计算机工程与应用
Computer Engineering and Applications
ISSN 1002-8331,CN 11-2127/TP

《计算机工程与应用》网络首发论文

题目：结合 CNN 和 catboost 算法的恶意安卓应用检测模型
作者：苏庆，林华智，黄剑锋，林志毅
网络首发日期：2020-09-23
引用格式：苏庆，林华智，黄剑锋，林志毅．结合 CNN 和 catboost 算法的恶意安卓应用检测模型．计算机工程与应用.
https://kns.cnki.net/kcms/detail/11.2127.TP.20200923.1603.006.html

网络首发时间：2020-09-23 16:24:31
网络首发地址：https://kns.cnki.net/kcms/detail/11.2127.TP.20200923.1603.006.html

结合 CNN 和 catboost 算法的恶意安卓应用检测模型

苏　庆，林华智，黄剑锋，林志毅

广东工业大学 计算机学院，广州 510006

摘　要：针对恶意安卓应用程序检测中存在的特征维度大、检测效率低的问题，结合卷积神经网络 CNN 良好的特征提取和降维能力以及 catboost 算法无需广泛数据训练即可产生较好分类结果的优点，构建一个 CNN-catboost 混合恶意安卓应用检测模型。首先通过逆向工程获取安卓应用的权限、API 包、组件、intent、硬件特性和 OpCode 特征等静态特征并映射为特征向量，再在特征处理层使用卷积核对特征进行局部感知处理以增强信号；然后使用最大池化对处理后的特征进行下采样，降低维数并保持特征性质不变；接着将处理后的特征作为 catboost 分类层的输入向量，利用遗传算法的全局寻优能力对 catboost 模型进行调参，进一步提升分类准确率；最后对训练完成的模型，分别使用已知和未知类型的安卓应用程序数据集作实际应用测试。实验结果表明 CNN-catboost 模型调参用时较少，在预测精度和检测效率上也展示出较为良好的效果。

关键词：恶意安卓应用；卷积神经网络；catboost 分类算法；遗传算法

文献标志码：A　　中图分类号：TP311　　**doi：10.3778/j.issn.1002-8331.2004-0385**

苏庆，林华智，黄剑锋，等. 结合 CNN 和 catboost 算法的恶意安卓应用检测模型.计算机工程与应用

SU Qing, LIN Huazhi, HUANG Jianfeng, et al. Malicious Android application detection combining CNN and catboost algorithm. Computer Engineering and Applications

Malicious Android application detection combining CNN and catboost algorithm

SU Qing, LIN Huazhi, HUANG Jianfeng, LIN Zhiyi

School of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, China

Abstract: In malicious Android application detection, there exists problems such as high dimensionality of features and low efficiency of detection. In order to solve the above problems, a CNN-catboost hybrid model is proposed. In the proposed CNN-catboost model, the convolution neural network can help feature extraction and dimension reduction, and the catboost classification algorithm has the good generalization ability. Firstly, the static features of android application, such as permissions, API packages, components, intents, hardware features and OpCode features, acquiring through reverse engineering, are encoded as feature vectors. And then, in the feature processing layer, the local features are extracted by using the convolution kernel. Secondly, the Maximum pooling is used to downsample the processed features to reduce the dimension while keeping the characteristic property the same. Finally, the downsampled features are used as the input vector of catboost classification layer, a genetic algorithm of global optimization ability is used to adjust the parameters of the catboost model to further improve classification accuracy. At last, we test our model with known and unknown type of Android app dataset. The experimental result shows that the CNN-catboost hybrid model takes less time to tune parameters, and can get promising prediction accuracy and detection efficiency.

Key words: Malicious Android application; Convolutional neural network; Catboost classification algorithm; Genetic algorithm

基金项目：国家自然科学基金(No.618002072)；广东省自然科学基金(No.2018A030313389)；广州市科技计划(No.201902020012，No.201907010021)。

作者简介：苏庆(1979-)，男，博士，副教授，CCF 会员，主要研究方向为软件安全与保护；林华智(1995-)，男，硕士研究生，主要研究方向为软件安全；黄剑锋（1979-），通信作者，男，硕士，讲师，主要研究方向为代码混淆；林志毅，博士，讲师，主要研究方向为智能计算。

1　引言

2018 年间，360 互联网安全中心记录了大约 1.1 亿次恶意安卓应用程序感染[1]，说明网络安全态势存在严重威胁，也使得恶意安卓应用检测成为网络安全

的研究热点之一。恶意安卓软件检测技术可分为静态分析技术和动态分析技术两大类[2]。其中动态分析技术是利用沙盒、虚拟机来仿真应用的执行过程，通过实时监控应用执行过程中所产生的行为来判断其是否为恶意软件，此方法必须在程序运行时才能实施检测，耗费资源和时间多。静态分析法不执行程序文件，而是先对程序进行反编译，然后提取特征值进行分析和研究，是一类较为简便的安卓恶意应用判定方法。

