基于四种机器学习算法检测网络诈骗 App 的对比研究

史晓苏[1,2] 李 欣[1]
SHI Xiaosu LI Xin

摘 要 为探索网络诈骗 App 检测的有效方法，运用决策树算法、随机森林算法、逻辑回归算法和支持向量机算法 4 种机器学习算法对 App 进行涉诈识别检测，验证算法的准确率、召回率、F_1 值、KS 值、G-mean 值、AUC 值，并从案情报告书中提取网络诈骗 App 验证算法预测结果。从算法指标结果和预测情况来看，支持向量机算法优于其他 3 种算法，App 存活时间、使用次数、名称以及与案件的关联程度是检测涉诈 App 的重要特征维度，同时结合打击整治网络诈骗 App 业务工作难点针对性给出对策建议，为公安机关打击网络诈骗犯罪提供参考。

关键词 网络诈骗 App；机器学习算法；决策树算法；随机森林算法；逻辑回归算法；支持向量机算法

doi：10.3969/j.issn.1672-9528.2024.04.042

0 引言

国家工信部 2023 年 8 月公布的数据显示[1]，据全国 App 技术检测平台统计，截至 2023 年 7 月底，我国国内市场上监测到活跃的 App 数量达到 261 万款（包括安卓和苹果商店），除了这些正规应用商店上架的 App，市面上还有大量 App 是需要通过下载链接或者二维码扫码进行安装的，犯罪分子利用这类“非官方”App 实施网络诈骗行为。《2022 年电信网络诈骗态势分析报告》[2] 显示，电信网络诈骗案件发案中网络诈骗的占比达到 80%，其中通过 App 来实施诈骗的占到 60% 以上。不仅一半以上的网络诈骗案件会利用 App，而且这些诈骗 App 普遍存活周期较短，URL 链接、二维码和 APK 安装包更新迅速。近年来，网络诈骗 App 数量呈爆炸式增长，面对大量的网络诈骗 App，传统检测技术难以应对反网络诈骗工作高效率、低成本的需求，利用机器学习算法进行网络诈骗 App 检测的研究是主流趋势。目前，对网络诈骗 App 的检测主要集中在 APK 包特征分析上，与正常 APK 相比，网络诈骗 APK 存在一定的特点，但大多数研究仅局限于 APK 包本身，忽略了与 APK 相关联的用户使用情况和前科情况。本研究以网络诈骗 App 为研究对象，提取分析维度，并对各个维度的特征进行标准化、归一化处理，采用决策树算法、随机森林算法、逻辑回归算法和支持向量机算法进行涉诈 App 的识别检测，提出一种适用于公安机关打击网络诈骗犯罪的 App 分类检测方法。

1. 中国人民公安大学 北京 100091
2. 上海市公安局网络安全保卫总队 上海 200025
[基金项目] 公安部应用创新计划项目 (2021YY14)

1 研究对象与方法

1.1 研究对象

本研究的 App 数据集主要来源于反诈中心涉诈 App 样本以及应用商店中爬取的合规 App，试验数据中包含 3257 个正规 App（正样本）和 2465 个网络诈骗 App（负样本），从中随机抽样出正样本（962 个）和负样本（1085 个）按照以 7:3 的比例分为训练集和测试集，将剩余的正样本（2295 个）和负样本（1380 个）作为验证集，验证各个算法模型评价指标，并从案情报告书中提取出 215 个网络诈骗 App 用于预测，检验试验结果与模型应用场景的匹配程度。

1.2 研究方法

（1）决策树算法

选用决策回归树，用来预测 App 的可疑情况。决策树分为 ID3（通过信息增益选择特征）、C4.5（通过信息增益比选择特征）、CART（通过 Gini 指数选择特征）等算法。考虑到数据是连续数值型，且是回归树，适合选用基于信息熵的决策树算法[3]。

信息熵公式：

$$Entropy(t) = -\sum_{i=0}^{c-1} p(i \mid t)\log_2 p(i \mid t) \quad (1)$$

式中：t 代表给定的节点，i 代表标签的任意类，$p(i|t)$ 代表标签分类 i 在节点 t 上所占的比例。这里采用 sklearn 中基于信息熵的信息增益（information gain），即父节点的信息熵和子节点的信息熵之差[4]。

