用人工智能进行移动端有害程序识别

马勇

(奇安信集团 数据安全技术事业部 研发总监)

(驱动开发网创始人 znsoft)

(北京理工大学信息系统及安全对抗实验中心 博士生)

注意

- 本分享仅提供理念，不包括任何实际代码
- 将分享PPT和相关论文合集
- 分享的最新成果来源于本人所在实验室师弟
- 本人曾在反病毒公司工作，熟悉反病毒流程，略知一二
- 仅起交流目的，任何错误请指正

主要内容

人工智能及机器学习简介

相关机器学习算法

有害移动应用的机器学习识别

引入知识图谱进行效果增强的设想

总结

传统检测方法

基于特征的检测方法,适用于Native接口, Signatures of Native code.

基于APK Hash值的方法（缺点：小变成大变，认不出）

基于应用特征的检测方法 **

基于应用代码序列特征 hash的检测方法

特征检测方法

“是否包含录音功能” “包含广告 sdk 的数量”、 “包含的支付 sdk 数量”、 “是否包含后台运行特征”、 “是否包含删除短信功能”、 “是否包含开机启动功能”、 “apk 包名是否是风险包名”、 “是否包含卸载应用的功能”、 “是否包含安装新应用的功能”、 “是否包含删除联系人的功能”、 “是否包含获取 10t 权限的功能”、 “是否包含添加新联系人的功能”、 “是否包含删除浏览器书签功能”、 “是否包含添加浏览器书签功能”、 “通过反射方式调用函数的数量”、 “apk 文件中是否包含子 apk 文件”	“是否包含自动下载文件的功能”、 “是否包含自动发送短信的功能”、 “apk 文件监听的系统事件的数量”、 “是否包含动态加载 jar 包的功能”、 “是否包含关闭其他应用的功能”、 “调用短信发送相关函数的数量”、 “apk 文件是否监控网络改变事件” “apk 文件是否申请有设备管理权限”、 “是否包含动态加载 dex 文件的功能”、 “是否包含获取安装应用列表的功能”、 “apk 解压后的 asset 目录下的文件数量”、 “apk 文件签名 issue 是否包含风险字符串”、 “apk 文件是否有在其他应用中弹出窗口的权限”、 “是否包含使用编译器默认的自带图标的功能”、 “apk 解压特定目录下文件后缀名与文件实际格式不符的文件数量”

