(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 109344614 B
(45)授权公告日 2021.04.20

(21)申请号 201810810463.2

(22)申请日 2018.07.23

(65)同一申请的已公布的文献号
申请公布号 CN 109344614 A

(43)申请公布日 2019.02.15

(73)专利权人 厦门大学
地址 361000 福建省厦门市思明区思明南路422号

(72)发明人 冯超 李汉波 黄联芬 叶超林 林英 叶国华 吴卫东 王威

(74)专利代理机构 厦门原创专利事务所(普通合伙) 35101
代理人 陈建华

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

(56)对比文件
US 2017154182 A1,2017.06.01
US 2018150724 A1,2018.05.31
陈建民.面向移动应用安全评估的多属性专家决策模型及应用研究.《中国博士学位论文全文数据库》.2016,

审查员 李婧雯

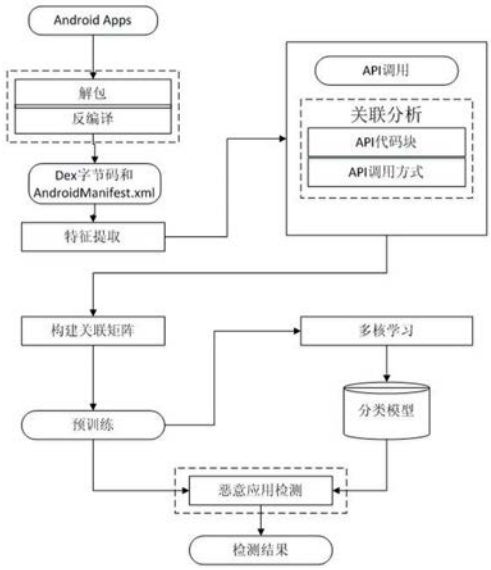
权利要求书1页 说明书6页 附图2页

(54)发明名称

一种Android恶意应用在线检测方法

(57)摘要

本发明提供一种Android恶意应用在线检测方法,在检测Android恶意软件的过程中,使用API函数字符串,提取8组特征信息,并映射为特征向量,而特征向量采用稀疏表示的形式;并且进一步分析API之间的不同关系并创建更高层次的关联分析;以图的方式来表示相关API作为结构化程序之间的关系;将API字符特征与关系图构成特征矩阵;采用多核学习方法训练出分类模型;部署在通用的Web架构中,实现Android应用软件的在线检测。本发明具有良好的分类效果,并且使用方便,快捷。



CN 109344614 B

1.一种Android恶意应用在线检测方法,其特征在于,包括以下步骤:

1)用于判断恶意应用的特征包含以下几类:

S1:硬件类:APP在运行时所需要的硬件信息,Camera,Touchscreen,GPS;

S2:权限请求类:由于权限是Android里最重要的安全机制,恶意代码在运行时经常会请求权限,如SEND_SMS Permission;

S3:APP组件类:每个APP都包含4组:Activities,Services,Content Providers,Broadcast Receivers;

S4:Intents类:Android下的Inter-Process以及Intra-Process通信;

2)反汇编字节码提取下列特征信息:

S5:受限API调用类:Android权限系统限制访问一系列关键的API调用;搜索这些调用发生时的反汇编代码,没有申请权限就调用关键API,正在使用ROOT以绕过Android平台限制恶意行为;

S6:已用权限类:包含了已请求的权限和正在执行的权限,并把API和其申请的权限对应匹配;

S7:可疑API调用类:某些API要用敏感数据和资源,包括getDeviceId()、getSubscriberId()、SetWifiEnabled()、execHttpRequest()、sendTextMessage()、Runtime.exec()、Cipher.getInstance();

S8:网络地址类:IP地址,Hostnames和URL;

3)以上特征构建出一个特征向量S,S=S1∪S2....∪S8,每一个Android应用x都要映射到这S中,$\varphi: X \to \{0,1\}^{|S|}, \varphi(x) \to (I(x,s))_{s \in S}$,其中I(x,s)表示为:$I(x,s) = \begin{cases} 1, & x\text{包含特征}s \\ 0, & x\text{不包含特征}s \end{cases}$