（2）随机森林算法

随机森林的本质是由多个决策树构成的有监督学习算

法，它是用“bagging”方法（bootstrap aggregating）训练的，并采用随机有放回的选择训练数据构造分类器，最后组合学习到的模型来增加整体的效果[5]。

（3）逻辑回归算法

逻辑回归算法主要是针对非线性的分类算法，且特点就是非线性，它将因变量的取值范围转变成 [-1,1]。它采用的函数（Sigmoid 函数）公式：

$$S(t) = \frac{1}{1 + e^{-t}} \tag{2}$$

本文选定的惩罚系数为 L2，分类方式为 OvR（one vs rest），由于训练样本小，采用不同的损失函数优化算子效果相差无几，故采用适合小样本集的坐标轴下降法迭代求损失函数的最优解。

（4）支持向量机算法

支持向量机的思想就是使用一条直线（二维）或超平面（多维）将数据分成两类，同时保证离超平面最近的点与超平面的间隔尽可能小。

支持向量：离超平面最近的几个训练样本，且满足：

$$w^T \cdot x_i + b \geq +1, y_i = +1 \tag{3}$$

$$w^T \cdot x_i + b \leq -1, y_i = -1 \tag{4}$$

间隔：分类不同的支持向量之间的距离$\gamma = \frac{2}{\|w\|}$。

在训练样本不是线性可分的情况下，引入核函数的概念，其作用是将样本维数映射到更高维的空间，使得样本在高维空间中线性可分。

2 数据预处理

2.1 APK 特征筛选

通过对网络诈骗 App 的 APK 包特征进行分析和提取，特征维度包括：包名中点的数量、包名长度、包名中小写字母的占比、点间平均长度、包名中大写字母的占比、包名中数字的占比、点间最大长度、包名中小写字母的数量、包名连续数字的最大长度、包名中大写字母的数量、包名中数字的数量。通过计算维度之间的皮尔逊系数，当线性相关程度高于 0.8 时，即可认为两维度之间高度线性相关，得到高度线性相关的维度。

计算发现，包名中数字的数量、包名中数字的占比、包名连续数字的最大长度隶属于 1 个维度组，包名中大写字母的数量、包名中大写字母的占比隶属于 1 个维度组，点间最大长度、点间平均长度隶属于 1 个维度组，其他各个维度均独立为 1 个维度组。后续选择检测分类维度时在满足显著性水平的前提下挑选各个维度组中与样本标签最相关的维度作为分类维度。

通过计算各维度值与类别标签的皮尔逊系数、卡方值以及维度间的相关性筛选出有效的分类维度，各维度值与类别标签的皮尔逊系数、卡方值计算结果如表 1 所示。

表 1 各维度值与类别标签的皮尔逊系数、卡方值计算结果

特征	皮尔逊系数	p 值（皮尔逊系数）	卡方值	p 值（卡方值查表得到）
包名中点的数量	-0.185 364	1.38E-17	13.199 543 824	0.000 280 017
包名长度	0.088 222	5.44E-05	24.144 744 477	8.935 97E-07
包名中小写字母的占比	-0.640 453	2.57E-241	39.343 813 302	3.553 77E-10
点间平均长度	0.266 973	2.17E-35	88.301 177 17	5.621 E-21
包名中大写字母的占比	0.423 58	1.22E-91	119.212 581 767	9.408 63E-28
包名中数字的占比	0.443 28	3.50E-101	120.240 411 402	5.604 04E-28
点间最大长度	0.257 082	7.44E-33	190.022 238 167	3.143 96E-43
包名中小写字母的数量	-0.400 781	2.30E-81	609.003 322 367	1.843 01E-134
包名连续数字的最大长度	0.424 419	4.92E-92	1 477.730 451 35	0
包名中大写字母的数量	0.410 881	8.09E-86	2 408.632 683 78	0
包名中数字的数量	0.421 576	1.05E-90	2 562.334 961 23	0