传统的就是经典的，值得记忆

暂时忘了传统的方法吧

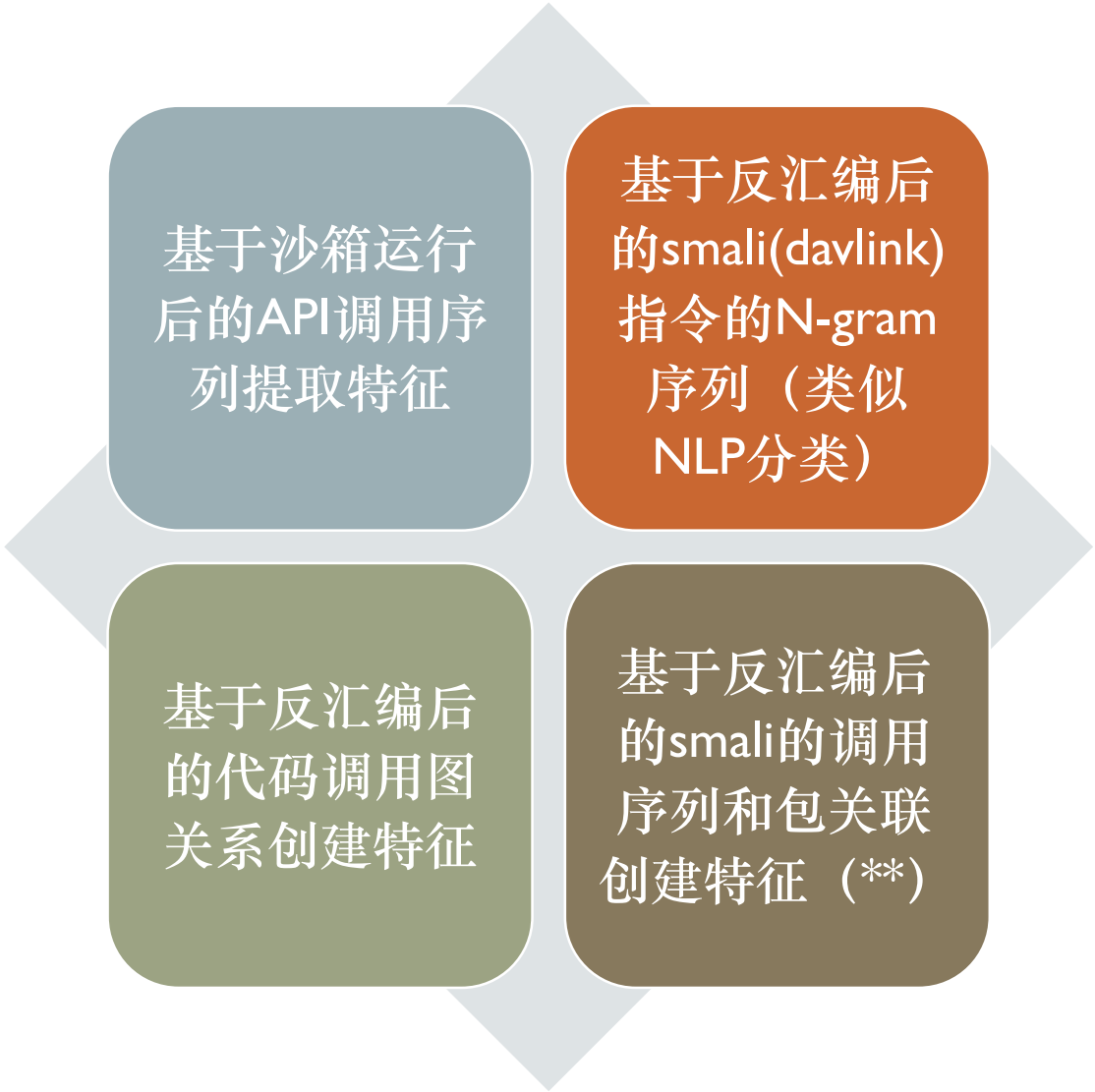
你们都是专家

往前才能进步

机器学习


人工智能
能够感知、推理、行动和适应的程序
机器学习
能够随着数据量的增加不断改进性能的算法
深度学习
机器学习的一个子集：利用多层神经网络从大量数据中进行学习