近年来，越来越多学者将机器学习或者深度学习方法应用于恶意安卓应用检测。Huijuan[3]等人提出通过提取权限、敏感的 API 监控系统事件和许可率等关键特性，采用整体旋转森林(rotation forest)构建一个经济有效的安卓应用检测模型，其检测效果优于支持向量机模型，但所选取的特征较少，检测准确率不高。Kun[4]等人通过提取权限、硬件功能和接收者动作等 122 个特征，使用多种机器学习分类器进行训练和测试，并使用随机森林分类器(random forest)获得较高的分类准确率，但由于未提取系统调用函数特征，导致误报率比一般防病毒程序高。Suman[5]等人提出通过提取权限和 API 等特征，使用逻辑回归(logistic regression)构建分类模型，并且通过删除低方差特征来优化特征，获得较好的分类效果。 Wei[6]等人提取了权限、intent、API 和硬件特性等 11 类共 3 万多个特征，然后使用支持向量机(SVM)根据特征对检测的重要性对特征进行排序，并集成 5 种机器学习分类器进行分类，实验结果表明该集成方法优于单一机器学习模型，然而该方法需要提取的特征较多，特征维度较大，检测效率相对较低。ElMouatez[7]等人提取应用程序 API 作为特征，使用卷积神经网络(CNN)来进行检测，实验表明该方法在多个数据集检测中具有较高的准确性。Zi[8]等人提出基于深度置信网络（DBN）的恶意安卓应用检测方法，通过提取安卓应用的权限和 API 函数作为特征，采用 DBN 进行训练和测试，从而减少学习过程中的人工干预。Wei[9]等人将卷积神经网络(CNN)与深度自编码网络(DAE)结合构建 CNN-DAE 检测模型，提取了 API、权限、硬件特性等多种特征来进行训练，该方法相比机器学习方法提高了检测精度，也相比 CNN 减少了训练时间。总体而言，深度网络检测方法的检测效率较低，需要较大的样本量。

针对上述文献中存在的安卓恶意检测中存在的效率低、特征维度过大和样本量不足等问题，本文提出一个结合卷积神经网络（CNN）和 catboost 算法的混合检测模型，同时具备了 CNN 的局部感知和降维后特征保持不变性的能力，以及 catboost 算法对样本量和特征要求不高，不易过拟合的优点。卷积神经网络的特征提取功能可以增强信号和降低向量维度，可以较好地解决特征筛选和特征维度过大的问题，提高检测效率；运用鲁棒性高和对样本数量要求不高的 catboost 模型作为分类层进行分类，可以解决样本量不足而影响分类精度的问题。另外，使用遗传算法对 catboost 进行快速的参数优化，进一步提高检测精确度和效率。

2 相关技术介绍

2.1 CNN神经网络模型

卷积神经卷网络（CNN）是一类包含卷积计算且具有深度结构的前馈神经网络[10]，主要包括以下两层：

2.1.1 卷积层

卷积层的功能是使用卷积核对输入数据进行特征提取，有规律地扫过输入特征，在感受野内对输入特征做矩阵元素乘法求和并叠加偏差量：

$$Z^{l+1}(i,j) = [Z^l \otimes w^{l+1}](i,j) \quad (1)$$

其中$i,j \in \{0,1,\cdots L_{l+1}\}$，$L_{l+1} = \frac{L_l+2p-f}{s_0}$。

2.1.2 池化层

池化层是对输入的对象进行抽象和降维，具有保持特征不变性和防过拟合作用。L_p池化是比较常用的一种池化方法，其一般表示形式为：

$$A_k^l(i,j) = \left[\sum_{x=1}^{f}\sum_{y=1}^{f} A_k^l \left(s_0 i + x, s_{0j} + y\right)^p\right]^{\frac{1}{p}} \quad (2)$$

其中 p 是预指定参数，当$p = 0$时，L_p池化在池化区域内取均值，被称为均值池化；当$p \to \infty$时，L_p池化在区域内取极大值，被称为极大池化[11]。

2.2 Catboost模型

Catboost 是一个基于梯度提升决策树的机器学习框架，它将所有样品的二进制特征存储于连续向量 B 中，叶子节点的值存储在大小为2^d的浮点数向量中。对于样本 x，建立一个二进制向量：

$$\sum_{i=0}^{d-1} 2^i \cdot B\big(x, f(t,j)\big) \quad (3)$$

其中$\mathrm{B}(x,f)$从向量 B 读取的样本 x 上的二进制特征 f 的值，而$\mathrm{f}(t,j)$从深度 i 上的第 t 棵树中的二进制特征的数目。向量以数据并行的方式构建，可以实现高达 3 倍的加速[12]。