对表 1 中的计算结果，先考虑卡方值及其 p 值。当 p 值小于 0.05 时，拒绝原假设 H0：特征与 label 相互独立，该独立变量与输出结果有关系，该独立变量重要，由表 1 卡方值可知，11 个维度特征均与标签不独立，具有相关性。再由各特征之间的相关性决定各特征的取舍，从高度相关的同一组维度中取皮尔逊系数最高的特征作为分类特征，即点间平均长度、包名中大写字母的占比、包名中数字的占比作为各组维度中筛选出来的特征，最终选取结果如表 2 所示。

表 2 分类特征选择结果

特征	是否分类特征
包名中点的数量	是
包名长度	是
包名中小写字母的占比	是
点间平均长度	是
包名中大写字母的占比	是
包名中数字的占比	是
点间最大长度	否
包名中小写字母的数量	是
包名连续数字的最大长度	否
包名中大写字母的数量	否
包名中数字的数量	否

鉴于网络诈骗 App 的存活时间一般较为短暂，App 的使用次数与正规 App 存在明显差距，同时结合诈骗案件中

App 的用户使用情况特点以及 APK 特征筛选结果，选定的分析维度特征如下：（1）App 的用户使用量；（2）App 的存活时间；（3）用户使用 App 的最大时长；（4）App 总使用次数；（5）App 应用名称命中敏感词（如贷、借等）的个数；（6）App 包名命中敏感词（如：plus.、w2a.W2A、.dcloud. 等）的种类数；（7）App 包名命中敏感词详情：包名以 plus. 开头，包名包含 w2a.W2A，包名包含 .dcloud.，包名包含 .bcloud.，包名以 .xyz 结尾，包名以 .top 结尾，包名以 .apk 结尾；（8）App 包名的组成规则：包名长度、包名中点的数量、包名中小写字母的数量、包名中大写字母的占比、包名中小写字母的占比、点间平均长度、包名中数字的占比；（9）App 的日使用次数，指近 3 个月内 App 所有用户日使用次数均值与用户日使用次数标准差两个特征，其中用户日使用次数 = 用户 3 个月内总使用次数 / 用户 3 个月内使用天数；（10）App 应用名是否案件应用名称，App 应用名是否案件应用名称指 App 应用名精确匹配案件 App 应用名，结果为是（1）或否（0）；（11）App 应用名得分，指 App 应用名精确匹配案件 App 应用名，若相等则取对应的案件 App 应用名权重分值，若不等则直接取 0。

2.2 特征预处理

（1）基于决策树的最优分箱与 IV 值计算

在检测网络诈骗 App 的建模问题中，需要对变量的预测能力做一个快速、初步的评估。针对二分类问题，本文用 IV 值（information value）来衡量特征变量的预测能力，然后再筛选出 IV 值高于某个阈值的一篮子特征来进行下一步的建模工作，一般变量 IV 值大于 0.5 即认为变量具有良好的预测能力。为了计算某个变量的 IV，首先需要对其进行分箱。为了尽可能使得 IV 值计算最大，同时尽可能保证分箱的单调性，利用决策树的信息增益最大化思想来实现变量的最优分箱。由于 App 用户量、App 总使用次数、App 存活时间、用户使用 App 的最大时长 4 个维度的数据量级差异较大，最高可达百万级别，最低为 0，故需要将 4 个维度特征分箱离散化后再输入分类器进行分类。

用户量：决策树得到的最优分箱边界为 [0,4.5,13.5,78.5,338.0,2544.5,100 000 1.1]，该变量 IV 值 =7.19，后期根据最优分箱将变量值装箱到 [1,2,3,4,5,6]，对应关系如表 3 所示。App 总使用次数：决策树得到的最优分箱边界为 [0,6.5,22.5,179.0,1 002.0,4 092.0, 395 582 1.1]，该变量 IV 值 =7.2，后期根据最优分箱将变量值装箱到 [1,2,3,4,5,6]，对应关系如表 4 所示。App 存活时间：由决策树算法寻得的最优分箱边界为 [0.0,115.807,166.095,206.884,212.706,272.475]，该变量 IV 值 = 9.74。由于 App 库中积累不足 1 年，故在最后的分箱边界中加入 365.1 作为上限值，根据最优分箱将变量值装箱到 [1,2,3,4,5,6]，对应关系如表 5 所示。