基于机器学习的检测方法

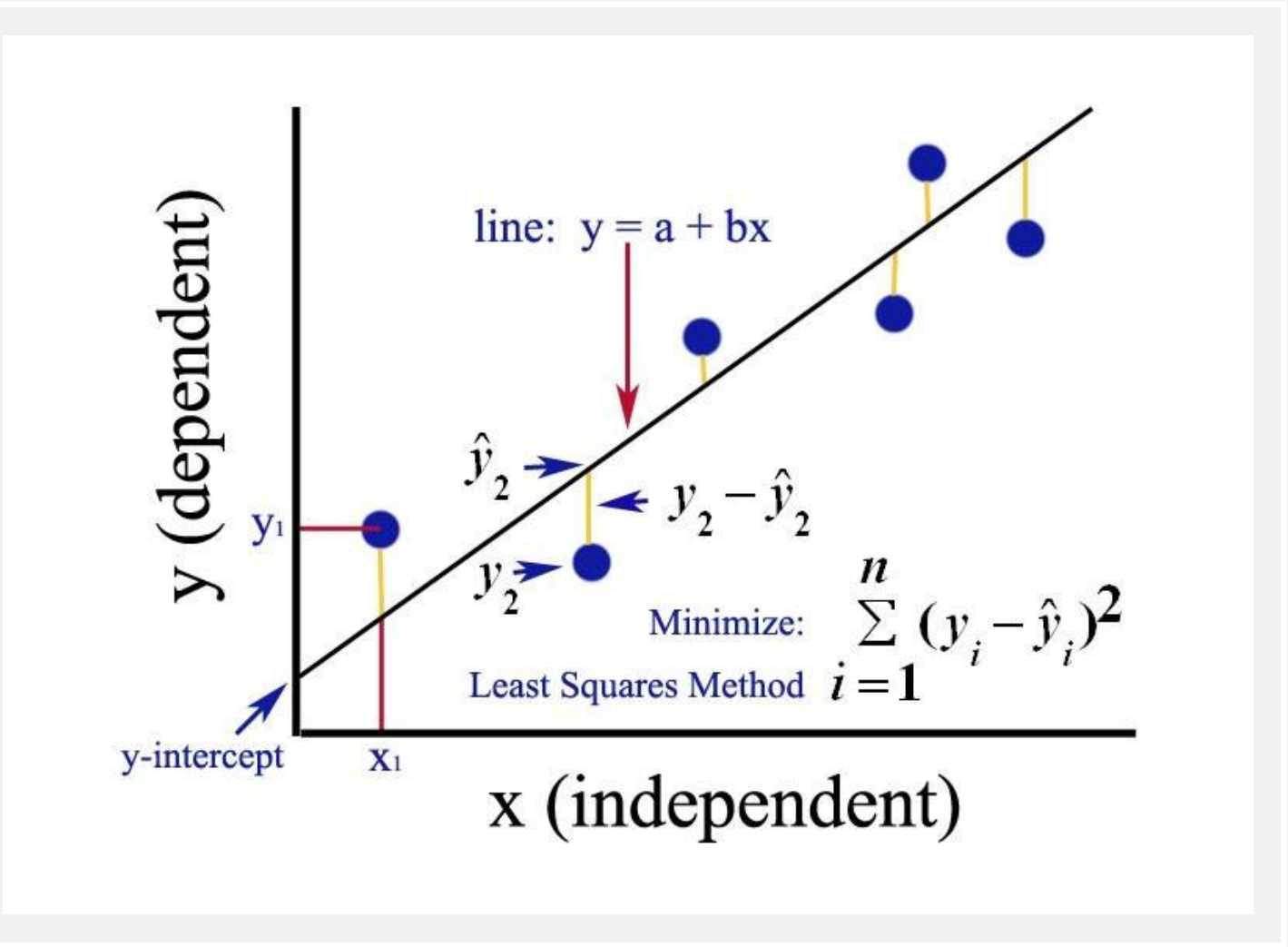


相关机器学习算法

- SVM 算法
- 朴素贝叶斯分类器
- 集成学习算法（XGBoost, 随机森林等）
- 基于深度学习的卷积模型 （分类器）
- 图神经网络(GNN)

