(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 110489968 B
(45)授权公告日 2021.02.05

(21)申请号 201910751207.5

(22)申请日 2019.08.15

(65)同一申请的已公布的文献号
申请公布号 CN 110489968 A

(43)申请公布日 2019.11.22

(73)专利权人 东北大学秦皇岛分校
地址 066004 河北省秦皇岛市经济技术开发区泰山路143号

(72)发明人 赵立超 李丹 陈璨 史闻博 李天祥

(74)专利代理机构 石家庄知住优创知识产权代理事务所(普通合伙) 13131
代理人 林艳艳

(51)Int.Cl.
G06F 21/56(2013.01)
G06N 3/04(2006.01)
G06N 3/08(2006.01)
G06F 8/53(2018.01)

审查员 刘燕

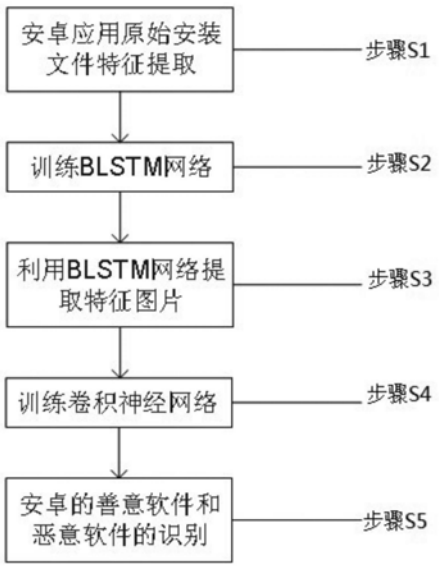
权利要求书3页 说明书7页 附图3页

(54)发明名称

基于RNN和CNN的Android恶意软件检测方法及系统

(57)摘要

本发明提供了一种基于RNN和CNN的Android恶意软件检测方法及系统,检测方法包括:对训练样本的原始安装文件进行特征提取,获得操作码序列;利用操作码序列训练BLSTM网络;利用训练好的BLSTM网络将操作码序列提取为特征图片;利用特征图片训练卷积神经网络;对待检测Android应用,首先对其安装文件进行特征提取,获得其操作码序列;然后将该操作码序列输入训练好的BLSTM网络中,提取出特征图片;最后将该特征图片输入到训练好的卷积神经网络中,输出是否属于恶意软件的分类结果。本发明实现对Android平台下的善意软件和恶意软件的识别区分,提高Android软件平台的安全性。



CN 110489968 B

1.一种基于RNN和CNN的Android恶意软件检测方法,其特征在于:包括以下步骤:

S1,对训练样本的原始安装文件进行特征提取,获得操作码序列;

S2,利用操作码序列训练BLSTM网络,得到训练好的BLSTM网络;

S3,利用训练好的BLSTM网络将操作码序列提取为特征图片;

所述步骤S3具体包括以下处理:

S3-1,对照Dalvik指令表对操作码序列进行one-hot编码后再输入到训练好的BLSTM网络中;

S3-2,提取每个隐藏层的输出向量{h_1,h_2,…,h_L};

S3-3,利用 $p_k=\begin{cases}0, & k=0\\ \left\lfloor\frac{L+k-1}{N}\right\rfloor+p_{k-1}, & 1\le k\le N\end{cases}$, $f_k=\frac{1}{p_k-p_{k-1}}\sum_{j=p_{k-1}+1}^{p_k}h_j$,将L个隐藏层的输出向量分成N个向量组,对每个向量组求平均,得到N个特征向量;其中,f_k是固定长度向量序列的一个元素,L是操作码序列的长度,N是特征图片的高度,p_k是第k个向量组的最后一个数字;

S3-4,将N个特征向量拼在一起构成N*W的特征矩阵F:

$$F=\begin{pmatrix}f_1\\ f_2\\ \vdots\\ f_N\end{pmatrix}\begin{pmatrix}f_{11} & f_{12} & \cdots & f_{1W}\\ f_{21} & f_{22} & \cdots & f_{2W}\\ \vdots & \vdots & \ddots & \vdots\\ f_{N1} & f_{N2} & \cdots & f_{NW}\end{pmatrix}$$

,其中W是BLSTM中隐藏层的维度,Dalvik指令共256个,故W=256;

S3-5,用sigmoid函数将特征矩阵F中的元素转化为[0,1],再乘255以形成256级灰色图片,得到尺寸为N*256的特征图片;

S4,利用特征图片训练卷积神经网络,得到训练好的卷积神经网络;

S5,对待检测Android应用,首先对其安装文件进行特征提取,获得其操作码序列;然后将该操作码序列输入步骤S2获得的训练好的BLSTM网络中,提取出特征图片;最后将该特征图片输入到步骤S4中训练好的卷积神经网络中,输出是否属于恶意软件的分类结果。