4)连续出现多个具有相同行为意图但是又调用的是不同的API,这一块代码就具有恶意行为的特征;这些特征API共存同一代码块的关系即可表述为高层次的关联分析;这些关联特征可以构造关联特征矩阵:矩阵A,元素a_{ij},描述:app_i包含API_j,则$a_{ij}=1$,否则$a_{ij}=0$;矩阵B,元素b_{ij},描述:API_i和API_j共存于同一代码块,则$b_{ij}=1$,否则$b_{ij}=0$;矩阵I,元素i_{ij},描述:API_i和API_j使用相同的调用方式,则$i_{ij}=1$,否则$i_{ij}=0$;上述各个矩阵的组合,用图论建立模型,即一个包含API之间关系的APP模块可以描述为G=(v,ε),v表示各个软件代码实体函数或参数,ε表示他们之间的关联;

5)在构建了特征空间后,采用监督学习的方式对已有的Android APP库进行训练学习;在原有的高维特征空间的基础上,进行多核学习进行特征聚合;从而达到更好的分类效果;

6)用户在目标网页上传待检测Android应用的APK文件;Web服务器把接收到的客户端请求转发给Flask程序实例;Flask程序首先对用户上传的文件进行文件类型检测,判断是否为有效的APK文件;对待检测应用进行静态分析、特征提取、特征分析及向量化;对待检测应用进行分类判别,得到分类结果及高权重特征信息;检测相应结果;Flask程序在相应网页上显示该应用的检测结果,包括对该应用是否为恶意应用的判定,检测过程中权值较高的特征以及其他特殊信息。

一种Android恶意应用在线检测方法

技术领域

[0001] 本发明涉及一种Android恶意应用在线检测方法。

背景技术

[0002] Android智能终端已被广泛应用于人们的日常生活中,如网络支付,智能家居等。不断扩展的功能,智能终端的使用近年来经历了指数级增长。但是,由于Android开放源代码开发的生态系统和较大的市场份额,Android开发人员不仅制作合法的Android应用程序而且还会传播恶意应用软件,故意将恶意行为付诸于智能终端用户。由于缺乏可靠的审查方法,开发者可以在谷歌Android市场上传他们的Android应用程序,其中甚至包含勒索病毒或特洛伊木马。这对智能终端用户构成严重威胁,尤其是移动支付用户,个人的隐私信息都有可能被窃取。通常使用基于签名的方法来识别威胁。然而,攻击者可以使用如代码混淆,重新打包等技术轻易躲避检测。而攻击者日趋增强的反检测意识带来的是日益复杂的Android恶意软件,这时就需要新的检测技术保护用户免受新的恶意应用带来的新威胁。传统的基于API的检测方法只是表面上的API字符串构建特征向量,而忽视了API之间的关联这一更高层次的分析。

发明内容

[0003] 本发明的目的,是要提供一种Android恶意应用在线检测方法,在检测Android恶意软件的过程中,使用API函数字符串,提取8组特征信息,并映射为特征向量,而特征向量采用稀疏表示的形式;并且进一步分析API之间的不同关系并创建更高层次的关联分析;以图的方式来表示相关API作为结构化程序之间的关系;将API字符特征与关系图构成特征矩阵;采用多核学习方法训练出分类模型;部署在通用的Web架构中,实现Android恶意应用的在线检测。

[0004] 本发明是这样实现的,所述一种Android恶意应用在线检测方法,包括以下步骤:

[0005] 1)用于判断恶意应用的特征包含以下几类:

[0006] S1:硬件类:APP在运行时所需要的硬件信息,Camera,Touchscreen,GPS;

[0007] S2:权限请求类:由于权限是Android里最重要的安全机制,恶意代码在运行时经常会请求权限,如SEND_SMS Permission;

[0008] S3:APP组件类:每个APP都包含4组:Activities,Services,Content Providers,Broadcast Receivers;

[0009] S4:Intents类:Android下的Inter-Process以及Intra-Process通信;

[0010] 2)反汇编字节码提取下列特征信息:

[0011] S5:受限API调用类:Android权限系统限制访问一系列关键的API调用;搜索这些调用发生时的反汇编代码,没有申请权限就调用关键API,正在使用ROOT以绕过Android平台限制恶意行为;