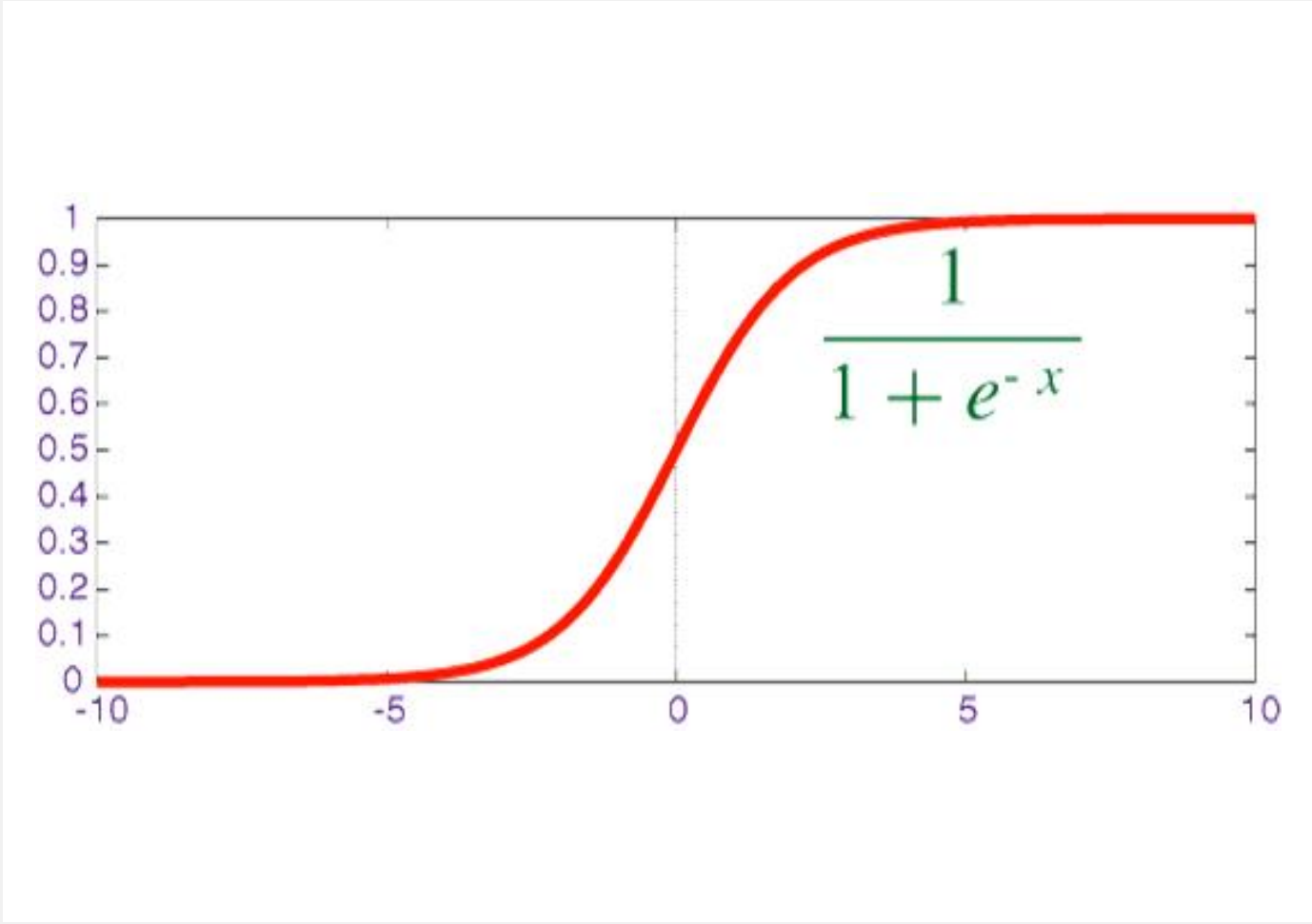
机器学习入门：简单分类模型

- 线性回归 Y=Wx+B



机器学习入门：简单分类模型

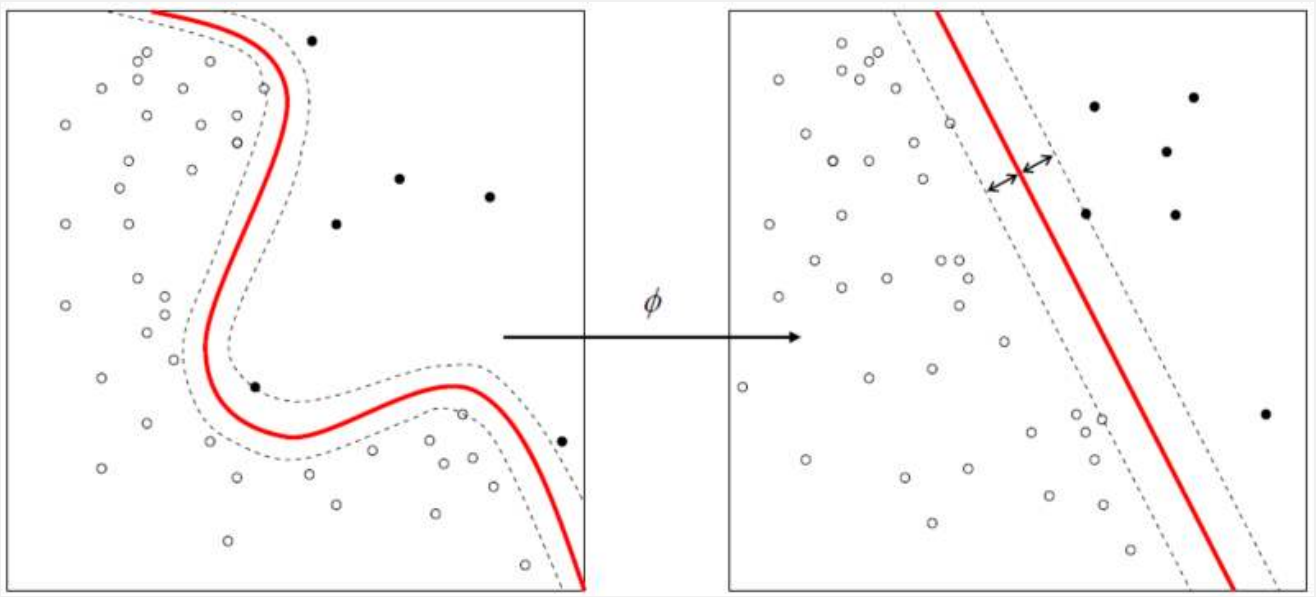
- 逻辑回归
  Y=sigmoid(Wx+B)