2.根据权利要求1所述的基于RNN和CNN的Android恶意软件检测方法,其特征在于:所述步骤S1具体包括以下处理:

S1-1,解压缩训练样本的.apk安装文件,获取.apk安装文件包含的class.dex文件;

S1-2,对class.dex文件进行反编译,获取Android应用的.smali文件;

S1-3,舍弃.smali文件中的操作数,得到操作码序列。

3.根据权利要求1所述的基于RNN和CNN的Android恶意软件检测方法,其特征在于:所述步骤S2具体包括以下处理:

S2-1,BLSTM网络参数和权重随机初始化;

S2-2,将操作码序列输入BLSTM网络,进行训练;

S2-3,通过对比当前神经元的输出向量和下一神经元的输入向量计算损失函数,通过反向传播算法更新参数和权重;循环执行S2-2至S2-3,直至BLSTM满足条件或训练周期结束。

4.根据权利要求1所述的基于RNN和CNN的Android恶意软件检测方法，其特征在于：所述步骤S4具体包括以下处理：

S4-1，卷积神经网络参数和权重随机初始化；

S4-2，将尺寸为N*256的特征图片输入卷积神经网络，进行训练；

S4-3，根据神经网络的误测精度反向传播进行权重和参数的调整，直至CNN满足条件或训练周期结束。

5.一种基于RNN和CNN的Android恶意软件检测系统，其特征在于：是根据权利要求1-4任一项所述的基于RNN和CNN的Android恶意软件检测方法实现的，所述系统包括：

训练样本获取模块：用于获取训练样本，所述训练样本为已知类型的软件的执行程序，所述类型包括良性和恶意；

训练样本处理模块：用于对所述训练样本进行解压、反编译和one-hot编码处理，得到所述训练样本处理后的操作码序列；

BLSTM训练模块：用于以编码后的操作码序列为输入的向量序列，训练BLSTM网络，当所述BLSTM网络输出的预测下一输入序列的准确度没有达到设定值时，则根据下一输入序列调整训练过程中的权重参数，继续训练BLSTM网络；当所述BLSTM网络预测下一输入向量的准确度达到设定值时，停止训练BLSTM网络，最终得到输出为预测的下一输入序列的BLSTM网络；

特征图片提取模块：用于利用训练好的BLSTM网络对操作码序列进行特征提取，得到特征图片；

卷积神经网络训练模块：用于以提取出的特征图片为输入矩阵，训练卷积神经网络，当所述卷积神经网络输出的恶意概率或非恶意概率的准确度没有达到设定值时，则根据所述卷积神经网络输出的恶意概率或非恶意概率的准确度调整训练过程中的权重参数，继续训练卷积神经网络；当所述卷积神经网络输出的准确度达到设定值时，停止训练卷积神经网络，最终得到输出为准确度的卷积神经网络；

准确度判断模块：用于判断所述BLSTM网络预测下一输入向量的准确度和所述卷积神经网络输出的恶意概率或非恶意概率的准确度是否达到设定值；

识别模块：用于利用训练好的卷积神经网络对待检测Android应用进行检测，输出该Android应用是否属于恶意软件的识别结果。

6.根据权利要求5所述的基于RNN和CNN的Android恶意软件检测系统，其特征在于：所述训练样本处理模块具体包括：

解压单元：用于对获取的训练样本进行解压，获得class.dex文件；

反编译单元：用于对class.dex文件进行反编译，获得含有操作码的.smali文件；

编码处理单元：用于提取.smali文件中的操作码，并对照Dalvik指令表对操作码进行one-hot编码，获得操作码序列。

7.根据权利要求5所述的基于RNN和CNN的Android恶意软件检测系统，其特征在于：所述特征图片提取模块具体包括：

特征向量提取单元：用于将操作码序列输入BLSTM网络中，提取每一个隐藏层的输出向量；

固定尺寸单元：用于将l个向量序列分成N份，对每份向量组求平均，形成N个特征向量；

向量拼接单元:用于将得到的N个特征向量拼在一起,形成固定尺寸的特征图片。

8.根据权利要求5所述的基于RNN和CNN的Android恶意软件检测系统,其特征在于:所述卷积神经网络训练模块具体包括:

参数设置单元:用于设置卷积神经网络的内层参数top K,K=3;

训练单元:用于以提取的特征图片为输入矩阵,训练卷积神经网络。

9.根据权利要求5所述的基于RNN和CNN的Android恶意软件检测系统,其特征在于:所述识别模块具体包括:

待检测软件获取单元:用于获取待检测Android应用的安装文件;