2.3 遗传算法

遗传算法（Genetic Algorithm, GA）是一种通过模拟自然进化过程搜索最优解的方法[13]，具有内在的隐并行性和更好的全局寻优能力，其基本流程如图 1 所示。

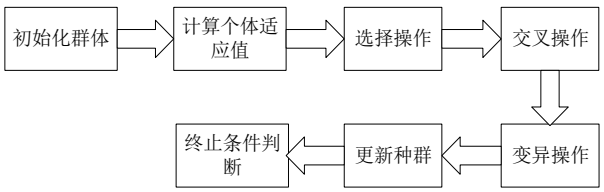

图 1　遗传算法基本流程

Fig.1　Workflow of genetic algorithm

3　模型设计与实现

3.1　CNN-catboost模型的网络结构

本文提出的 CNN-catboost 混合分类模型，继承了卷积神经网络提取局部特征能力强和降维后保留有有效信息效果较好的优势，以及 catboost 模型无需广泛数据训练就可以产生较好分类效果的特点，并利用遗传算法的全局寻优调参能力缩减调参用时。其模型结构如图 2 所示。

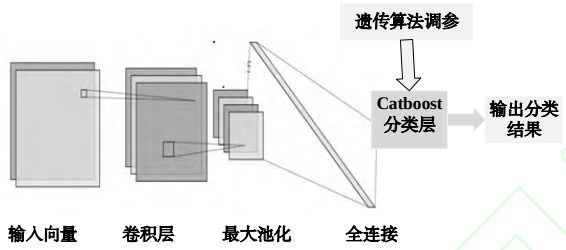

图 2　CNN-catboost 模型结构图

Fig.2　Structure of CNN-catboost model

在 CNN-catboost 模型中，首先输入层输入特征向量，卷积层对输入的特征向量进行提取和局部感知，利用卷积运算增强原始信号特征；然后再利用池化层对提取整合的特征进行下采样，在减少数据处理量的同时保留有用信息，减少冗余信息，特征维度降低，能促进最后一层分类模型的快速收敛，从而降低模型的训练时间，有效提高分类模型的泛化能力；另外，通过使用遗传算法对 catboost 进行参数优化以进一步提高 catboost 模型的分类准确率；最后在 catboost 层对特征向量进行训练和测试。

安卓应用的特征种类和数量较多，特征的选择和处理是构建泛化性强的安卓恶意应用检测模型的难点。将 catboost 应用于安卓恶意应用检测时，需要人工进行特征选择和过滤，其结果对 catboost 的最终分类效果有重要影响。而 CNN 模型具有自动学习特性，可以通过卷积层和池化层进行自动的特征优化，无需进行人工构造和过滤特征，可有效弥补 catboost 在特征处理方面的不足。但是，将 CNN 应用于分类时，存在对数据量要求高、学习过程长，训练速度慢的缺陷，而 catboost 的优势在于不易过拟合，训练速度快，无需大量数据进行训练即可获得高准确率的分类结果，正好可以改进 CNN 模型的上述问题。

Catboost 的参数选取会影响其分类结果的准确性，而且最优化的参数有助于巩固其鲁棒性。而遗传算法以多点和并行搜索寻找全局最优解，在快速确定最优参数群的同时避免陷入局部最优解。因此应用遗传算法对 catboost 进行调参，借助其良好的全局搜索能力，以较少的资源消耗和时间确定最优化参数，加快 catboost 的收敛速度。

综上所述，本文结合 CNN、catboost 和遗传算法的各自优势，利用 CNN 中卷积层和池化层自动提取和选择特征，自动抽取安卓应用的本质特征，将其作为 catboost 模型的输入向量，并使用遗传算法快速确定 catboost 模型的最优化参数，构造一个既避免人工选择过滤特征，又无需较多的数据样本就可以快速生成较好的检测效果的模型。

3.2　特征提取

CNN-catboost 模型的第一步是输入特征向量，因此需要特征提取。本文首先使用 androguard 工具提取到安卓应用的权限、API 类包、四大组件、intent 特征和硬件特性等各种静态特征；然后使用 apktool 工具将 APK 文件解码得到 smali 文件，进而提取 OpCode 特征[14]。

（1）权限特征：在一般情况下，安卓恶意应用申请的敏感权限比正常应用多，正常应用所申请的权限总量比恶意的多。本文选取了 Android6.0 版本的 25 个敏感权限和 33 个正常权限作为权限特征。

（2）API 特征：Android 应用的相关功能需要通过 API 调用来实现[15]。由于安卓 API 函数数量众多且每个 API 函数敏感高低难以统计，本文选取了 Android SDK 所提供的 150 个安卓 API 类作为特征。