机器学习入门：支持向量机(SVM)

- 支持向量机是一种二分类模型，它的目的是寻找一个超平面来对样本进行分割，分割的原则是边界最大化，最终转化为一个凸二次规划问题来求解。

- SVM的关键是核函数，常用高斯核，包括线性核 ，高斯核 ，sigmoid核 (sigmoid核，支持向量机实现的就是一种多层感知器神经网络)



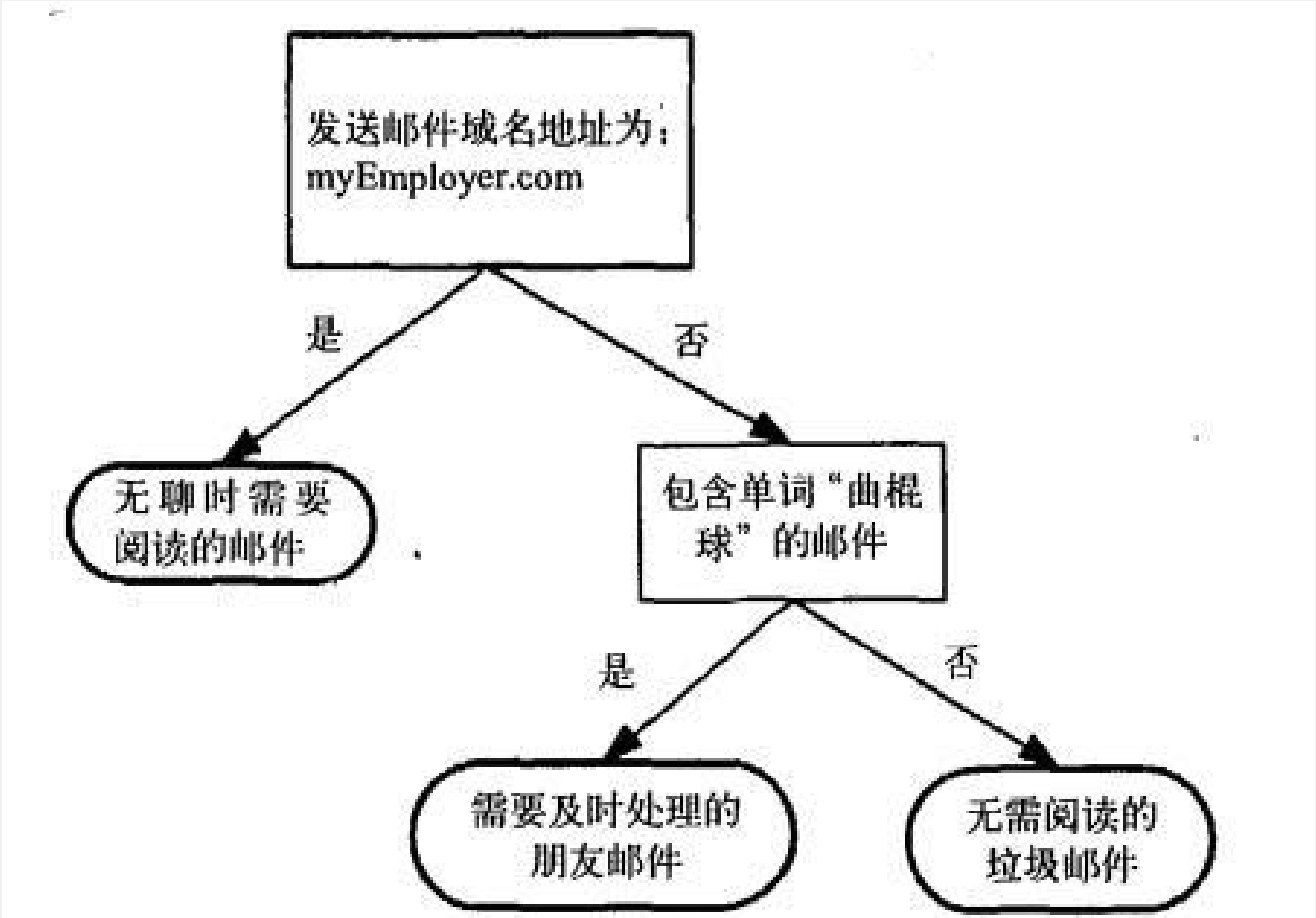
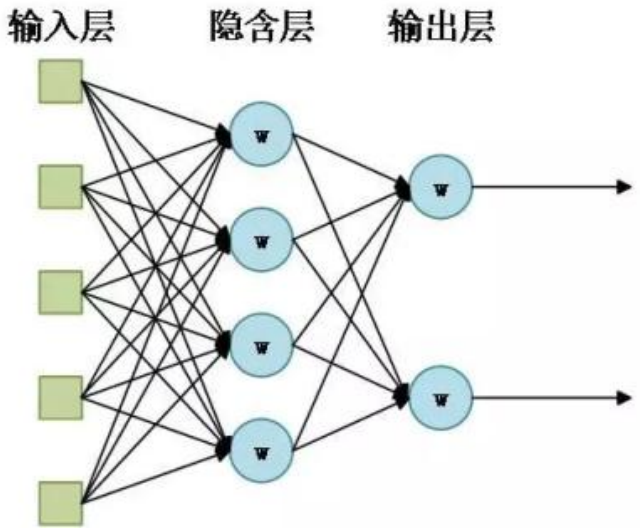