[0012] S6:已用权限类:包含了已请求的权限和正在执行的权限,并把API和其申请的权

限对应匹配；

[0013] S7：可疑API调用类：某些API要用敏感数据和资源，包括getDeviceId()、getSubscriberId()、SetWifiEnabled()、execHttpRequest()、sendTextMessage()、Runtime.exec()、Cipher.getInstance()；

[0014] S8：网络地址类：IP地址，Hostnames和URL；

[0015] 3)以上特征构建出一个特征向量S，S＝S1∪S2....∪S8，每一个Android应用x都要映射到这S中，$\varphi: X \to \{0,1\}^{|s|}, \varphi(x) \to (I(x,s))_{s \in S}$，其中I(x，s)表示为：

$$I(x,s) = \begin{cases} 1, & x\text{包含特征}s \\ 0, & x\text{不包含特征}s \end{cases}$$

[0016] 4)连续出现多个具有相同行为意图但是又调用的是不同的API，这一块代码就具有恶意行为的特征；这些特征API共存同一代码块的关系即可表述为高层次的关联分析；这些关联特征可以构造关联特征矩阵：矩阵A，元素a_{ij}，描述：app_i包含API_j，则a_{ij}＝1，否则a_{ij}＝0；矩阵B，元素b_{ij}，描述：API_i和API_j共存于同一代码块，则b_{ij}＝1，否则b_{ij}＝0；矩阵I，元素i_{ij}，描述：API_i和API_j使用相同的调用方式，则i_{ij}＝1，否则i_{ij}＝0；上述各个矩阵的组合，用图论建立模型，即一个包含API之间关系的APP模块可以描述为G＝(v,ε)，v表示各个软件代码实体函数或参数，ε表示他们之间的关联；

[0017] 5)在构建了特征空间后，采用监督学习的方式对已有的Android APP库进行训练学习；在原有的高维特征空间的基础上，进行多核学习进行特征聚合；从而达到更好的分类效果；

[0018] 6)用户在目标网页上传待检测Android应用的APK文件；Web服务器把接收到的客户端请求转发给Flask程序实例；Flask程序首先对用户上传的文件进行文件类型检测，判断是否为有效的APK文件；对待检测应用进行静态分析、特征提取、特征分析及向量化；对待检测应用进行分类判别，得到分类结果及高权重特征信息；检测相应结果；Flask程序在相应网页上显示该应用的检测结果，包括对该应用是否为恶意应用的判定，检测过程中权值较高的特征以及其他特殊信息。

[0019] 本发明的有益效果是，在API字符串的基础上，以及他们的调用关系共同构建出特征，并映射到特征空间；再通过Android应用数据集进行多核学习，训练出各个权重系数从而确定模型。为了方便使用，利用Python Flask Web框架搭建Android恶意应用检测系统服务器，将训练好的检测模型部署在该服务器上，用户可以通过Web页面根据提示上传Android应用软件APK，服务器后台对应用进行分析并将分析结果返回至Web页面。分析结果包括对该应用是否为恶意应用的判定，检测过程中权值较高的特征以及其他特殊信息。本发明具有良好的分类效果，并且使用方便，快捷。

附图说明

[0020] 图1是本发明的总体结构图。

[0021] 图2是本发明的Web架构示意图。

具体实施方式

[0022] 本发明所述一种Android恶意应用在线检测方法，如图1、2所示，在Android应用软

件的dex字节码和AndroidManifest.xml中收集特征,这些特征主要是API函数字符串的集合,主要包含以下几类:

[0023] S1:硬件类:APP在运行时所需要的硬件模块信息,如Camera,Touchscreen,GPS等。由于硬件请求具备一定的安全风险,如接入GPS和Network,它能够把私人位置信息通过网络发送给攻击者,所以硬件类特征是特征之一;

[0024] S2:权限请求类:由于权限是Android里最重要的安全机制,恶意代码在运行时经常会请求权限,如SEND_SMS Permission;

[0025] S3:APP组件类:每个APP都包含4组:Activities,Services,Content Providers,Broadcast Receivers;每一个APP都可以声明若干组件,这些组件的名称也是作为一类特征;

[0026] S4:Intents类:Android下的Inter-Process以及Intra-Process通信,都是靠Intent;一些恶意代码经常监听某些特定的Intent,可作为一类特征;