待检测软件处理单元:用于对待检测Android应用的安装文件进行解压、反编译和one-hot编码处理,得到待检测Android应用处理后的操作码序列;

待检测软件特征图片提取单元:用于将待检测Android应用的操作码序列输入训练好的BLSTM网络中,提取特征图片;

识别单元:用于将所述待检测Android应用提取出的特征图片作为特征矩阵输入卷积神经网络,进行识别。

基于RNN和CNN的Android恶意软件检测方法及系统

技术领域

[0001] 本发明涉及恶意软件检测领域，具体涉及一种基于RNN和CNN的Android恶意软件检测方法及系统。

背景技术

[0002] 现在，互联网是我们生活工作中非常重要的一部分。但是，基于恶意软件的网络攻击也是一个很严重的问题。随着科学技术的发展，恶意软件的种类以及复杂度越来越高，对恶意软件的识别也越来越具有难度，尤其是在移动领域平台。鉴于移动设备和手机应用商店的快速增长，新应用程序的数量太大而无法手动检查每个程序的恶意行为，恶意软件检测已经成为了现今移动互联网领域发展的重要技术保障。研究和实现高准确度的恶意软件检测系统具有很重要的现实意义，受到了相关学术界和业界的密切关注。

[0003] 深度学习是近年来新兴起的一种新的机器学习领域，通过建立具有阶层结构的人工神经网络，在计算机系统中实现人工智能。其中多层神经元通过不同的权重和激活函数相互连接，以学习输入和输出之间的隐藏关系。深度学习被用于对复杂结构和大样本的高维数据进行学习，在人像识别、机器翻译、自动驾驶等现实问题中取得了成功。

[0004] Android恶意软件检测方法目前主要有两种，即静态分析和动态检测。静态分析是指通过分析程序代码来判断程序行为。动态分析是指在严格控制的环境下执行应用程序，尽可能的触发软件的全部行为并记录，以检测应用程序是否包含恶意行为。目前已有的静态分析方法有依赖于字节码和操作码的n-gram分析，该方法首先计算字节码的n-gram，然后根据KNN算法执行恶意软件检测。此外，还有一类恶意软件检测方法依赖于将恶意软件转换为图像。该方法将二进制字节码转换成灰度图像，并对图像进行模式识别。以上方法均达到了一定的检测精度。但是，随着恶意软件数量的急剧增加，用于训练模型的数据集的大小对检测精度和训练效率也有显著影响。虽然n-gram方法的检测精度很高，但是它需要大量的计算资源和时间来处理所需模型参数的动态增长。然而，CNN能够处理爆炸性的数据增长，因为参数数量的增加并不意味着计算资源和所需时间的增长。但是如果直接将操作码序列用one-hot编码转换成特征矩阵作为卷积神经网络的输入，就忽略了操作码序列的前后关联关系。

发明内容

[0005] 为了解决现有技术问题，本发明提供一种基于RNN和CNN的Android恶意软件检测方法及系统，实现对Android平台下的善意软件和恶意软件的识别区分，具有识别精度高的特点，提高Android软件平台的安全性。

[0006] 为解决上述技术问题，本发明所采取的技术方案是：

[0007] 一种基于RNN和CNN的Android恶意软件检测方法，包括以步骤：

[0008] S1，对训练样本的原始安装文件进行特征提取，获得操作码序列；

[0009] S2，利用操作码序列训练BLSTM网络，得到训练好的BLSTM网络；

[0010] S3,利用训练好的BLSTM网络将操作码序列提取为特征图片;

[0011] S4,利用特征图片训练卷积神经网络,得到训练好的卷积神经网络;

[0012] S5,对待检测Android应用,首先对其安装文件进行特征提取,获得其操作码序列;然后将该操作码序列输入步骤S2获得的训练好的BLSTM网络中,提取出特征图片;最后将该特征图片输入到步骤S4中训练好的卷积神经网络中,输出是否属于恶意软件的分类结果。

[0013] 进一步的,所述步骤S1具体包括以下处理:

[0014] S1-1,解压缩训练样本的.apk安装文件,获取.apk安装文件包含的class.dex文件;

[0015] S1-2,对class.dex文件进行反编译,获取Android应用的.smali文件;

[0016] S1-3,舍弃.smali文件中的操作数,得到操作码序列。

[0017] 进一步的,所述步骤S2具体包括以下处理:

[0018] S2-1,BLSTM网络参数和权重随机初始化;

[0019] S2-2,将操作码序列输入BLSTM网络,进行训练;

[0020] S2-3,通过对比当前神经元的输出向量和下一神经元的输入向量计算损失函数,通过反向传播算法更新参数和权重;循环执行S2-2至S2-3,直至BLSTM满足条件或训练周期结束。