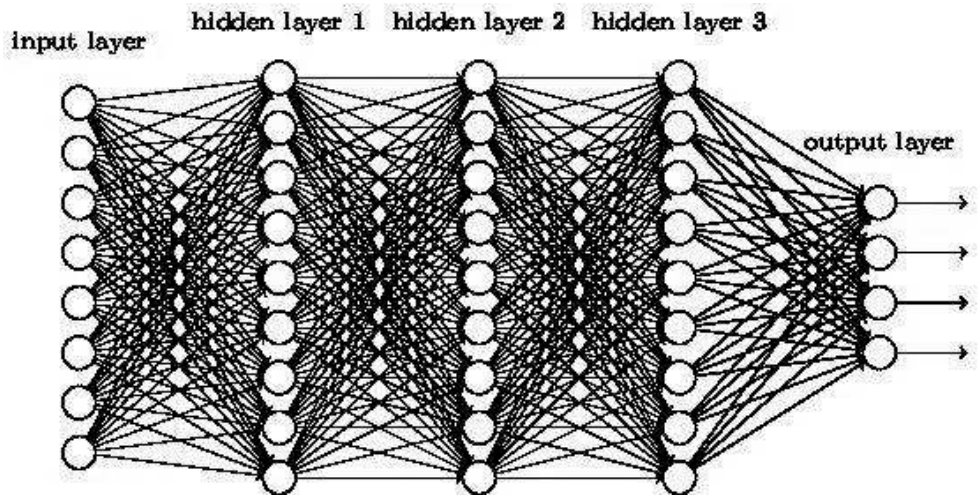
图3-1　流程图形式的决策树

决策树算法-分类用

- 在分类问题中，表示基于特征对实例进行分类的过程，可以认为是if-then的集合，也可以认为是定义在特征空间与类空间上的条件概率分布。
- **决策树通常有三个步骤**：特征选择、决策树的生成、决策树的修剪。

