

DOI:10.19651/j.cnki.cmt.2107384

基于系统调用的手机 APP 能耗漏洞检测*

于兴磊¹ 朱正伟¹ 朱晨阳² 诸燕平²

(1.常州大学 微电子与控制工程学院 常州 213000; 2.常州大学 计算机与人工智能学院 常州 213000)

摘要: 针对传统研究中 APP 能耗漏洞模型存在尾部能耗与应用对象限制的问题,构建基于系统调用的能耗漏洞模型。首先使用集合交叉递归特征消除法选择影响每个类别 APP 能耗的重要系统调用作为特征,提高特征细化粒度。然后为每个类别 APP 构建多个回归模型,通过比较不同模型的平均绝对误差与决定系数,选择线性核支持向量机回归作为分类 APP 的能耗模型。最后基于测试集比较集合交叉递归特征消除法与交叉递归特征消除法所构模型的平均绝对误差,结果表明集合交叉递归特征消除法所构模型精度最多提高 4.4%,同时基于测试集比较分类模型与未分类模型的平均绝对误差,结果表明分类模型精度最多提高 6.7%,并且分类模型能准确检测出 APP 历史版本的能耗漏洞。

关键词: 能耗漏洞;机器学习模型;分类与未分类模型;系统调用;特征选择

中图分类号: TP311.5 **文献标识码:** A **国家标准学科分类代码:** 510.4030

Detection of phone's application energy bug based on system call

Yu Xinglei¹ Zhu Zhengwei¹ Zhu Chenyang² Zhu Yanping²

(1. School of Microelectronics and Control Engineering, Changzhou University, Changzhou 213000, China;

2. School of Computer Science and Artificial Intelligence, Changzhou University, Changzhou 213000, China)

Abstract: Aiming at the problem of tail energy consumption and application object limitation in APP energy bug model in traditional research, a system call based energy bug model is constructed. First, mix-cross-recursive feature elimination method is used to select important system calls that affect the energy consumption of each category of APP as features to improve the feature refinement granularity. Then, multiple regression models are constructed for each category of APP. By comparing the average absolute error and coefficient of determination of different models, linear kernel support vector machine regression is selected as the energy bug model of the classified APP. Finally, based on the test set, the average absolute error of the model constructed by the mix cross recursive feature elimination method and the cross recursive feature elimination method is compared. The result shows that the accuracy of the model constructed by the mix recursive feature elimination method is increased by up to 4.4%. At the same time, the classification model is compared with the classification model based on the test set. The average absolute error of the unclassified model shows that the accuracy of the classification model is increased by up to 6.7%, and the classification model can accurately detect the energy bug of the historical version of the APP.

Keywords: energy bug; machine learning models; classified and unclassified models; system calls; feature selected

0 引 言

为提高手机使用时长,现有研究集中于手机电池的研发^[1-2],然而电池研发周期过长,因此一种可行的方法是检测手机 APP 的能耗漏洞,开发人员对 APP 做出代码优化,消除能耗漏洞使能耗降低,延长手机使用时长。

传统 APP 能耗漏洞检测模型的研究集中于建立基于

利用率与基于指令的能耗漏洞模型。基于利用率的模型^[3-5]通过统计手机组件的利用率和相应的能耗模拟 APP 的能耗,然而网卡、GPS 等组件在完成的任务后的一段时间内处于高功率模式^[6],影响了组件与组件实际能耗之间建立的功耗系数导致模型精度降低。基于指令的模型^[7-8]分析了路径与 APP 的能耗事件,利用回归分析计算源代码行的能耗,得到手机运行的总能耗,然而这类基于指令的模型