（3）组件数目：Android 系统中的 4 种实用组件分别是 Activity、Service、BroadcastReceiver 和 ContentProvider。有些恶意家族应用为了保持恶意家族特性，会在同族软件中使用相同的服务名，所以本文提取这四种组件的数目作为特征。

（4）intent 特征：Intent-Filter 动作 action 用于描述意图要执行的行为[16]。比如 DIAL 拨打电话不需要打电话的权限，恶意应用在用户不知情的情况下可以越权拨打电话，因此本文选取了恶意应用最常用的 10 个 action 来作为特征。

（5）硬件特征：uses_feature 描述了安卓应用所

需要硬件特性，例如某恶意应用会申请 GPS 来窃取用户定位数据，所以本文提取用户最常使用的 27 个硬件特性作为特征。

（6）OpCode 特征：同族恶意应用变种往往具有高度相似的代码逻辑，并且可以由操作符序列进行表征。通过反编译 APK 得到 smali 文件中包含的 Dalivk 指令[17]，从中抽取反映程序语义的七大类核心指令集合 V、T、M、G、I、R、P，并去掉操作数，形成 OpCode 特征，如表 1 所示。

表 1　OpCode 特征核心指令集

Table 1　Core instruction set of OpCode feature

标签	指令类集	动作
V	Invoke 类指令集合，包括 INVOKE_ DIRECT, 等 15 种调用指令	调用
T	aget,iget 和 sget 这 3 类指令集合，包括 AGET_BOOLEAN 等 30 种指令	获取
M	move 类指令，包括 MOVE 等 13 种指令	移动
G	goto 类集合，包括 GOTO 三类指令集	跳转
I	if 类指令集，包括 IF_EQ 等 12 种指令	判断
R	return 类指令集合，包括 RETURN 等 4 种指令	返回
P	Aput、input 和 put 三类指令集合，包括 APUT_BOOLEAN 等 30 类赋值指令集合	赋值

3.3　特征预处理

对于安卓权限特征，将选取的 25 个敏感权限和 33 个正常权限依次排列组成一个特征向量，排列方式如图 3 所示。若某一位取 0 则表示该权限在 APK 中未出现，取 1 则反之。

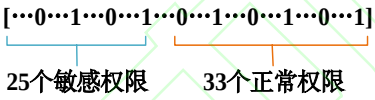

图 3　安卓权限排列示意图

Fig.3　Schematic diagram of Android permission arrangement

与敏感行为相关的安卓 API 函数大部分集中于 content，location 和 net 等 10 个类包中。首先将 140 个普通 API 类包则按照 Android 官方 API 文档的顺序排列在前，再将 10 个敏感类包排在后，构成一个 API 类包特征向量。一个 API 类包在该 APK 中出现的次数为该状态的取值，如图 4 所示。

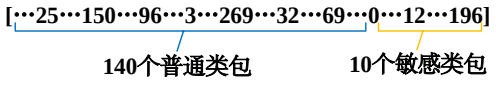

图 4　API 类包排列示意图

Fig.4　Schematic diagram of API packages arrangement

对于 Activity、Service、BroadcastReceiver 和 Content Provider 这 4 种组件，取每种组件的数量为特征；intent 特征中的动作所出现的次数即该向量值。

对于 27 个硬件特性特征，特性相关联的排列相邻，如 Camera/camera.Flash/camera.front 这几个排列在相邻，某特性若是被引用，则对应特征值为 1，否则为 0。

对于提取到的 OpCode 特征，建立一个 2-Gram 模型，模型中每个子集出现的次数则为特征向量的该状态值。

将以上 6 类共 298 个特征映射到向量空间，形成一个安卓应用的特征向量，如表 2 所示。

表 2　特征向量表示

Table 2　Representation of the feature vectors

特征	类型	数量	向量化	组合向量
权限	0-1	58	[…0…1]	
API 包	Number	150	[…66…120]	
组件数目	Number	4	[2,5,3,4]	该 6 类特征连
Intent	Number	10	[…5…8]	接构成 298 维
硬件特性	0-1	27	[…1…0]	向量
OpCode	2-Gram	49	[…75…150]	

3.4　模型的训练过程

CNN-catboost 混合模型的训练过程主要分为特征处理、分类与调参两个过程，如图 5 所示。

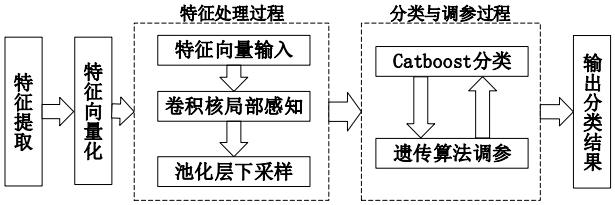

图 5　CNN−catboost 模型训练过程

Fig.5　Training procedure of CNN-catboost model

在特征处理过程中，由于特征向量中相似特性的特征相邻，值为 0 的元素较多，导致训练模型的时间较长，因此设置一个卷积层来聚合特征。因为特征向量为一维，所以卷积核的尺寸可以为1×2或者1×3，并且将卷积核的值都设为整数。特征向量经过卷积层处理后输入到池化层，池化层为最大池化，池化窗口大小为1×2，步长为 2，特征向量经过下采样后，其维度会至少降低为输入向量的一半。