[0021] 进一步的,所述步骤S3具体包括以下处理:

[0022] S3-1,对照Dalvik指令表对操作码序列进行one-hot编码后再输入到训练好的BLSTM网络中;

[0023] S3-2,提取每个隐藏层的输出向量$\{h_1,h_2,\cdots,h_L\}$;

[0024] S3-3,利用 $p_k=\begin{cases}0, & k=0\\ \left\lfloor\frac{L+k-1}{N}\right\rfloor+p_{k-1}, & 1\le k\le N\end{cases}$, $f_k=\frac{1}{p_k-p_{k-1}}\sum_{j=p_{k-1}+1}^{p_k}h_j$,将L个隐藏层的输出向量分成N个向量组,对每个向量组求平均,得到N个特征向量;其中,f_k是固定长度向量序列的一个元素,L是操作码序列的长度,N是特征图片的高度,p_k是第k个向量组的最后一个数字;

[0025] S3-4,将N个特征向量拼在一起构成N*W的特征矩阵F:

[0026] $F=\begin{pmatrix}f_1\\ f_2\\ \vdots\\ f_N\end{pmatrix}\begin{pmatrix}f_{11} & f_{12} & \cdots & f_{1W}\\ f_{21} & f_{22} & \cdots & f_{2W}\\ \vdots & \vdots & \ddots & \vdots\\ f_{N1} & f_{N2} & \cdots & f_{NW}\end{pmatrix}$,其中W是BLSTM中隐藏层的维度,Dalvik指令共256个,故W=256;

[0027] S3-5,用sigmoid函数将特征矩阵F中的元素转化为[0,1],再乘255以形成256级灰色图片,得到尺寸为N*256的特征图片。

[0028] 进一步的,所述步骤S4具体包括以下处理:

[0029] S4-1,卷积神经网络参数和权重随机初始化;

[0030] S4-2,将尺寸为N*256的特征图片输入卷积神经网络,进行训练;

[0031] S4-3,根据神经网络的误测精度反向传播进行权重和参数的调整,直至CNN满足条

件或训练周期结束。

[0032] 一种基于RNN和CNN的Android恶意软件检测系统,所述系统包括:

[0033] 训练样本获取模块:用于获取训练样本,所述训练样本为已知类型的软件的执行程序,所述类型包括良性和恶意;

[0034] 训练样本处理模块:用于对所述训练样本进行解压、反编译和one-hot编码处理,得到所述训练样本处理后的操作码序列;

[0035] BLSTM训练模块:用于以编码后的操作码序列为输入的向量序列,训练BLSTM网络,当所述BLSTM网络输出的预测下一输入序列的准确度没有达到设定值时,则根据下一输入序列调整训练过程中的权重参数,继续训练BLSTM网络;当所述BLSTM网络预测下一输入向量的准确度达到设定值时,停止训练BLSTM网络,最终得到输出为预测的下一输入序列的BLSTM网络;

[0036] 特征图片提取模块:用于利用训练好的BLSTM网络对操作码序列进行特征提取,得到特征图片;

[0037] 卷积神经网络训练模块:用于以提取出的特征图片为输入矩阵,训练卷积神经网络,当所述卷积神经网络输出的恶意概率或非恶意概率的准确度没有达到设定值时,则根据所述卷积神经网络输出的恶意概率或非恶意概率的准确度调整训练过程中的权重参数,继续训练卷积神经网络;当所述卷积神经网络输出的准确度达到设定值时,停止训练卷积神经网络,最终得到输出为准确度的卷积神经网络;

[0038] 准确度判断模块:用于判断所述BLSTM网络预测下一输入向量的准确度和所述卷积神经网络输出的恶意概率或非恶意概率的准确度是否达到设定值;

[0039] 识别模块:用于利用训练好的卷积神经网络对待检测Android应用进行检测,输出该Android应用是否属于恶意软件的识别结果。

[0040] 进一步的,所述训练样本处理模块具体包括:解压单元:用于对获取的训练样本进行解压,获得class.dex文件;反编译单元:用于对class.dex文件进行反编译,获得含有操作码的.smali文件;编码处理单元:用于提取.smali文件中的操作码,并对照Dalvik指令表对操作码进行one-hot编码,获得操作码序列。

[0041] 进一步的,所述特征图片提取模块具体包括:特征向量提取单元:用于将操作码序列输入BLSTM网络中,提取每一个隐藏层的输出向量;固定尺寸单元:用于将l个向量序列分成N份,对每份向量组求平均,形成N个特征向量;向量拼接单元:用于将得到的N个特征向量拼在一起,形成固定尺寸的特征图片。

[0042] 进一步的,所述卷积神经网络训练模块具体包括:参数设置单元:用于设置卷积神经网络的内层参数top K,K=3;训练单元:用于以提取的特征图片为输入矩阵,训练卷积神经网络。