收稿日期:2021-07-25

* 基金项目:国家自然科学基金(61801055)、常州市重点研发计划(CJ20210123)项目资助

受到 APP 对象限制,基于 Java 指令的模型不能检测由 C++ 构建的 APP 的能耗漏洞。针对上述问题,Aggarwal 等^[9]提出基于系统调用的能耗漏洞模型,该模型不使用组件利用率与指令统计能耗,通过研究系统调用计数与能耗之间的关系,表明能耗的差异具有统计学意义。Zhu 等^[10]研究了影响能耗的系统调用的重要性,构建并对比不同机器学习模型的预测精度。文献[9-10]在训练模型过程中,选择影响所有类别 APP 能耗的系统调用作为特征,并对所有类别 APP 构建未分类情况下的能耗模型,然而该未分类模型所选择的系统调用缺乏细化粒度,影响了模型精度。

针对以上问题,本文在文献[10]的研究基础上,提出集合交叉递归特征消除法(mix recursive feature elimination cross validation, MREFCV),基于自动化测量平台收集的数据集分别为每个类别 APP 选择特征,细化特征的选择。

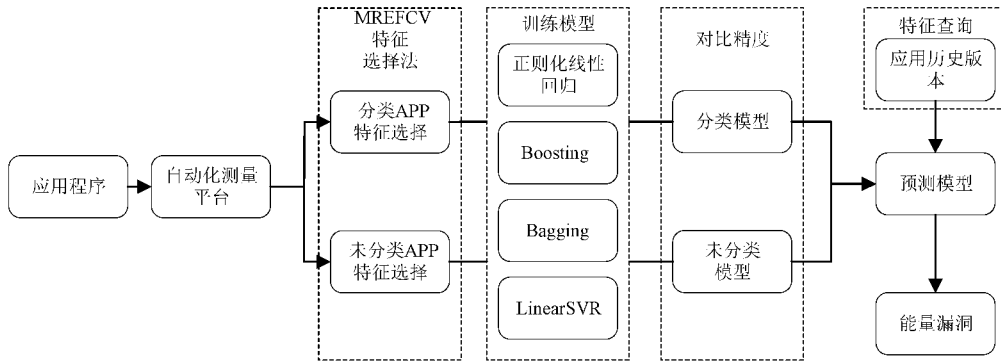


图 1 总体框架

1.1 正则化的线性回归

给定训练集 $(x_i, y_i), i=1, 2, \dots, n$ 。其中 x_i 为不同特征对应的数量, y_i 为对应的能耗。线性回归通过学习多个变量的线性组合关系来拟合实际值,如式(1)。为确定变量的系数向量 θ ,最小化式(2)中线性回归的目标函数。

$$f(x_i) = \theta^T x_i \tag{1}$$

$$L(y_i, f(x_i)) = \frac{\sum_{i=1}^n (y_i - f(x_i))^2}{n} \tag{2}$$

线性回归模型通常学习过多的噪声,模型存在过拟合的风险。正则化能够降低模型的复杂度,避免模型预测出现过拟合,使用岭回归(Ridge)和套索回归(Lasso)增加正则化惩罚项来平衡线性回归模型的复杂度,Ridge 在式(2)目标函数中加入了 L_2 正则化。

$$\theta = \operatorname{argmin}_{\theta} [(y_i - \sum_{j=1}^n \theta_j x_{ij})^2 + \lambda \sum_{j=1}^n \theta_j^2] \tag{3}$$

Ridge 和 Lasso 的数学区别是在式(3)中,Lasso 使用 $L_1(\sum |\theta_j|)$ 正则化^[11],Ridge 使用 $L_2(\sum \theta_j^2)$ 正则化^[11]。

使用随机梯度下降法^[12](stochastic gradient descent, SGD)来最小化线性回归模型的正则化损失函数。为了使式(2)中线性模型的损失函数最小化,SGD 算法沿着梯度的反方向进行系数的更新,若损失函数是凸函数,则找到

然后为每个类别 APP 建立基于系统调用的机器学习模型,讨论分类与不分类模型在测试集的预测精度,最后收集不同历史版本的 APP,执行相同的 UI 操作,通过观察分类模型预测的能耗趋势图,若出现较大的波动,则表明该版本的 APP 存在能耗漏洞。