此外，直接将 App 存活时间按天数分为新 App（存活 1 个月内，小于 30 天）、存在 1 ～ 3 个月（大于等于 30 天，小于 90 天）、存在 3 ～ 6 个月（大于等于 90 天，小于 180 天）、存在半年以上（大于等于 180 天），分别映射为 [1,2,3,4]。用户使用 App 的最大时长：由决策树算法寻得的最优分箱边界为 [0.0, 11.053, 50.066, 74.057, 84.815,88.235, 90.099]，该变量 IV 值 = 6.76。由于该维度为近 90 天内的用户使用 App 的最大时长，故上限值为 90。根据最优分箱将变量值装箱到 [1,2,3,4,5,6]，对应关系如表 6 所示。

表 3 用户量最优分箱

bins	good	bad	total	good_pct	bad_pct	total_pct	bad_rate	woe	iv	映射离散值
[0.0, 4.5)	20	743	763	0.020790021	0.684792627	0.372740596	0.97378768	-3.4[illegible]4642956	2.32045203	1
[4.5, 13.5)	19	178	197	0.01975052	0.1640553	0.096238398	0.903553299	-2.1[illegible]7023756	0.305496647	2
[13.5, 78.5)	62	139	201	0.064449064	0.128110599	0.098192477	0.691542289	-0.6[illegible]7018733	0.043736667	3
[78.5, 338.0)	83	20	103	0.086278586	0.01843318	0.050317538	0.194174757	1.5[illegible]2915	0.104714578	4
[338.0, 2544.5)	160	3	163	0.166320166	0.002764977	0.079628725	0.018404908	4.0[illegible]2342	0.670066367	5
[2544.5, 1000001.1)	618	2	620	0.642411642	0.001843318	0.302882267	0.003225806	5.8[illegible]2092	3.749670518	6

表 4 APP 总使用次数最优分箱

bins	good	bad	total	good_pct	bad_pct	total_pct	bad_rate	woe	iv	映射离散值
[0.0, 6.5)	18	746	764	0.018711019	0.687557604	0.373229116	0.976439791	-3.6[illegible]4033027	2.410545182	1
[6.5, 22.5)	23	193	216	0.023908524	0.177880184	0.105520274	0.893518519	-2.0[illegible]6875158	0.3090019	2
[22.5, 179.0)	83	133	216	0.086278586	0.122580645	0.105520274	0.615740741	-0.3[illegible]1187705	0.012748837	3
[179.0, 1002.0)	95	8	103	0.098752599	0.007373272	0.050317538	0.077669903	2.5[illegible]6165	0.237107072	4
[1002.0, 4092.0)	128	3	131	0.133056133	0.002764977	0.063996092	0.022900763	3.8[illegible]8791	0.504713905	5
[4092.0, 3955821.1)	615	2	617	0.639293139	0.001843318	0.301416707	0.003241491	5.8[illegible]5903	3.728313903	6

表 5 APP 存活时间最优分箱

bins	good	bad	total	good_pct	bad_pct	total_pct	bad_rate	woe	iv	映射离散值
[0.0, 115.807)	4	875	879	0.004158004	0.806451613	0.429408891	0.995449374	-5.2[illegible]60871	4.226168801	1
[115.807, 166.095)	4	99	103	0.004158004	0.09124424	0.050317538	0.961165049	-3.0[illegible]4674	0.268966245	2
[166.095, 206.884)	23	91	114	0.023908524	0.083870968	0.055691255	0.798245614	-1.2[illegible]4475	0.075255534	3
[206.884, 212.706)	86	17	103	0.089397089	0.015668203	0.050317538	0.165048544	1.7[illegible]768	0.128395521	4
[212.706, 272.475)	845	3	848	0.878378378	0.002764977	0.414264778	0.003537736	5.7[illegible]5154	5.044448343	5
[272.475, 365.100)										6