在分类与调参过程中，首先将经过处理的特征向量作为 catboost 分类层的输入向量，使用默认参数进

行训练；然后运用遗传算法对 catboost 分类层进行调参，首先随机初始化种群，使用二进制编码将待优化参数转换为染色体；然后计算模型的 AUC 值作为个体适应值,来评定各个体的优劣程度；然后采用轮盘赌法选择出适应性强的个体，并对染色体进行单点交叉和基本位变异，保留其优秀基因和产生有实质性差异的新品种；最后更新种群，再次计算适应度。重复上述步骤直至找到最优参数。调参结束后再使用最优参数值对特征向量进行训练，最后输出分类结果。

4 实验结果与分析

本实验方案设计思路如下：首先将应用遗传算法进行调参，得到最优参数；然后在相同样本量和特征维度的前提下，保持卷积层和池化层相同，在分类层分别选取 catboost 算法和其他 5 种机器学习算法与 CNN 进行适配,共构成 6 种 CNN+机器学习算法混合模型，继而对比这 6 种混合模型在分类准确度方面的效果；之后将 CNN-catboost 模型与其他 10 种较有代表性的和新出现的模型进行比较；然后将 CNN-catboost 与单一机器学习模型、其它混合模型以及单一深度学习模型的训练时间进行比较讨论；紧接着将 CNN-catboost 与单一机器学习模型、其它混合模型以及单一深度学习模型在检测耗时和资源利用率方面进行比较和讨论；最后在未知类型的安卓应用数据集上检测各模型的分类性能。

4.1 实验环境、数据集选取和评估指标

在实验中，物理主机为 8GB 内存，处理器为 Intel Core i7-8750H，操作系统为 64 位 Windows 10。

实验所采集的数据样本源自加拿大网络安全研究所[1]和 VirusShare 网站[2]，共有 3861 个，其中良性样本包括杀毒软件、浏览器、通信软件、生活常用软件及管理类软件等多种良性应用，共 1690 个；恶意样本包括 2012 年至 2017 年的广告软件、勒索软件、恐吓软件、短信恶意软件及僵尸网络软件等多个家族的恶意应用，共 2171 个。

本文使用准确率（Accuracy），精确率（Precision），召回率（Recall），F1 值、AUC 值和检测耗时这几个标准进行作为实验结果判定指标，具体定义如下：

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+FP+TN+FN)} \times 100\% \quad (4)$$

$$\text{Precision} = \frac{TP}{TP+FP} \times 100\% \quad (5)$$

$$\text{Recall} = \frac{TP}{TP+FN} \times 100\% \quad (6)$$

$$\text{F1} = \frac{2\times Precision\times Recall}{Precision+Recall} \times 100\% \quad (7)$$

将恶意软件视为负例，良性软件视为正例，其中 *TP*：预测为正，判断正确；*FP*：预测为正，判断错误；*TN*：预测为负，判断正确；*FN*：预测为负，判断错误。AUC 被定义为 ROC 曲线（接收者操作特征）与横坐标所组成的面积，可以直观地评价分类器的性能，其值越大代表模型分类能力越好。

4.2 基于遗传算法的参数值优化实验

分类模型的参数值会直接影响到分类的准确性和效率。由于 catboost 减少了对广泛超参数调优的需求，所以主要选取对模型影响比较大的 4 个参数进行讨论：最大树数 iterations、学习率 learning_rate、树深 depth 和数值特征分割数 border_count。其他参数设置为默认值。根据遗传算法的特点，结合常用的参数经验来初始化遗传算法参数[13]，设置初始群体大小为 50，交叉概率为 0.6，变异概率为 0.01，以此产生新的基因，同时避免陷入单纯的随机搜索。

在 catboost 算法的默认参数中，最大树为 1000，学习率为 0.1，树深为 6，数值分割数为 256；而本文运用遗传算法的参数值进行优化尝试，得到以下最优参数：最大树为 180，学习率为 0.242，树深为 8，数值分割数为 250。分别将上述两种参数进行检验，结果如图 6 所示。可以发现，最优参数在各项分类评估指标中都优于默认参数。