1 能耗模型

本文为视频类、音乐类、小说类以及未分类(视频类、音乐类和小说类的集合)构建多个机器学习回归模型,包括为正则化的线性回归、线性核支持向量机回归(LinearSVR)、Boosting、Bagging,可以较好地捕捉系统调用与能耗之间的线性与非线性关系。通过对比不同模型的精度,为视频类、音乐类、小说类、未分类选择能耗模型。最后对比分类模型与未分类模型在测试集的精度。图 1 为总体流程。

全局最优解,若不是,则找到局部最优解。式(4)为权重的更新过程,其中 β 为学习率。

$$\theta := \theta - \beta \frac{\partial}{\partial \theta} (y_i - \sum_{i=1}^n \theta \cdot x_i)^2 \tag{4}$$

1.2 LinearSVR

给定训练集 (x_i, y_i) ,其中 $i=1, 2, \dots, n$ 。支持向量机回归^[13](support vector regression, SVR)假设能容忍预测值 $f(x_i)$ 与实际值 y_i 最多有 ϵ 的偏差,即只有当 $f(x_i)$ 与 y_i 的绝对误差大于 ϵ 时计算损失,其中损失值定义为残差 ζ_i 与 ζ'_i 。在 SVR 中,目标是式(5)的损失函数,这里 C 是惩罚项参数。

$$\frac{1}{2} \|\theta\|^2 + \sum_{i=1}^n C(\zeta_i + \zeta'_i) \tag{5}$$

SVR 使用线性超平面来分离数据进行预测,但是数据分布可能是非线性的,为了解决这个问题,使用核函数将数据中非线性特征空间映射到高纬度的特征空间,然后使得数据样本在高纬度中线性可分。

1.3 Boosting 算法

Boosting 算法通常将几个弱学习器组合之后产生一个强学习器组合之后产生一个强学习器,自适应增强算法^[14](Adaboost)是常用的 Boosting 算法之一。Adaboost 首先基于初始训练集训练一个基学习器,根据基学习器在训练

样本中的最大误差和每个样本的相对误差,得到基学习器的权重,并更新下一轮的训练样本分布,如此反复,直到基学习器数量达到指定数目 K ,最终将 K 个学习器进行加权结合。与 Adaboost 不同的是,梯度提升决策树^[15](GBDT)学习器直接由基学习器相加求和。GBDT 构建 K 棵回归树,通过每颗树的预测值与实际值的残差值来计算下一颗回归树,经过 K 次迭代,把每颗回归树相加得到最终的回归树。

1.4 Bagging 算法

Bagging 方法构建几个学习器然后对学习器预测结果进行平均。随机森林(RandomForest)^[16]从特征空间 F 与训练数据 (x_i, y_i) 中随机选择特征和样本训练 K 个树模型,将测试集数据 x_m 输入每颗树 $f_k(x)$ 中得到 K 个预测值,对 K 棵树的预测值取平均,最终得到预测值 y_m 。

2 实验设置

2.1 UI 脚本

为了避免人为执行应用场景导致采集手机能耗数据出现误差,使用 Monkeyrunner 模拟人为操作 APP 的执行场景,分别为视频类、音乐类、小说类的 APP 编写通用的 UI 脚本,在 UI 脚本中设计随机模拟 APP 的不同执行场景。

视频类 UI 脚本:执行随机拖拽行为 N 次,切换到不同类别的分区。向下拖拽 M 次,APP 界面向下移动。随机点击屏幕坐标,实现随机播放视频的功能。观看 45 s 后退出。将上述行为重复 4 次。

音乐类 UI 脚本:点击“分类”目录的坐标,循环执行随机的拖拽行为 K 次,切换到不同音乐风格的类别目录。随机点击屏幕坐标,实现随机选择音乐专辑的功能。进入音乐专辑内,循环执行随机的拖拽行为,进行上下文切换不同音乐内容,随机点击一首歌听 30 s。最后将上述行为重复 4 次。