表 6 用户使用 APP 的最大时长最优分箱

bins	good	bad	total	good_pct	bad_pct	total_pct	bad_rate	woe	iv	映射离散值
[0.0, 11.053)	20	696	716	0.020790021	0.641474654	0.349780166	0.972067039	-3.4[illegible]9296571	2.128512	1
[11.053, 50.066)	24	249	273	0.024948025	0.229493088	0.133365901	0.912087912	-2.2[illegible]9078251	0.453901	2
[50.066, 74.057)	77	110	187	0.08004158	0.101382488	0.0913532	0.588235294	-0.2[illegible]6354129	0.005044	3
[74.057, 84.815)	111	24	135	0.115384615	0.022119816	0.065950171	0.177777778	1.65[illegible]97186	0.154055	4
[84.815, 88.235)	165	4	169	0.171517672	0.003686636	0.082559844	0.023668639	3.82[illegible]971928	0.644466	5
[88.235, 90.099)	565	2	567	0.587318087	0.001843318	0.276990718	0.003527337	5.76[illegible]999366	3.374676	6

表中指标释义如下：

$$\mathrm{woe}_i = \ln\left(\frac{\text{第}i\text{个分箱正样本数}}{\text{第}i\text{个分箱负样本数}}\right) - \ln\left(\frac{\text{正样本总数}}{\text{负样本总数}}\right) \tag{5}$$

$$\mathrm{IV}_i = \left(\frac{\text{第}i\text{个分箱负样本数}}{\text{负样本总数}} - \frac{\text{第}i\text{个分箱正样本数}}{\text{正样本总数}}\right) \times \left(\ln\left(\frac{\text{第}i\text{个分箱负样本数}}{\text{负样本总数}}\right) - \ln\left(\frac{\text{第}i\text{个分箱正样本数}}{\text{正样本总数}}\right)\right) \tag{6}$$

good_pct= 第 *i* 个分箱正样本数 / 正样本总数

bad_pct= 第 *i* 个分箱负样本数 / 负样本总数

total_pct= 第 *i* 个分箱样本总数 / 总样本数

bad_rate= 第 *i* 个分箱负样本数 / 第 i 个分箱样本总数

（2）One-Hot 编码

基于树的分类算法，无需对离散特征进行 One-Hot 编码，对于 SVM、逻辑回归分类等算法，将（1）中离散化后的特征进行独热编码。独热编码可以进行特征转换，增强模型的非线性能力，将离散特征的取值扩展到欧式空间，让特征之间的距离计算更加合理。同时，将离散特征编码后可视为多个连续特征，可以采用连续型特征的归一化方法进行归一化。

（3）数据标准化

将经历特征预处理步骤（1）或（1-2）后的特征进行数据标准化，将每列的特征值标准化为方差为 0、标准差为 1 的分布。

3 结果对比分析

3.1 各算法预测结果

经过上述 4 种算法的训练在验证集上进行预测，得到每个算法的评价指标如表 7 所示，从 F_1 值、KS 值、G-mean 值和 AUC 值综合判断，模型评价指标中支持向量机、逻辑回归算法相对最优。

表 7 模型评价指标

分类算法	类别	精确率	召回率	F_1 值	G-mean	KS	AUC
决策树	正规 App	0.96	0.62	0.76	0.762 737	0.582 078	0.791 039
	网络诈骗 App	0.6	0.96	0.74			
随机森林	正规 App	0.98	0.57	0.72	0.753 003	0.551 241	0.775 62
	网络诈骗 App	0.58	0.98	0.73			
逻辑回归	正规 App	0.98	0.66	0.79	0.785 63	0.632 883	0.816 442
	网络诈骗 App	0.63	0.97	0.77			
支持向量机	正规 App	0.99	0.64	0.78	0.788 246	0.635 749	0.817 874
	网络诈骗 App	0.63	0.99	0.77			