机器学习入门：神经网络


输入层
隐含层
输出层
w
w
w
w
w
w

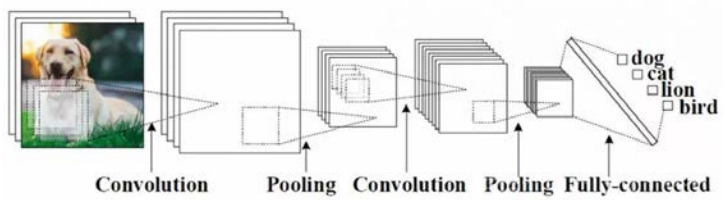

input layer
hidden layer 1
hidden layer 2
hidden layer 3
output layer

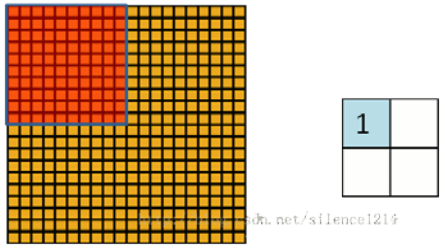
公式

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K}} \qquad for\ j = 1,\ \ ..., K$$

多分类的功臣：SOFTMAX

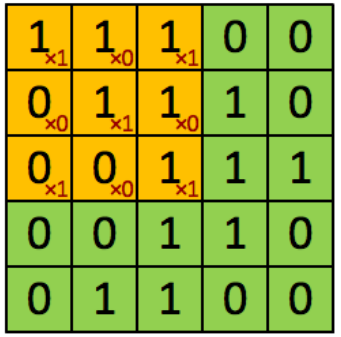
- 意义函数通常的意义：对向量进行归一化，凸显其中最大的值并抑制远低于最大值的其他分量。
- 通常用于CNN后的激活函数， CNN+FCN+SOFTMAX分类神器



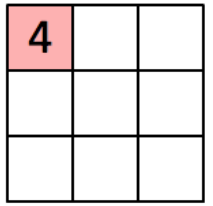


Convolved feature

Pooled feature



Image



Convolved Feature

卷积神经网络

机器学习算法的实质

特征表达来源

- Smali 语言
- 字节码

分类器

- 机器学习（SVM)
- 深度学习（CNN及GCN)

我们的方法

- **An efficient Android malware detection system based on method-level behavioral semantic analysis**
- **作者：HANQING ZHANG1, SENLIN LUO1,YIFEI ZHANG1, LIMIN PAN1**
- 北京理工大学信息系统及安全对抗实验中心

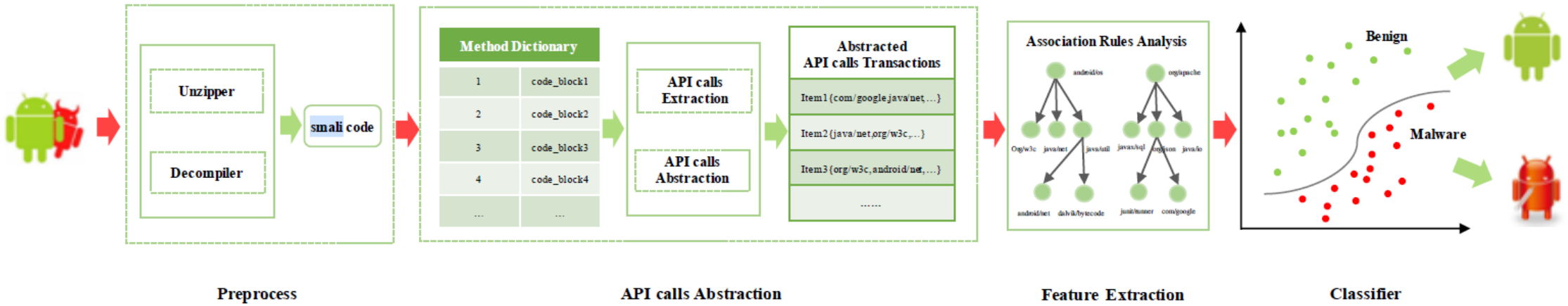
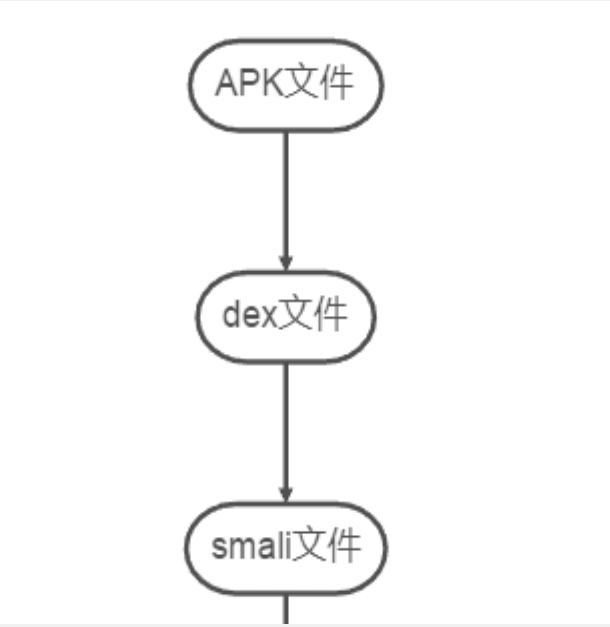


FIGURE 1. The framework of the detection system.