[0027] Android APP用Java语言编写,并且编译成优化的字节码,这些字节码可以被反汇编,从而可以从中提取特征信息;

[0028] S5:受限API调用类:Android权限系统限制访问一系列关键的API调用,搜索这些调用发生时的反汇编代码,可以揭露恶意行为,没有申请权限就调用关键API,这可能表明了恶意软件正在使用ROOT以绕过Android平台限制;

[0029] S6:已用权限类:包含了已请求的权限和正在执行的权限,并把API和其申请的权限对应匹配;

[0030] S7:可疑API调用类:某些API要用敏感数据和资源,这些都是恶意代码经常用的,包括getDeviceId()、getSubscriberId()、SetWifiEnabled()、

[0031] execHttpRequest()、sendTextMessage()、Runtime.exec()、

[0032] Cipher.getInstance();

[0033] S8:网络地址类:恶意代码通常建立网络连接来接收命令、发送数据,因此,IP地址,Hostnames和URL这些在反汇编代码里的关键词也是恶意代码里常见的,所以也要包含在特征集当中。

[0034] 以上特征构建出一个特征向量S,S=S1∪S2....∪S8,每一个Android应用x都要映射到这S中,$\varphi: X \to \{0,1\}^{|S|}, \varphi(x) \to (I(x,s))_{s \in S}$,其中I(x,s)表示为:

$$I(x,s) = \begin{cases} 1, & x\text{包含特征}s \\ 0, & x\text{不包含特征}s \end{cases}$$

[0035] Android APP字节码除了可以反编译成Java源码外,还可以反编译成Smali代码,其中可提取若干特征用来描述API之间的关联分析。通常,在某一代码块中,会出现某一种行为的API,但如果连续出现多个具有相同行为意图但是又调用的是不同的API,这一块代码就具有恶意行为的特征。这些特征API共存同一代码块的关系即可表述为高层次的关联分析。这种关联定义为矩阵$A_{ij}=a_{ij} \in \{0,1\}$。

[0036] 为了描述这样的关系R,每一个API代码块定义成一个矩阵B,对于检测模块和待测模块中出现的相同的API,当出现在相同代码块中时,该API在矩阵B中为1,出现在不同的代码块中,则为0,即$B_{ij}=b_{ij} \in \{0,1\}$。

[0037] API之间的调用方式,分为以下几种:1、静态调用:根据参数,参数由其他函数返

回;2、直接调用:API中直接调用其他API;3、通过中间接口调用;

[0038] 这样一种关联可以定义为矩阵$I_{ij}=i_{ij}\in\{0,1\}$。

[0039] 以上各种特征我们可以归结为如下表所示的特征矩阵:

[0040]

矩阵	元素	描述
A	a_{ij}	app_i包含API_j，则$a_{ij}=1$，否则$a_{ij}=0$
B	b_{ij}	API_i和API_j共存于同一代码块，则$b_{ij}=1$，否则$b_{ij}=0$
I	i_{ij}	API_i和API_j使用相同的调用方式，则$i_{ij}=1$，否则$i_{ij}=0$

[0041] 上述各个矩阵的组合,用图论建立模型,即一个包含API之间关系的APP模块可以描述为G=(v,ε),v表示各个软件代码实体函数或参数,ε表示他们之间的关联。用图的方式构建包含API代码及其关系的特征空间。

[0042] 在构建了特征空间后,采用监督学习的方式对已有的Android APP库进行训练学习。在原有的高维特征空间的基础上,进行多核学习进行特征聚合。从而达到更好的分类效果。其框架如下:

[0043] $$\min\frac{1}{2}\sum_{k}\left\|wk\right\|^{2}/\beta k+C\sum_{i}\xi_{i}+\frac{\lambda}{2}(\sum_{k}\beta_{k}^{p})^{2/p}$$

[0044] 其中,对一组训练数据可以表示为$\{x_i,y_i\}_{i=1}^{N}$,x_i表示APP而$y_i\in\{+1,-1\}$表示训练数据集的标签。

[0045] 目前用于分类的特征,这些特征都是高维向量。

[0046] 准备工作:

[0047] 1.人工标注APP类别正负样本;

[0048] 2.提取正负训练样本APP的各个特征;