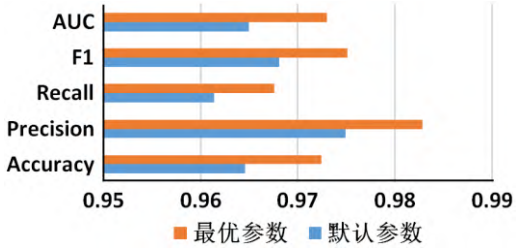

图 6 最优参数与默认参数对比

Fig.6 Comparison between optimal parameters and default parameters

4.3 CNN-catboost与其他CNN+机器学习模型对比实验

在保持卷积层和池化层相同的前提下，将 CNN-catboost 模型与其他 5 种 CNN+机器学习算法混合模型进行比。表 3 列出了各混合模型中机器学习算法的重要参数。

表 3 混合模型中机器学习算法的主要参数

[1]https://www.unb.ca/cic/datasets/android-adware.html
[2]https://virusshare.com/

Table 3　Dominating parameters of machine learning algorithm in hybrid model

模型	主要参数
CNN-xgboost	learning_rate=0.1, nestimators=1000, max_depth=8, min_child_weight=4, gamma=0, subsample=0.8
CNN-decisiontree	criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1
CNN-randomforest	n_estimators=10, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1
CNN-logisticregression	penalty='l2', dual =False,tol=0.0001, C=1.0, fit_intercept=True
CNN-ensemble	C=0.25, kernel='linear' ,K=9, alpha=1.0,max_features='auto'

实验结果如表 4 所示，可以发现 CNN-catboost 模型比其他 CNN+机器学习算法具有更优的分类准确度。

表 4　CNN-catboost 与其他 CNN+机器学习模型实验对比

Table 4　Comparison between the CNN-catboost and other CNN+ machine learning models

Model	评估指标			
	Accuracy	Precision	Recall	F1
CNN-catboost	0.9724	0.9828	0.9676	0.9751
CNN-xgboost	0.9655	0.9720	0.9660	0.9690
CNN-decisiontree	0.9525	0.9554	0.9598	0.9576
CNN-randomforest	0.9560	0.9745	0.9460	0.9601
CNN-logisticregression	0.9275	0.9134	0.9614	0.9368
CNN-ensemble	0.9525	0.9744	0.9398	0.9567

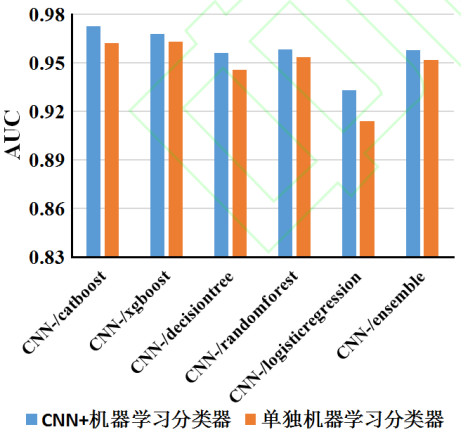

图 7　CNN+机器学习与单一机器学习实验 AUC 值对比

Fig.7　Comparison of experimental AUC values between CNN+ machine learning and single one

以 AUC 值为评估指标，将 CNN-catboost 与 catboost、CNN-xgboost 与 xgboost,、CNN-decisiontree 与 decision tree、CNN-randomforest 与 random forest、CNN-logisticregression 与 logistic regression 、CNN-ensemble 与 ensemble 分别进行对比。其中 catboost、xgboost,decision tree、random forest、logistic regression 等算法的参数分别与其相对应的 CNN+机器学习算法混合模型中的机器学习算法的参数一致。在图 7 中，CNN-catboost 与 catboost 简化表述为 CNN-/catboost，其余类似。由图 7 可发现，CNN+机器学习分类器混合模型的分类效果比单一机器学习分类器要好，而 CNN-catboost 模型的 AUC 值最高，显示该模型具有较好的性能。

4.4　CNN-catboost模型与其他多种模型对比实验

将 CNN-catboost 模型分别与 catboost，xgboost 模型，决策树 decision tree，多层感知机 MLP[18]，random forest 模型[4]，logistic regression 模型[5]，ensemble 模型[6]，CNN 模型[7]，DBN 模型[8]和 CNN-DAE 模型[9]等分类模型进行比较，实验结果如表 5 所示，可知 CNN-catboost 的分类效果优于其他模型。至于 CNN 等神经网络模型的分类效果不够理想，原因在于样本数量不够，造成了过拟合。

