域技术人员在不脱离所附权利要求的范围的情况下可设计出替换实施例。在权利要求中，不应将位于括号之间的任何参考符号构造成对权利要求的限制。单词“包含”不排除存在未列在权利要求中的元件或步骤。位于元件之前的单词“一”或“一个”不排除存在多个这样的元件。本发明可以借助于包括有若干不同元件的硬件以及借助于适当编程的计算机来实现。在列举了若干装置的单元权利要求中，这些装置中的若干个可以是通过同一个硬件项来具体体现。单词第一、第二、以及第三等的使用不表示任何顺序。可将这些单词解释为名称。

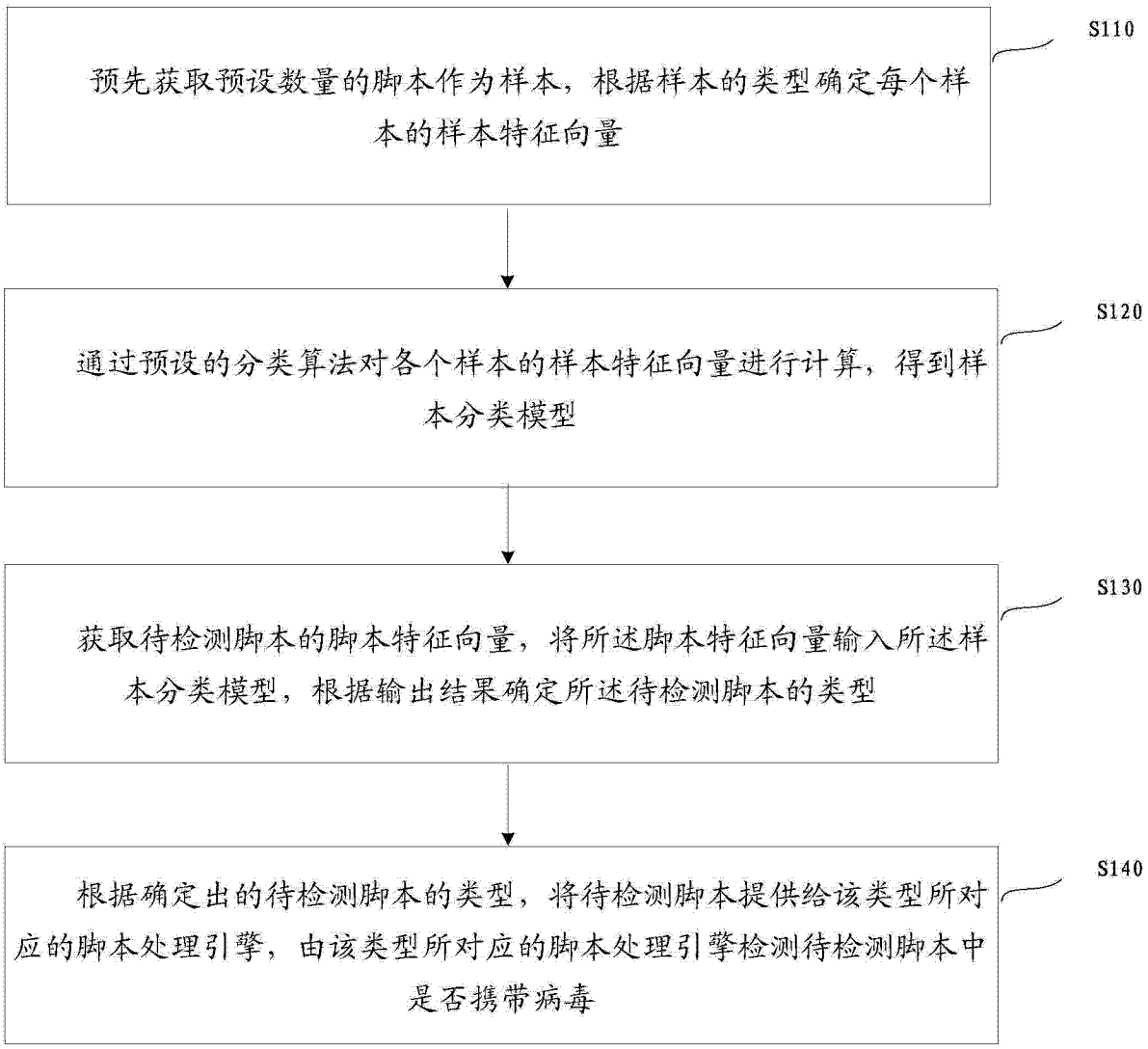

S110
预先获取预设数量的脚本作为样本，根据样本的类型确定每个样本的样本特征向量
S120
通过预设的分类算法对各个样本的样本特征向量进行计算，得到样本分类模型
S130
获取待检测脚本的脚本特征向量，将所述脚本特征向量输入所述样本分类模型，根据输出结果确定所述待检测脚本的类型
S140
根据确定出的待检测脚本的类型，将待检测脚本提供给该类型所对应的脚本处理引擎，由该类型所对应的脚本处理引擎检测待检测脚本中是否携带病毒

图 1

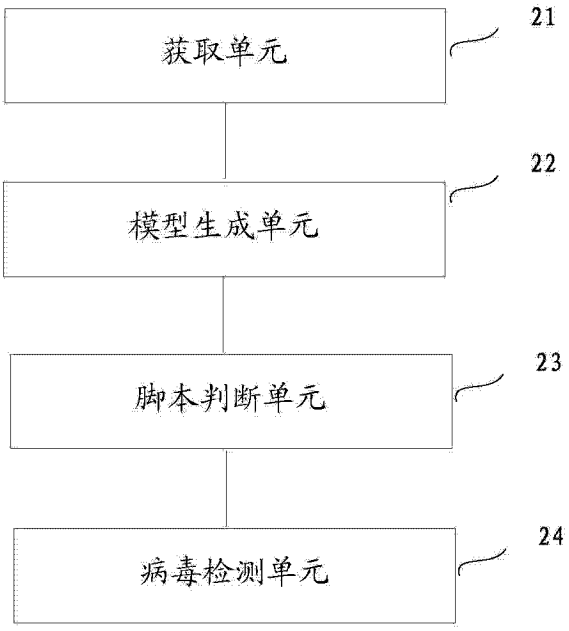


图 2