[0049] 3.归一化特征;

[0050] 4.为每个特征配置对应的核函数,以及参数;

[0051] 经过训练,输出训练模型文件,以及包含的核函数权重。

[0052] 为了使本发明的目的、技术方案及优点更加清楚明白,以下结合附图1、2及实施,对本发明进行进一步详细说明。

[0053] 本发明的整体思路是采用基于静态分析的方法,提取样本集里各样本的静态特征(包括API调用,权限等)并分析API之间的调用关系,进而构建各应用的特征向量,然后利用机器学习SVM分类算法对样本进行训练,获得性能较好的分类模型。用户可以通过Web页面根据提示上传待检测的Android应用软件APK,服务器后台调用前述的分类器模型对应用进行分析并将分析结果返回至Web页面。分析结果包括对该应用是否为恶意应用的判定,检测过程中权值较高的特征以及其他特殊信息。

[0054] 图1示出了本发明一种Android恶意应用在线检测方法的总体结构图。本发明主要包括如下五个步骤:

[0055] S1:训练样本数据集获取:利用爬虫技术从各大应用市场、网络论坛及相关研究机构中获取非恶意Android应用和恶意Android应用的APK安装包文件。

[0056] S2:训练样本静态分析:利用Apktool、dex2jar等静态分析软件对样本集里的各应用进行静态分析,获得各样本对应的配置文件AndroidManifest.xml文件以及反编译后的Smali文件和Java源码文件。

[0057] S3:训练样本特征提取。利用S2步骤获得AndroidManifest.xml、Smali文件和Java源码文件后,可以对相关文件进行特征提取。

[0058] 本系统提取的特征包括:

[0059] 1:硬件类:APP在运行时所需要的硬件信息,如Camera,Touchscreen,GPS等;由于硬件请求具备一定的安全风险,如接入GPS和Network,它能够把私人位置信息通过网络发送给攻击者。所以硬件类特征是特征之一;

[0060] 2:权限请求类:由于权限是Android里最重要的安全机制,恶意代码在运行时经常会请求权限,如SEND_SMS Permission;

[0061] 3:APP组件类:每个APP都包含4组:Activities,Services,Content Providers,Broadcast Receivers。每一个APP都可以声明若干组件,这些组件的名称也是作为一类特征;

[0062] 4:Intents类:Android下的Inter-Process以及Intra-Process通信,都是靠Intent。一些恶意代码经常监听某些特定的Intent,可作为一类特征;

[0063] Android APP用Java语言编写,并且编译成优化的字节码,这些字节码可以被反汇编,从而可以从中提取特征信息;

[0064] 5:受限API调用类:Android权限系统限制访问一系列关键的API调用,搜索这些调用发生时的反汇编代码,可以揭露恶意行为,没有申请权限就调用关键API,这可能表明了恶意软件正在使用ROOT以绕过Android平台限制;

[0065] 6:已用权限类:包含了已请求的权限和正在执行的权限,并把API和其申请的权限对应匹配;

[0066] 7:可疑API调用类:某些API要用敏感数据和资源,这些都是恶意代码经常用的,包括getDeviceId()、getSubscriberId()、SetWifiEnabled()、execHttpRequest()、sendTextMessage()、Runtime.exec()、Cipher.getInstance();

[0067] 8:网络地址类:恶意代码通常建立网络连接来接收命令,发送数据,因此,IP地址,Hostnames和URL这些在反汇编代码里的关键词也是恶意代码里常见的,所以也要包含在特征集当中。

[0068] S4:训练样本特征分析及向量化;将S3步骤提取的特征构建出一个特征向量S,S=S1∪S2....∪S8,每一个Android应用x都要映射到这S中,$\varphi: X \to \{0,1\}^{|S|}, \varphi(x) \to (I(x,s))_{s \in S}$,其中I(x,s)表示为:

[0069] $$I(x,s) = \begin{cases} 1, & x\text{包含特征}s \\ 0, & x\text{不包含特征}s \end{cases}$$

[0070] 此外,利用图论理论对API之间的调用关系进行分析建模,并用图的方式构建包含API代码及其关系的特征空间;最后,还需标定每个样本的类别信息,表明其属于恶意应用还是正常应用。