小说类 UI 脚本:进行随机拖拽行为 J 次,切换到不同小说风格的分区。向下拖拽 L 次,将 APP 界面向下拉。随机点击屏幕坐标,实现随机观看小说的功能。进入小说后,执行切换一下页的行为 10 次。返回到分区。将上述行为重复 4 次。

2.2 自动化测量平台

为获取手机能耗,选择小米 8 作为目标设备,拆卸目标设备的电池,连接稳压电源和 QWAY USB 测量仪。该测量仪提供收集目标设备实时能耗与运行时长,选择荣耀 v8 安装此 APP。自动化测量平台收集数据的步骤如图 2 所示,荣耀 v8 运行 Monkeyrunner 编写的数据采集脚本记录小米 8 运行 UI 脚本期间的能耗与运行时长采集。

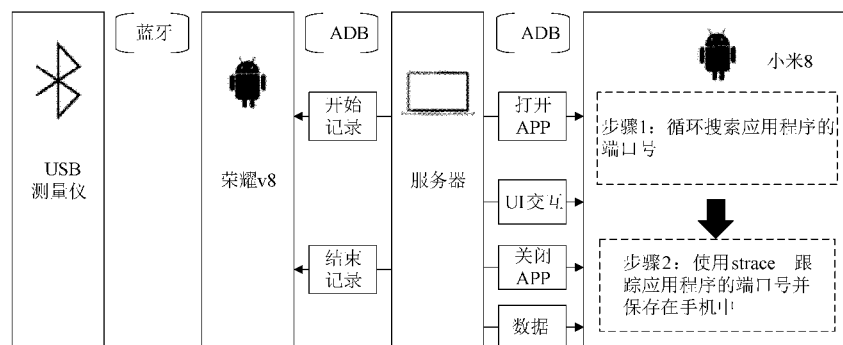


图 2 自动化测量平台收集

小米 8 中运行采集系统调用的脚本,当 APP 开启后,启动 strace 工具跟踪 APP 的端口号,当 APP 结束执行场景的模拟后,停止 strace 工具对系统调用的收集。

2.3 实验数据集

豌豆荚拥有开放的 APP 资源并支持下载不同应用历史版本。从豌豆荚为小说类、音乐类、视频类挑选 4 个 APP,分别为黑猫小说、七猫小说、番茄小说、起点小说、网易云音乐、酷我音乐、QQ 音乐、荔枝 FM、秒拍、Bilibili、西瓜视频、皮皮虾。基于自动化测量平台将每个 APP 运行 200 次,收集系统调用、运行时长以及能耗作为小说类、音乐类、视频类的训练集 D_{novel} 、 D_{music} 、 D_{media} 与测试集 T_{novel} 、 T_{music} 、 T_{media} 。其中未分类模型的训练集为所有类别训练集的集合 $D_{all} = D_{novel} + D_{music} + D_{media}$,未分类模型的测试集为所有类别测试集的集合 $T_{all} = T_{novel} + T_{music} + T_{media}$ 。

在各个类别的测试集上对比 MREFCV 与 REFCV 所特征构建的模型的精度,并对比分类与未分类模型的预测精度。数据集[9]对于不同类别 APP 的采集分散,每个类别只有 1~2 个 APP,且数据量较小,共有 474 组数据。与数据集[9]相比,本文数据集收集了多个不同类别的 APP,并多次测量,共有 2 400 组数据。

3 特征细化

未分类模型将同时影响每个类别 APP 能耗的系统调用作为特征,却排除影响某一类别 APP 能耗却不影响其他类别 APP 能耗的系统调用作为特征,这降低了模型在预测该类别 APP 能耗时的精度。为了提高模型精度,对每个类别 APP 筛选特征并构建模型。本节使用 MREFCV 对音乐类、视频类、小说类分别进行特征的筛选。

3.1 MREFCV 算法

算法 1 展示了从众多特征中选择重要特征的过程,先使用随机种子将数据集分割成训练集与测试集,将特征空间 F 分为 $2^N - 1$ 个特征子集 S_i ,把每个特征子集放入基模型中训练,选择在交叉验证中使基模型精度最高的特征子集作为 f_c ,重复上述步骤 M 次,选取在每次循环中都出现在 f_c 中的特征作为影响能耗的关键特征集 f 。