[0043] 进一步的,所述识别模块具体包括:待检测软件获取单元:用于获取待检测Android应用的安装文件;待检测软件处理单元:用于对待检测Android应用的安装文件进行解压、反编译和one-hot编码处理,得到待检测Android应用处理后的操作码序列;待检测软件特征图片提取单元:用于将待检测Android应用的操作码序列输入训练好的BLSTM网络中,提取特征图片;识别单元:用于将所述待检测Android应用提取出的特征图片作为特征矩阵输入卷积神经网络,进行识别。

[0044] 采用上述技术方案所产生的有益效果在于：

[0045] 本发明提供的基于RNN和CNN的Android恶意软件检测方法及系统，采用软件的操作码为识别对象，通过BLSTM作为特征提取器来提取特征图片，BLSTM的正向能保留操作码序列和上文的关系，反向能保留操作码序列和下文的关系，因此能更好的提取操作码的相关特征。此外，本发明对卷积神经网络的trunk k层进行了分块处理，保留了更多的特征值。对操作码的特征提取以及对trunk k层的分块处理，使得本发明的Android恶意软件检测方法及系统具有识别精度高的特点。而且，相比于传统的手动识别，本发明操作更加简便。

附图说明

[0046] 图1是本发明基于RNN和CNN的Android恶意软件检测流程图；

[0047] 图2为本发明提出的特征提取的方法；

[0048] 图3是本发明基于RNN和CNN的Android恶意软件检测系统结构示意图。

具体实施方式

[0049] 下面结合附图和具体实施方式对本发明作进一步详细的说明。

[0050] 本发明提供了一种基于RNN和CNN的Android恶意软件检测方法及系统，用于实现对Android平台下的善意软件和恶意软件进行识别区分，以提高Android软件平台的安全性。

[0051] 如图1所示，本发明的基于RNN和CNN的Android恶意软件检测方法，包括以下步骤：

[0052] S1，对训练样本的原始安装文件进行特征提取，获得操作码序列；

[0053] 步骤S1具体包括以下处理：

[0054] S1-1，采用7-zip工具解压缩训练样本的.apk安装文件，获取.apk安装文件包含的class.dex文件；

[0055] S1-2，采用.apktool反编译程序对class.dex文件进行反编译，获取Android应用的操作码清单.smali文件；

[0056] S1-3，舍弃.smali文件中的操作数，得到操作码序列。

[0057] S2，利用操作码序列训练BLSTM网络，得到训练好的BLSTM网络；

[0058] 步骤S2具体包括以下处理：

[0059] S2-1，BLSTM网络参数和权重随机初始化；

[0060] S2-2，将操作码序列输入BLSTM网络，进行训练；

[0061] 遗忘门：$f_t=\sigma(W_f \cdot [h_{t-1},x_t]+b_f)$，其中σ为sigmoid函数；

[0062] 输入门：$i_t=\sigma(W_i \cdot [h_{t-1},x_t]+b_i)$；

[0063] 更新细胞状态：$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t$，其中$\widetilde{C}_t = \tanh(W_c \cdot [h_{t-1},\ x_t]+b_c)$；

[0064] 输出门：$o_t=\sigma(W_o \cdot [h_{t-1},x_t]+b_o)$，其中$h_t=o_t*\tanh(C_t)$；

[0065] S2-3，通过对比当前神经元的输出向量和下一神经元的输入向量计算损失函数，通过反向传播算法更新参数和权重；循环执行S2-2至S2-3，直至特征提取模型BLSTM满足条件或训练周期结束，得到训练好的BLSTM网络。

[0066] S3，利用训练好的BLSTM网络将操作码序列提取为特征图片；

[0067] 如图2所示，步骤S3具体包括以下处理：

[0068] S3-1,对照Dalvik指令表对操作码序列进行one-hot编码后再输入到训练好的BLSTM网络中;

[0069] S3-2,提取每个隐藏层的输出向量{$h_1,h_2,\cdots,h_L$};

[0070] S3-3,利用$p_k=\begin{cases}0, & k=0\\ \left\lfloor\frac{L+k-1}{N}\right\rfloor+p_{k-1}, & 1\le k\le N\end{cases}$,$f_k=\frac{1}{p_k-p_{k-1}}\sum_{j=p_{k-1}+1}^{p_k}h_j$,将L个隐藏层的输出向量分成N个向量组,对每个向量组求平均,得到N个特征向量,以实现将不定长的操作码序列转换为固定尺寸的特征图片;其中,f_k是固定长度向量序列的一个元素,L是操作码序列的长度,N是特征图片的高度,p_k是第k个向量组的最后一个数字;