在对案件报告书中提取的 215 个网络诈骗 App 预测时，得到准确率如表 8 所示，决策树和支持向量机的预测准确率相对较高。

表 8 模型预测准确率

算法	预测为正规 App	预测为网络诈骗 App	预测正确样本占比 /%
决策树	7	208	96.744 186 05
随机森林	45	170	79.069 767 44
逻辑回归	52	163	75.813 953 49
支持向量机	30	185	86.046 511 63

综合验证集上各指标以及预测准确率比较，支持向量机表现较为稳定。

3.2 特征重要性评价

试验得到算法特征重要性如表 9 所示，其中特征重要性为空表示该模型未使用该特征作为分类特征，x0-x32 为用户量 _ 决策树、App 存活时间 _ 决策树、App 存活时间 _ 分组、用户最长使用 App 时间 _ 决策树、App 总使用次数 _ 决策树等特征经独热编码后的特征。从表 11 可知，App 存活时间、总使用次数、日使用次数、App 名称以及与案件的关联程度均是较为重要的特征维度，有助于提高网络诈骗 App 的检测准确率。

表 9 模型结果算法特征重要性

指标维度	决策树特征重要性	随机森林特征重要性	逻辑回归特征重要性
包名长度	0.000 000 00e+00	0.002 462 43	0.470 998 29
包名中点的数量	1.605 596 28e-03	0.000 596 94	-0.346 285 23
包名中小写字母的数量	1.644 462 54e-04	0.004 088 31	-0.304 096 46
包名中大写字母的占比	0.000 000 00e+00	0.001 392 9	-0.107 982 24
包名中小写字母的占比	1.705 236 42e-02	0.015 693 61	-0.783 644 08
点间平均长度	6.482 519 01e-04	0.003 411 02	-0.003 840 66
包名中数字的占比	0.000 000 00e+00	0.008 206 62	1.001 818 9
用户量 _ 决策树	1.090 227 67e-02	0.040 197 82	
App 存活时间 _ 决策树	9.311 464 18e-01	0.249 746 2	
App 存活时间 _ 分组	1.984 949 95e-02	0.137 779 78	
用户最长使用 App 时间 _ 决策树	0.000 000 00e+00	0.185 972 74	
App 总使用次数 _ 决策树	4.148 188 85e-04	0.126 491 55	
日使用次数 _ 均值	1.462 574 79e-03	0.038 109 24	-0.110 770 94
日使用次数 _ 标准差	3.837 884 76e-03	0.102 186 18	-0.545 135 52
应用名称命中种类数	1.139 354 54e-03	0.007 535 84	0.344 233 48
包名命中种类数	1.732 572 08e-03	0.005 854 64	0.345 788 85
是否案件应用名称	9.420 184 60e-03	0.022 132 96	1.242 876 72
App 名称得分	6.237 579 11e-04	0.047 257 67	0.082 348 84

App包名规则	plus. 开头	0.000 000 00e+00	0.000 260 14	-0.269 264 64
	w2a.w2a 开头	0.000 000 00e+00	0.000 000 00e+00	0.033 565 33
	包含 .bcloud.	0.000 000 00e+00	0.000 623 4	0.619 511 19
	包含 .dcloud.	0.000 000 00e+00	0.000 000 00e+00	-0.059 638 4
	.xyz 结尾	0.000 000 00e+00	0.000 000 00e+00	0.258 292 92
	.top 结尾	0.000 000 00e+00	0.000 000 00e+00	0.173 458 05
	.apk 结尾	0.000 000 00e+00	0.000 000 00e+00	-0.036 760 39
x0				0
x1				0.102 210 4
x2				0.080 354 5
x3				0.055 453 62
x4				0.394 026 87
x5				0.027 672 4
x6				-0.416 550 06
x7				0
x8				1.086 368 94
x9				-0.122 966 77
x10				-0.060 245 29
x11				0.039 081 55
x12				-1.054 984 92
x13				-0.036 584 14
x14				0
x15				0.449 909 26
x16				0.282 096 66
x17				0.321 188 36
x18				-0.803 769 39
x19				0
x20				-0.068 874 04
x21				0.097 086 42
x22				0.212 092 51
x23				-0.040 740 55
x24				-0.002 956 15
x25				-0.116 933 94
x26				0
x27				0.633 845 59
x28				0.069 614 15
x29				-0.211 000 02
x30				-0.627 681 2
x31				0.139 953 17
x32				-0.368 625 71