算法 1: MREFCV 算法

```

1) 初始化特征集  $F$ 
   迭代次数:  $M$  次
   交叉验证:  $K$  次
2) 初始化字典:  $f_c, f$ 
3) for  $m = 1$  to  $M$  do
   1. for  $k = 1$  to  $K$  do
     a) 将训练集随机分为训练集  $d_{train}$  和测试集  $d_{test}$ 
     b) 将  $F$  分为  $2^N - 1$  个子集  $S_i, i = 1, 2, \dots, N$ 
     c) 将  $S_i$  放入基模型中
   2. 计算基模型的精度,选择使模型精度最高的  $S_i$ ,将  $S_i$  添加到字典  $f_c$  中,其中  $j = 1, 2, \dots, M$ 
        $f_c[j] = S_i$ 
   end for
   end for
4) 选择在  $M$  次迭代中出现在  $S_i$  中的特征
    $f \leftarrow f_c[1] \cap f_c[2] \cap \dots \cap f_c[M]$ 

```

为消除特征数据之间的量纲影响,使用 z -score 归一化使特征数据符合正态分布。 f_i 是 f 的其中一个特征,数据为 $[a_1, a_2, \dots, a_n]$, $\sigma(a)$ 为标准差, a 为平均数,特征数据归一化如式 (8) 所示,归一化后的 f_i 特征数据为 $[b_1, b_2, \dots, b_n]$ 。

$$a = \frac{1}{n} \sum_{i=1}^n a_i, \quad \sigma(a) = \sqrt{\frac{1}{n} \sum_{i=1}^n (a_i - a)^2}$$

$$b_i = \frac{a_i - \bar{a}}{\sigma(a)} \quad (6)$$

3.2 特征分析

通过 MREFCV 对每个类别的特征进行筛选,得到小说类的特征个数为 40 个,音乐类的特征个数为 13 个,视频类的特征个数为 16 个,未分类的特征个数为 22 个。小说类、音乐类、视频类与未分类的特征种类差异个数为 23、14 个、10 个。由此可见,未分类模型的特征并不能包含影响每个类别的重要特征,表 1 为分类与未分类特征种类差异个数。

表 1 分类与未分类的特征种类差异个数

类别	个数
小说	23
音乐	14
视频	10

4 实验结果分析

4.1 超参数调整

选择特征并归一化后,对分类与未分类 APP 建立回归模型。回归模型中超参数的调整能够提升模型的精度并使目标函数最小化,选取到最优的模型。本文使用交叉验证和网格搜索法获取每个模型的最佳参数集。网格搜索法排列组合所有可能取值的参数集,每组参数集使用交叉验证来评估得分,这里使用决定系数 (R^2) 作为交叉验证的得分,当 R^2 越接近 1 时,表明模型拟合越好。超参数调整具体步骤如下:

步骤 1): 将超参数进行排列组合,生成不同的参数集。

步骤 2): 将训练集数据分为 K 个集合,选择 $K - 1$ 个集合作为训练集 d_{train} ,剩下的集合作为测试集 d_{test} 。

步骤 3): 将每个参数集放入模型中基于训练集 d_{train} 训练,并在测试集 d_{test} 中测试,得到模型在 d_{test} 中的得分 R^2 。