[0071] S3-4,将N个特征向量拼在一起构成N*W的特征矩阵F:

[0072] $F=\begin{pmatrix}f_1\\ f_2\\ \vdots\\ f_N\end{pmatrix}\begin{pmatrix}f_{11} & f_{12} & \cdots & f_{1W}\\ f_{21} & f_{22} & \cdots & f_{2W}\\ \vdots & \vdots & \ddots & \vdots\\ f_{N1} & f_{N2} & \cdots & f_{NW}\end{pmatrix}$,其中W是BLSTM中隐藏层的维度,Dalvik指令共256个,故W=256;

[0073] S3-5,用sigmoid函数将特征矩阵F中的元素转化为[0,1],再乘255以形成256级灰色图片,得到特征图片。

[0074] S4,利用特征图片训练卷积神经网络,得到训练好的卷积神经网络;

[0075] 步骤S4具体包括以下处理:

[0076] S4-1,卷积神经网络参数和权重随机初始化;

[0077] S4-2,将尺寸为N*256的特征图片输入卷积神经网络,进行训练;

[0078] Embedding Layer:将N*256的矩阵乘以256*8的权重矩阵得到N*8的矩阵;

[0079] Reshape层:将N*8的矩阵变为1*N*8*1的矩阵;

[0080] 卷积层:将1*N*8*1的矩阵经过k*8*64的卷积核得到1*(N-k+1)*1*64的矩阵;

[0081] trunk k层(将卷积层输出的矩阵分为k块,本发明选的是三块,以保留更多特征):将卷积层输出的矩阵平均分为三份,在每份中取最大的三个值(top 3),得到1*9*1*64的矩阵;

[0082] Reshape层:将1*9*1*64的矩阵变成9*64的矩阵;

[0083] 全连接层:用1*9的全连接权重矩阵乘以上一步得到的9*64的矩阵变成1*64的矩阵;

[0084] 全连接层:将1*64的矩阵乘以64*16的全连接权重矩阵变成1*16的矩阵;

[0085] 全连接层:将1*16的矩阵乘以16*2的全连接矩阵变成1*2的矩阵获得最终解果(x,y)即预测应用是恶意和非恶意的概率;

[0086] S4-3,根据神经网络的误测精度反向传播进行权重和参数的调整,直至CNN满足条件或训练周期结束,得到训练好的卷积神经网络。

[0087] S5,对待检测的Android应用,首先对其安装文件进行特征提取,获得其操作码序列;然后将该操作码序列输入步骤S2获得的训练好的BLSTM网络中,提取出特征图片;最后将该特征图片输入到步骤S4中训练好的卷积神经网络中,输出是否属于恶意软件的分类结

果。

[0088] 如图3所示，本发明的基于RNN和CNN的Android恶意软件检测系统，包括训练样本获取模块、训练样本处理模块、BLSTM训练模块、特征图片提取模块、卷积神经网络训练模块、准确度判断模块和识别模块。

[0089] 其中，训练样本获取模块用于获取训练样本，所述训练样本为已知类型的软件的执行程序，所述类型包括良性和恶意；训练样本处理模块用于对所述训练样本进行解压、反编译和one-hot编码处理，得到所述训练样本处理后的操作码序列；BLSTM训练模块用于以编码后的操作码序列为输入的向量序列，训练BLSTM网络，当所述BLSTM网络输出的预测下一输入序列的准确度没有达到设定值时，则根据下一输入序列调整训练过程中的权重参数，继续训练BLSTM网络；当所述BLSTM网络预测下一输入向量的准确度达到设定值时，停止训练BLSTM网络，最终得到输出为预测的下一输入序列的BLSTM网络。

[0090] 特征图片提取模块用于利用训练好的BLSTM网络对操作码序列进行特征提取，得到特征图片；卷积神经网络训练模块用于以提取出的特征图片为输入矩阵，训练卷积神经网络，当所述卷积神经网络输出的恶意概率或非恶意概率的准确度没有达到设定值时，则根据所述卷积神经网络输出的恶意概率或非恶意概率的准确度调整训练过程中的权重参数，继续训练卷积神经网络；当所述卷积神经网络输出的准确度达到设定值时，停止训练卷积神经网络，最终得到输出为准确度的卷积神经网络；

[0091] 准确度判断模块用于判断所述BLSTM网络预测下一输入向量的准确度和所述卷积神经网络输出的恶意概率或非恶意概率的准确度是否达到设定值；识别模块用于利用训练好的卷积神经网络对待检测Android应用进行检测，输出该Android应用是否属于恶意软件的识别结果。