[0071] S5:分类器模型训练。利用SVM机器学习算法训练分类器模型的步骤如下:

[0072] 1、把所有样本的特征向量和其对应的分类标记交给算法进行训练;

[0073] 2、如果发现线性可分,就直接找出超平面;

[0074] 3、如果发现线性不可分,那就映射到n+1维空间,找出超平面;

[0075] 4、最后得到超平面的表达式,也就是分类器模型的参数。

[0076] 图2示出了本发明一种Android恶意应用在线检测方法的Web架构示意图,本发明采用Flask web应用框架,用户可以在web网页上上传待检测应用的APK文件,Web服务器收到用户的请求之后会调用相应的处理函数进行处理及反馈。

[0077] 本发明的主要工作流程如下:

[0078] S1:用户在目标网页上传待检测Android应用的APK文件。

[0079] S2:Web服务器把接收到的客户端请求转发给Flask程序实例。

[0080] S3:Flask程序首先对用户上传的文件进行文件类型检测,判断是否为有效的APK文件。

[0081] S4:调用上文所述的特征提取模块,对待检测应用进行静态分析、特征提取、特征分析及向量化。

[0082] S5:调用上文所述的分类器模型对待检测应用进行分类判别,得到分类结果及高权重特征信息。

[0083] S6:检测结果相应,Flask程序在相应网页上显示该应用的检测结果,包括对该应用是否为恶意应用的判定,检测过程中权值较高的特征以及其他特殊信息。

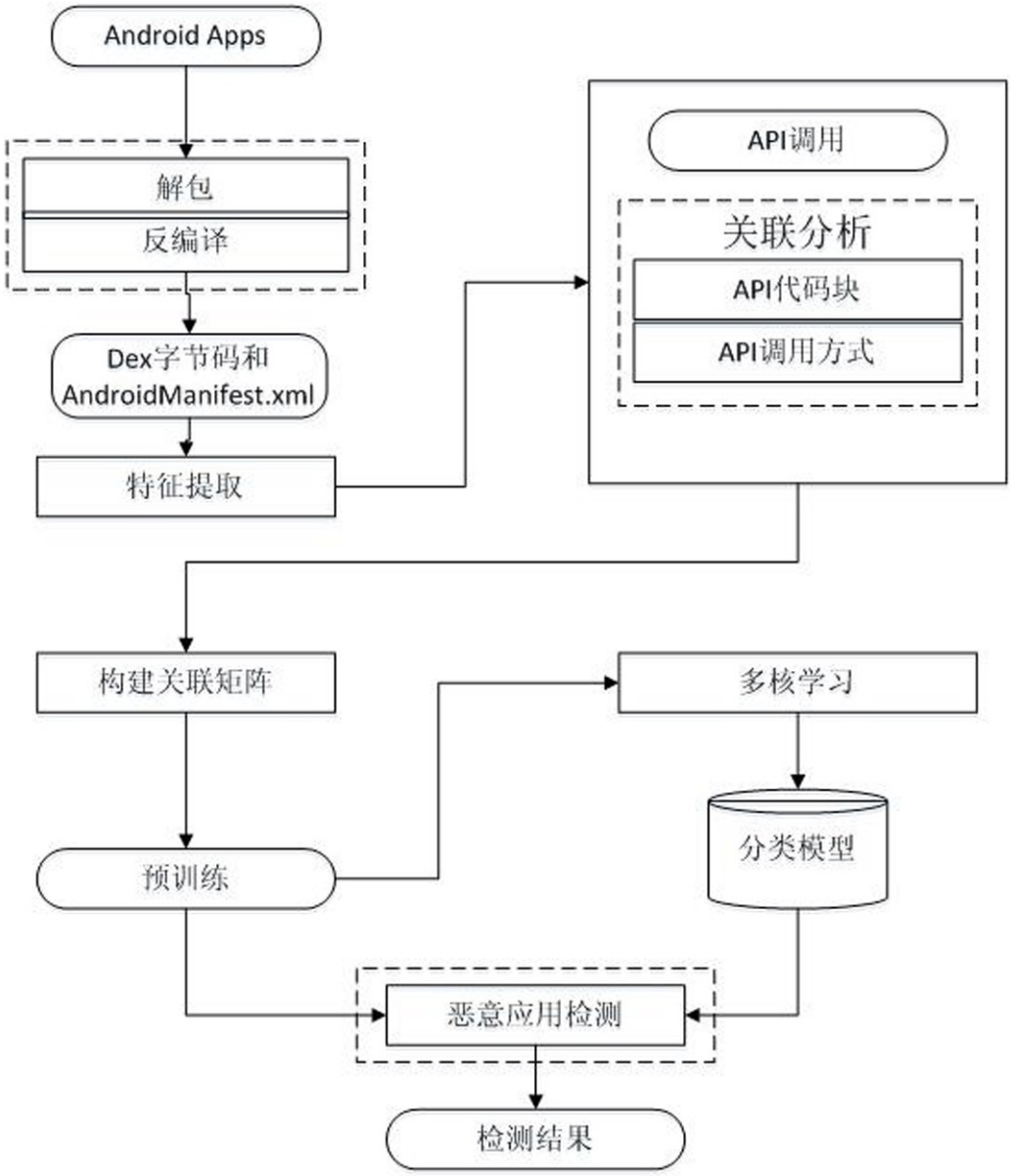

Android Apps
解包
反编译
Dex字节码和
AndroidManifest.xml
特征提取
API调用
关联分析
API代码块
API调用方式
构建关联矩阵
多核学习
分类模型
预训练
恶意应用检测
检测结果

图1

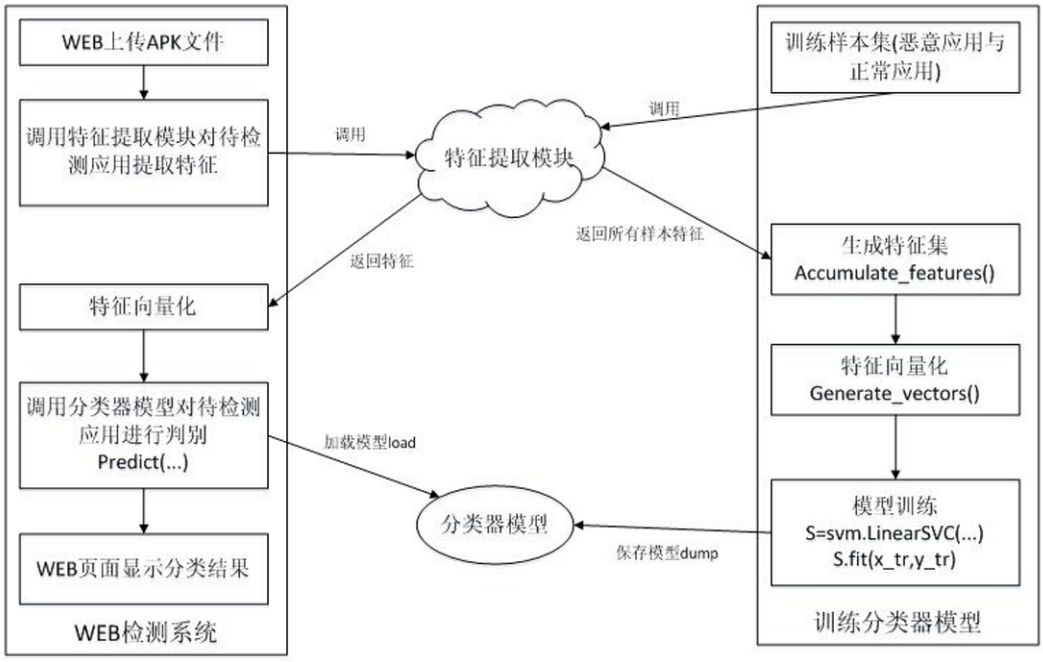

WEB上传APK文件
调用特征提取模块对待检测应用提取特征
特征向量化
调用分类器模型对待检测应用进行判别
Predict(...)
WEB页面显示分类结果
WEB检测系统
调用
返回特征
加载模型load
特征提取模块
分类器模型
调用
返回所有样本特征
保存模型dump
训练样本集(恶意应用与正常应用)
生成特征集
Accumulate_features()
特征向量化
Generate_vectors()
模型训练
S=svm.LinearSVC(...)
S.fit(x_tr,y_tr)
训练分类器模型

图2