步骤 4): 重复步骤 1~3 K 次,直到每个集合都作为测试集,计算参数组合的平均 R^2 。

步骤 5): 选择 R^2 最高的参数集作为模型最佳的参数集。

4.2 模型评估

首先分别对小说类、音乐类、视频类与未分类 APP 构建基于 MREFCV 和 REFCV 的模型,并对比 MREFCV 与 REFCV 构建模型的精度,然后对比小说类、音乐类、视频类与未分类模型的精度。首先使用随机种子将每个类别的训练集 D_{novel} 、 D_{music} 、 D_{media} 分为训练集 t_1 以及测试集 t_2 ,模型基于 MREFCV 选定的特征与最佳的参数集在训练集 t_1 训练,然后在测试集 t_2 中计算每个机器学习模型的平均绝对误差 (mean absolute error, MAE) 与 R^2 ,为了平衡随机种子的影响,用随机种子重复分析 D_{novel} 、 D_{music} 、 D_{media} 10 次,计算模型在 10 次循环中的平均 MAE 与平均 R^2 。观察表 2 中不同类别的各个模型的 R^2 与 MAE,其中 LinearSVR 的 R^2 最高,并且 MAE 最小,这表明模型的拟合度或者是预测误差方面,LinearSVR 表现都比其他模型好,因此选择 LinearSVR 模型作为每个类别的能耗模型。如上述步骤,使用 REFCV 选择特征并构建模型,最终通过比较,选择 LinearSVR 模型作为每个类别的能耗模型。

基于测试集 T_{novel} 、 T_{music} 、 T_{media} 、 T_{all} 分别对比基于 MREFCV 所选特征训练的模型与 REFCV 所选特征训练的模型精度。表 3 中列出了每个类别中基于 MREFCV 构建的模型与基于 REFCV 构建的模型误差对比,结果表明 MREFCV 特征选择法有效提高了模型的精度。

在测试集 T_{novel} 、 T_{music} 、 T_{media} 分别对比分类与未分类模型的精度。表 4 中小说类模型在测试集 T_{novel} 的 MAE 远远小于未分类模型在测试集 T_{novel} 的 MAE,并且精度提高了 2.9%~5.3%。音乐类模型在测试集 T_{music} 中的精度略微提升。视频类模型在测试集 T_{media} 的 MAE 小于未分

表 2 基于 MREFCV 构建的多个模型对比

模型	未分类		音乐类		小说类		视频类	
	R^2	MAE/J	R^2	MAE/J	R^2	MAE/J	R^2	MAE/J
Lasso	0.810	16.82	0.692	18.10	0.813	20.41	0.707	18.91
Ridge	0.807	16.80	0.689	18.15	0.817	19.78	0.707	18.92
惩罚项 Lasso 的 SGD	0.805	17.09	0.692	18.24	0.818	19.73	0.704	18.64
惩罚项 Ridge 的 SGD	0.805	16.97	0.688	19.09	0.818	19.94	0.704	18.67
LinearSVR	0.823	15.91	0.699	17.57	0.823	19.35	0.708	17.50
Adaboost	0.755	21.93	0.520	25.21	0.793	22.19	0.518	26.62
RandomForest	0.804	18.03	0.605	21.67	0.752	21.64	0.523	24.84
GBDT	0.489	38.90	0.504	27.31	0.815	20.34	0.440	28.95

表 3 MREFCV 与 REFCV 所训练模型在测试集的

MAE 对比

	未分类	音乐类	小说类	视频类
	(MAE/J)	(MAE/J)	(MAE/J)	(MAE/J)
MREFCV	31.76	14.10	17.88	31.05
REFCV	38.42	16.09	22.04	43.53
测量范围/J	250~580	280~530	250~460	250~580
精度提升/%	1.1~2.6	0.03~0.07	0.07~1.6	2.1~4.4

类的 MAE,视频类的精度相比未分类模型的精度提高了 2.8%~6.7%。

表 4 分类和未分类模型的 MAE 对比

测试集	分类	未分类	测量	精度
	MAE/J	MAE/J	范围/J	提升/%
小说	17.88	31.27	250~460	2.9~5.3
音乐	14.10	16.70	280~530	0.04~0.09
视频	31.05	47.76	250~580	2.8~6.7