4 对策建议

为有效整治 APP 乱象，围绕网络诈骗 APP 的特点可从以下三方面着手：

（1）加强应用商店审查制度，完善用户评价体系。应用商店作为 APP 的推广平台，必须严格审查 APP 内容，严格把控 APP 上架原则，强化机器审核和人工复核两道关卡，同时，应用商店应根据用户举报信息对疑似网络诈骗 APP 立即予以下架，形成应用商店和用户对 APP 的双向监督作用。

（2）斩断非法 APP 制作、封装的上下链条，虽然第三方应用开发平台和封装平台并不会直接实施网络诈骗行为，但其行为实际上却为网络诈骗 APP 提供了技术帮助，公安机关应严肃规范管理第三方 APP 的开发、封装平台，强化落实 APP 的注册备案制度，加强 APP 监管力度，对违法犯罪分子形成一定的威慑作用 [6]。

（3）完善网络诈骗 APP 对比分析库，网络诈骗 APP 的特征以及源码存在一定的雷同性，可以利用这一特点，在涉 APP 的网络诈骗案件侦办过程中，积累 APP 包文件、下载地址、邀请码、文件哈希值等相关信息，完善网络诈骗 APP 库的建立，便于公安机关快速掌握更多的涉案样本。

5 结语

本文研究了利用机器学习算法进行网络诈骗 App 检测的方法，通过对比决策树算法、随机森林算法、逻辑回归算法和支持向量机算法 4 种机器学习算法的指标结果和预测准确性，综合得出支持向量机算法是最好的选择，并给出各算法的特征重要性评价结果，提供了打击整治网络诈骗 App 的对策建议，对网络犯罪 App 分类研究具有参考价值和借鉴意义，可推广应用至网络色情、网络赌博、网络黑灰产等网络犯罪 App 的识别检测中。

参考文献：

[1] 工信部网站 .2023 年 1—7 月份互联网和相关服务业运行情况 [EB/OL]. (2023-08-31).https://wap.miit.gov.cn/gxsj/tjfx/hlw/art/2023/art_a150f7ab60be4acb8b39dd292b2d695b.html.

[2] 中国移动 .2022 年电信网络诈骗态势分析报告 [EB/OL].(2023-04-12).https://mp.weixin.qq.com/s?__biz=MzIzMDQyMTk5OA==&mid=2247496328&idx=2&sn=c05df93d6620dca22028298ead24bbe8&chksm=e8b11609dfc69f1f31e0e838a63dc577664ee8c51c991a69117f83e4f9cba5c2b02893719868&scene=27.

[3] 刘春梅 , 孙改平 . 决策树 ID3 分类算法的研究 [J]. 科技信息 , 2010(26): 498-500.

[4] 张曼雪 , 李欣 , 张勇斌 , 等 . 基于决策树的印刷包装企业智慧数据空间研究 [J]. 数字印刷 ,2022(3):19-26.

[5] 刘铖朴 . 基于集成学习投票系统的瓦斯涌出强度分类模型 [J]. 内蒙古煤炭经济 ,2019(7):111-112.

[6] 李玲玲 , 牛军生 , 蔡政 . 涉虚假 App 的电信网络诈骗犯罪侦查难点与对策研究 [J]. 中国人民公安大学学报 (自然科学版), 2021, 27(2):56-61.

【作者简介】

史晓苏（1996—），女，安徽枞阳人，硕士研究生在读，研究方向：图像处理，大数据分析。

李欣（1977—），通讯作者（email：lixin@ppsuc.edu.cn），男，江西宁都人，博士，教授，研究方向：网络安全、视频网络、人工智能。

（收稿日期：2024-01-25）