表 5　CNN−catboost 模型与其他模型实验对比

Table 5　Comparison between the CNN-catboost and other models

Model	评估指标			
	Accuracy	Precision	Recall	F1
CNN-catboost	0.9724	0.9828	0.9676	0.9751
catboost	0.9664	0.9735	0.9660	0.9697
xgboost	0.9638	0.9704	0.9645	0.9674
decision tree	0.9431	0.9449	0.9537	0.9493
MLP[18]	0.9620	0.9552	0.9589	0.9570
random forest[4]	0.9534	0.9714	0.9444	0.9577
logisticregression[5]	0.9180	0.9084	0.9490	0.9283
ensemble[6]	0.9482	0.9711	0.9351	0.9528
CNN[7]	0.9465	0.9481	0.9295	0.9387
DBN[8]	0.9594	0.9640	0.9432	0.9535
CNN-DAE[9]	0.9431	0.9441	0.9256	0.9347

4.5　训练时间对比实验

由于 decision tree 模型、random forest 模型[4]、logistic regression 模型[5]、ensemble 模型[6]等机器学习模型复杂度低，所以训练耗时比较小，但他们的分类准确性也比较差，因此在本实验中不列入训练时间比较范围。本文将 CNN-catboost 模型分别与分类准确率较高的模型，包括混合模型（如 CNN-xgboost、

CNN-DAE[9]），以及单一机器学习模型（如 catboost）和单一深度学习模型（如 CNN[7]、MLP[18]和 DBN[8]）等进行训练时间方面的比较，结果如表 6 所示。

从表 6 看出，CNN-catboost 的训练时间与 CNN-xgboost 相当，比其它混合模型以及单一机器学习和深度学习模型的训练时间少。经分析，原因如下：一方面，在本文的 CNN-catboost 模型中，经过卷积层和池化层处理的特征向量，在保留了特征的有效信息的同时，也明显降低了特征维度；另一方面，catboost 算法对训练次数要求低，分类预测速度较快，而神经网络由于其非线性，梯度噪音等原因导致优化步骤多，导致训练时间较长，因此导致 CNN-catboost 的训练开销低于混合模型和单一深度学习模型和机器学习模型。

表 6　各模型的训练时间

Table 6　Training time of each model

Model	Time/s
CNN-catboost	9
CNN-xgboost	9
CNN-DAE[9]	28
catboost	14
CNN[7]	138
MLP[18]	35
DBN[8]	54

4.6　模型检测耗时和资源利用率对比实验

将 CNN-catboost 模型应用于安卓应用检测，分析其在检测耗时、CPU 使用率和内存使用率方面的表现。从 4.1 节所述的数据集中选取 511 个良性应用和 649 个恶意应用（共 1159 个）进行实验。继续选取其他分类准确率较高的模型与 CNN-catboost 进行对比，包括混合模型 CNN-xgboost、CNN-DAE[9]，单一机器学习模型 catboost 和单一深度学习模型 CNN[7]、MLP[18]和 DBN[8]等，结果如表 7 所示。

表 7　检测耗时和资源利用率对比

Table 7　Comparison of detection time and resource utilization

Model	Time/s	CPU 使用率/%	内存使用率/%
CNN-catboost	0.037	11.9	2.4
CNN-xgboost	0.048	28.3	3.7
CNN-DAE[9]	0.072	44.9	3.2
catboost	0.054	10.3	2.6
CNN[7]	0.190	73.9	3.8
MLP[18]	0.062	28.3	2.8
DBN[8]	0.081	21.3	3.6

由表 7 可知，CNN-catboost 的检测时间和内存使用率优于其他模型，CPU 使用率也优于大部分模型，说明将 CNN-catboost 应用于安卓应用检测时具有检测耗时短、资源耗费少的优点。

4.7　未知类型应用检测实验

在本次实验中，利用 CNN-catboost 模型对随机搜集的未知类型的安卓应用进行检测，以检验其在实际应用中的效果。首先从各大安卓应用市场下载 1007 个未知类型的安卓 APK 包，然后上传至 VirusTotal[3]在线查毒网站进行分类检测，检测结果作为本次实验的检测基准。VirusTotal 网站对此 1007 个样本进行检测的结果为：良性样本 751 个，恶意样本 256 个。继续使用 4.6 节中提及的其它模型所得的检测结果进行对比，实验结果如表 8 所示，其中 Recall1 为良性样本召回率，Recall2 为恶意样本召回率。

表 8　基于未知类型安卓应用数据集的各模型对比

Table 8　Comparison of models based on unknown Android app dataset

Model	TP	FN	TN	FP	Recall1	Recall2
CNN-catboost	725	26	244	12	96.53	95.31
CNN-xgboost	720	31	241	15	95.87	94.14
CNN-DAE[9]	702	49	228	28	93.47	89.06
catboost	723	28	240	16	96.27	93.75
CNN[7]	711	40	224	32	94.67	87.50
MLP[18]	716	35	236	20	95.34	92.18
DBN[8]	719	32	230	26	95.74	89.84