[0092] 训练样本处理模块具体包括解压单元、反编译单元和编码处理单元。其中，解压单元用于对获取的训练样本进行解压，获得class.dex文件；反编译单元用于对class.dex文件进行反编译，获得含有操作码的.smali文件；编码处理单元用于提取.smali文件中的操作码，并对照Dalvik指令表对操作码进行one-hot编码，获得操作码序列。

[0093] 特征图片提取模块具体包括特征向量提取单元、固定尺寸单元和向量拼接单元。其中，特征向量提取单元用于将操作码序列输入BLSTM网络中，提取每一个隐藏层的输出向量；固定尺寸单元用于将l个向量序列分成N份，对每份向量组求平均，形成N个特征向量；向量拼接单元用于将得到的N个特征向量拼在一起，形成固定尺寸的特征图片。

[0094] 卷积神经网络训练模块具体包括参数设置单元和训练单元。其中，参数设置单元用于设置卷积神经网络的内层参数top K，K＝3；训练单元用于以提取的特征图片为输入矩阵，训练卷积神经网络。

[0095] 识别模块具体包括待检测软件获取单元、待检测软件处理单元、待检测软件处理单元、待检测软件特征图片提取单元和识别单元。其中，待检测软件获取单元用于获取待检测Android应用的安装文件；待检测软件处理单元用于对待检测Android应用的安装文件进行解压、反编译和one-hot编码处理，得到待检测Android应用处理后的操作码序列；待检测软件特征图片提取单元用于将待检测Android应用的操作码序列输入训练好的BLSTM网络中，提取特征图片；识别单元用于将所述待检测Android应用提取出的特征图片作为特征矩阵输入卷积神经网络，进行识别。

[0096] 为了提取操作码序列中隐含的前后关系，本发明引入了BLSTM模型，这是双向循环神经网络(BRNN)的一种，这种网络的内部状态可以展示动态时序行为，可以利用它内部的记忆来处理任意时序的输入序列。每一个训练序列向前和向后分别是两个循环神经网络(RNN)，而且这两个都连接着一个输出层。这个结构提供给输出层输入序列中每一个点的完整的过去和未来的上下文信息。

[0097] 本发明首先使用操作码序列训练BLSTM网络，再利用BLSTM网络作为特征提取器，将操作码序列转化为图片，然后使用该图片作为输入对卷积神经网络进行训练，最后卷积神经网络输出待检测Android应用的恶意概率和非恶意概率。

[0098] 以上所述的实施例仅仅是对本发明的优选实施方式进行描述，并非对本发明的范围进行限定，在不脱离本发明设计精神的前提下，本领域普通技术人员对本发明的技术方案做出的各种变形和改进，均应落入本发明权利要求书确定的保护范围内。

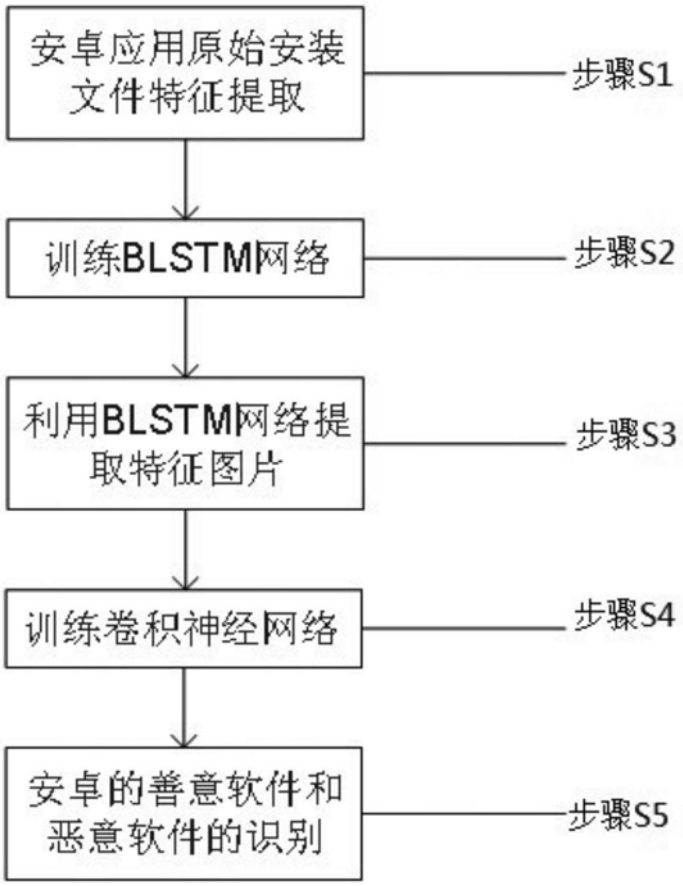

安卓应用原始安装文件特征提取
步骤S1
训练BLSTM网络
步骤S2
利用BLSTM网络提取特征图片
步骤S3
训练卷积神经网络
步骤S4
安卓的善意软件和恶意软件的识别
步骤S5