上述结论表明,分类模型的能耗漏洞检测能力优于文献[10]所提的未分类模型,且预测精度最多提高 6.7%。

5 检测能耗漏洞

从豌豆荚分别收集 AcFun 视频,酷狗音乐、宜搜小说 6、7、7 个历史版本,其中每个版本的 APP 都执行相同的 UI 操作。收集 AcFun 视频、酷狗音乐、宜搜小说运行期间的系统调用与运行时长,输入进视频类、音乐类、小说类的模型中进行预测,观察预测的能耗趋势图,若出现较大的波动,则表明该版本的 APP 出现能耗漏洞。图 3~5 为预测 APP 运行期间的能耗趋势图,图中每 10 个区间为 1 个 APP 历史版本的 10 次预测结果,可以看出每个模型对 APP 的预测趋势图都与实际趋势图相符。

图 3 与 4 中 APP 历史版本的能耗波动较为平稳,表明收集的 APP 历史版本中未出现能耗漏洞。结合表 5 数据与图 5,可以看到 10~40 相比于 0~10 区间的能耗较高,

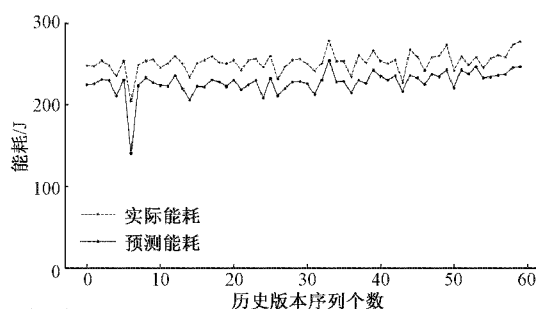


图 3 Acfun 历史版本预测

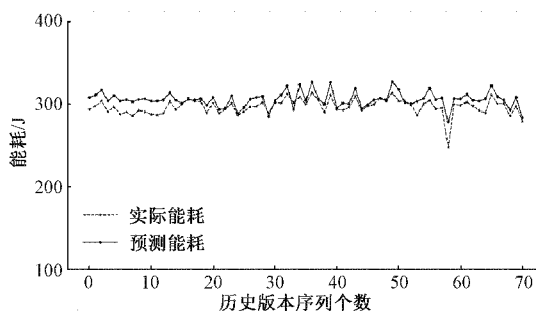


图 4 宜搜小说历史版本预测

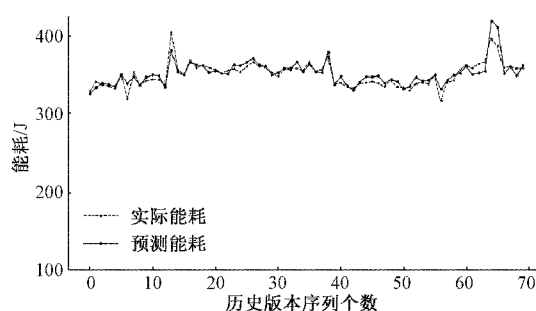


图 5 酷狗音乐历史版本预测

60~70 相比于 50~60 区间整体能耗上升,这表明该 4 个版本可能存在能耗漏洞。开发人员需要对这些版本检查并进行优化,降低 APP 的能耗。其中 Acfun 的 MAE 为 24.81,宜搜小说的 MAE 为 9.09,酷狗音乐的 MAE 为 5.32,均小于模型在测试集中的 MAE,表明本文提出的分

类模型在预测分类 APP 能耗时具有高度的精度。发人员可以使用该模型对 APP 的能耗进行预测,并跟不同历史版本的能耗进行对比来分析是否出现能耗漏洞。

表5 音乐类不同版本的平均能耗值

区间	1~10	11~20	21~30	31~40	41~50	51~60	61~70
能耗/J	338.6	356.8	360.7	360.0	342.7	342.8	367.9

6 结 论

本文对每个类别 APP 构建基于系统调用的能耗模型,将影响每个类别的重要系统调用作为特征,提高了特征细化粒度,并使用 MREFCV 进行特征的筛选。在测试集中的实验数据表明,MREFCV 构建的能耗模型精度优于 REFCV 构建的能耗模型,并且通过特征细化后的分类模型在测试集上的精度优于未分类模型。本文构建的分类模型的预测精度优于文献[10]提出的未分类模型,并且能准确检测出 APP 历史版本中的能耗漏洞,该模型具有一定的实用价值,帮助开发人员检测能量漏洞从而实现软件节能开发。