由表 8 可知，在未知类型的应用检测中，相比于其他模型，CNN-catboost 可较好地区分良性和恶意应用，误判率比较低，切合对各种安卓应用进行安全检测的实际需求。

5　结论

本文提出一种有效地将卷积神经网络特征提取和降维能力以及 catboost 模型鲁棒性强的分类能力结合的混合模型 CNN-catboost，该模型不仅兼具卷积神经网络可以增强特征信号，减少冗余信息的优点，同时还融合了 catboost 模型的低数据量要求即可获得较好分类结果的优势，通过运用遗传算法进行调参进一步提升 catboost 模型的分类准确率。经过实验验证，该模型在提高了恶意安卓应用检测准确率的同时，也有效地缩短了训练时间，因而具有一定的实践应用价值。

[3]https://www.virustotal.com/

鉴于深度学习方法对安卓恶意软件检测具有良好的发展前景，因此在下一步工作中将继续优化深度学习模型，进一步提高恶意安卓应用检测精度和效率。

参考文献：

[1] 李旭. 基于应用分类的安卓恶意应用检测模型[D].广州大学,2019. Li Xu. Android Malicious Application Detection Model based on Application Classification[D]. Guangzhou University,2019.

[2] Li L, Bissyandé T F, Papadakis M, et al. Static analysis of android apps: A systematic literature review[J]. Information and Software Technology, 2017, 88: 67-95.

[3] Zhu H J, You Z H, Zhu Z X, et al. DroidDet: effective and robust detection of android malware using static analysis along with rotation forest model[J]. Neurocomputing, 2018, 272: 638-646.

[4] Wang K, Song T, Liang A. Mmda: Metadata based malware detection on android[C]//2016 12th International Conference on Computational Intelligence and Security (CIS). IEEE, 2016: 598-602.

[5] Suman R, Ravi U. An Android Malware Detection Technique Based on Optimized Permissions and API[C]//2018 International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore: IEEE, 2018: 258-263.

[6] Wang W, Li Y, Wang X, et al. Detecting Android malicious apps and categorizing benign apps with ensemble of classifiers[J]. Future Generation Computer Systems, 2018, 78: 987-994.

[7] Karbab E M B, Debbabi M, Derhab A, et al. MalDozer: Automatic framework for android malware detection using deep learning[J]. Digital Investigation, 2018, 24: 48-59.

[8] Wang Z, Cai J, Cheng S, et al. DroidDeepLearner: Identifying Android malware using deep learning[C]. ieee sarnoff symposium, 2016: 160-165.

[9] Wang W, Zhao M, Wang J. Effective android malware detection with a hybrid model based on deep autoencoder and convolutional neural network[J]. Journal of Ambient Intelligence and Humanized Computing, 2019, 10(8): 3035-3043.

[10] Krizhevsky A, Sutskever I, Hinton G E, et al. ImageNet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.

[11] Milletari F, Navab N, Ahmadi S A. V-net: Fully convolutional neural networks for volumetric medical image segmentation[C]//2016 Fourth International Conference on 3D Vision (3DV). IEEE, 2016: 565-571.

[12] Weirong L, Kunyuan D, Xiaoyong Z, et al. A Semi-Supervised Tri-CatBoost Method for Driving Style Recognition[J]. Symmetry, 2020, 12(3), 1-18.

[13] Jiaqing W, Yan Z, Tung-Shou C, et al. Optimum feature selection based on genetic algorithm under Web spam detection[J]. Journal of Computer Applications, 2018.

[14] Kang B J, Yerima S Y, McLaughlin K, et al. N-opcode analysis for android malware classification and categorization[C]//2016 International Conference on Cyber Security and Protection of Digital Services (Cyber Security). IEEE, 2016: 1-7.

[15] Muslukhov I, Boshmaf Y, Beznosov K. Source Attribution of Cryptographic API Misuse in Android Applications[C]// Proceedings of the 2018 on Asia Conference on Computer and Communications Security. ACM, 2018: 133-146.

[16] 肖卫, 张源, 杨珉. 安卓应用软件中 Intent 数据验证漏洞的检测方法[J]. 小型微型计算机系统, 2017, 38(4): 813-819.

Xiao Wei, Zhang Yuang,Yang Min. Find Intent Data Validation Vulnerability in Android Application Automatically and Efficiently[J]. Journal of Chinese Computer Systems, 2017, 38(4): 813-819.

[17] Varsha M V, Vinod P, Dhanya K A. Identification of malicious android app using manifest and opcode features[J]. Journal of Computer Virology and Hacking Techniques, 2017, 13(2): 125-138.

[18] Sumsion G Rex, Bradshaw Michael S, Hill Kimball T, Pinto Lucas D G, Piccolo Stephen R. Remote sensing tree classification with a multilayer perceptron[J]. Peer J, 2019, 7.