图1

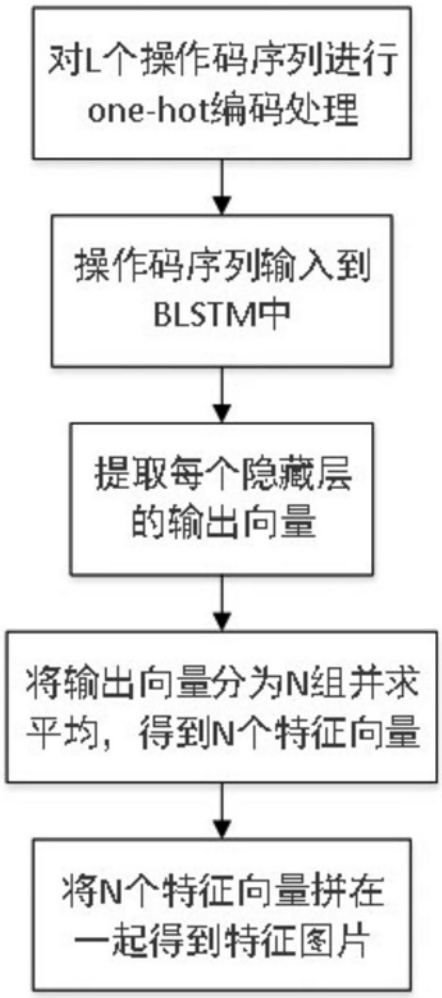

对L个操作码序列进行one-hot编码处理
操作码序列输入到BLSTM中
提取每个隐藏层的输出向量
将输出向量分为N组并求平均，得到N个特征向量
将N个特征向量拼在一起得到特征图片

图2

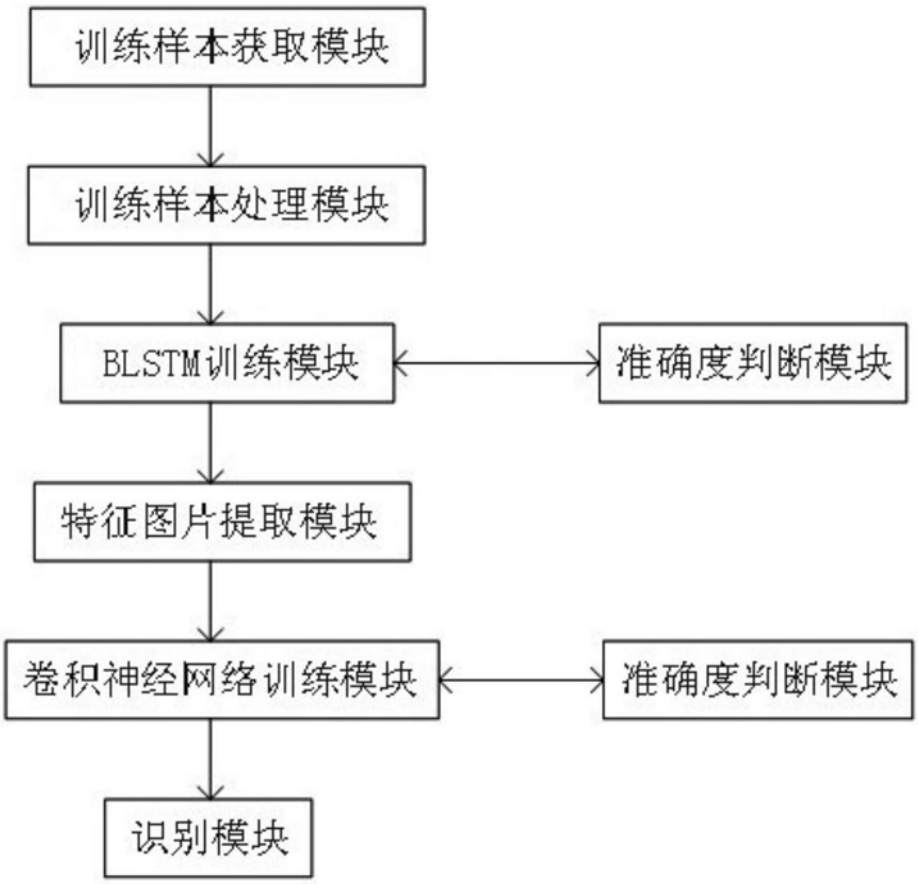

训练样本获取模块
训练样本处理模块
BLSTM训练模块
准确度判断模块
特征图片提取模块
卷积神经网络训练模块
准确度判断模块
识别模块

图3