参考文献

- [1] 但强. 中小锂电池组切换式均衡方法研究[J]. 国外电子测量技术, 2020, 39(11): 82-87.
- [2] 刘大同, 宋宇晨, 武巍, 等. 锂离子电池组健康状态估计综述[J]. 仪器仪表学报, 2020, 41(11): 1-18.
- [3] DAMASEVICIUS R, TOLDINAS J, GRIGARAVICIUS G. Modelling battery behaviour using chipset energy benchmarking [J]. Elektronika ir Elektrotechnika, 2013, 19(6): 117-120.
- [4] 程茹洁, 陆建, 蒋厚明, 等. 智能移动终端的能耗模型研究[J]. 计算机技术与发展, 2017, 27(12): 128-132.
- [5] 张贵玲, 朱正伟, 蒋威, 等. 基于 UPPAAL 的手机功耗流量建模方法研究[J]. 电子测量与仪器学报, 2020, 34(12): 144-150.
- [6] PATHAK A, HU Y C, ZHANG M, et al. Fine-grained power modeling for smartphones using system call tracing[C]. Proceedings of the Sixth Conference on Computer Systems, 2012: 153-168.
- [7] DING L, SHUAI H, HALFOND W, et al. Calculating source line level energy information for android applications [C]. Proceedings of the 2013 International Symposium on Software Testing and Analysis, 2013: 78-89.
- [8] SHUAI H, DING L, HALFOND W, et al. Estimating mobile application energy consumption using program analysis [C]. 2013 35th International Conference on Software Engineering (ICSE), 2013: 92-101.
- [9] AGGARWAL K, ZHANG C, CAMPBELL J C, et al. The power of system call traces: Predicting the software energy consumption impact of changes[C]. Proceedings of 24th Annual International Conference on Computer Science and Software Engineering, 2014: 219-233.
- [10] ZHU C, ZHU Z, XIE Y, et al. Evaluation of machine learning approaches for android energy bugs detection with revision commits[J]. IEEE Access, 2019, 7(85): 241-252.
- [11] NORDHAUSEN KLAUS. An introduction to statistical learning—with applications in R by garth james, daniela witten, trevor hastie & robert tibshirani[J]. International Statistical Review, 2014, 82(1): 156-157.
- [12] BISHOP C, SVENSÉN M, WILLIAMS C. GTM: The generative topographic mapping [J]. Neural Computation, 2014, 10(1): 215-234.
- [13] CEPERIC E, CEPERIC V, BARIC A. A strategy for short-term load forecasting by support vector regression machines[J]. IEEE Transactions on Power Systems, 2013, 28(4): 4356-4364.
- [14] ZHANG M, ZHOU Z. A review on multi-label learning algorithms [J]. IEEE Transactions on Knowledge & Data Engineering, 2014, 26(8): 1819-1837.
- [15] HINDSLEY P, LANDRY C E, GENTNER B. Addressing onsite sampling in recreation site choice models[J]. Journal of Environmental Economics & Management, 2011, 62(1): 95-110.
- [16] DU P, XIA J, ZHANG W, et al. Multiple classifier system for remote sensing image classification: A review[J]. Sensors, 2012, 12(4): 4764-4792.

作者简介

于兴磊, 硕士研究生, 主要研究方向为嵌入式系统、机器学习。

E-mail: y1351517124@163.com

朱正伟, 教授, 主要研究方向为智能检测技术及应用、嵌入式系统及应用。

E-mail: zhuzw@cczu.edu.cn

朱晨阳, 讲师, 主要研究方向为机器学习、集成算法。

E-mail: zcy@cczu.edu.cn

诸燕平, 副教授, 主要研究方向为无线传感器网络、数值优化、移动网络计算。

E-mail: zhuyanping@cczu.edu.cn