系统框架

各语言之间的转换

工具	作用
javac	java------>class
ddx	class------>dex
baksmali	dex------>smali
smali	smali------>dex
dex2jar	dex------>jar(class的压缩包)
apktool	apk------>smali



SMALI语言

- Davlink 执行的语言，类似汇编
- Java->class (byte code) ->dex
- Smali 语言类似PC机的汇编语言（dex里面就是davlink的机器码）
- 用apktool 反编译APK包得到 .smali文件

SMALI 示例

```
.method private fillPostData()V
    .locals 5
    .prologue
    const-string v2, "phone"
    invoke-virtual {p0, v2}, Lcom/satismangrooup/stlstart/Bragushterra;->getSystemService(Ljava/lang/String;)Ljava/lang/Object;
    move-result-object v1
    check-cast v1, Landroid/telephony/TelephonyManager;
    .local v1, "telephonyManager":Landroid/telephony/TelephonyManager;
    invoke-virtual {v1}, Landroid/telephony/TelephonyManager;->getDeviceId()Ljava/lang/String;
    move-result-object v2
    iput-object v2, p0, Lcom/satismangrooup/stlstart/Bragushterra;->imei:Ljava/lang/String;
    invoke-virtual {v1}, Landroid/telephony/TelephonyManager;->getNetworkOperator()Ljava/lang/String;
    move-result-object v2
    iput-object v2, p0, Lcom/satismangrooup/stlstart/Bragushterra;->operatorName:Ljava/lang/String;
    :try_start_0
    invoke-virtual {p0}, Lcom/satismangrooup/stlstart/Bragushterra;->getPackageManager()Landroid/content/pm/PackageManager;
    move-result-object v2
    invoke-virtual {p0}, Lcom/satismangrooup/stlstart/Bragushterra;->getPackageName()Ljava/lang/String;
    move-result-object v3
    const/4 v4, 0x0
    invoke-virtual {v2, v3, v4}, Landroid/content/pm/PackageManager;-
>getPackageInfo(Ljava/lang/String;I)Landroid/content/pm/PackageInfo;
    ……
.end method
```

预处理过程

反编译成 Smali 代码

将用户自定义函数逐个分开， 如 function foo() { ……}

对每个函数生成 调用事务（ a Call Transaction）， 如在 foo函数中调用了（fun1，fun2, fun3), 则生成列表 foo=[fun1,fun2,fun3], 只考虑出现关系，不考虑先后。

将整个程序转换成一个 call Transaction list.

调用事务

- 将APK反汇编成smali 语言
- 按API类型分为 75类， 如java.io, android.net
- 将每个函数的调用序列写成 分类列表，
- 如 T1=[java.io, android.net] , 它的事务宽度是 2 (2 of 75)
- 支持数， 某个事务Tk中的有效组合数，比如T1 中有5个抽象调用，支持数是指这5个调用可以组成的子事务数

关联规则分析

- 记得每个函数的调用事务不？ 里面是一系列抽象API调用
- 我们把它们两两排列（ $A_{75}^{2} = 5550$ 种关联规则)

- 对每个函数中的API调用，形成两两组合
- 对所有的这些两两组合 计算置信度

置信度公式

- Confidence(X->Y)=(X&Y)/X 表示，在X出现的时候，x和Y同时出现的比例，这是一个有向关系。

- 这个有向关系的值越大，表示他们之前的依存关系越大，越紧密 。

置信度计算例子

- 例如有三个方法 func1, func2, func3
- Func1=[api1,api2]， Func2=[api1,api3] ，Func3=[api1,api2]
- Api1->API2=2/3 (API1出现了三次，共现两次)
- API3->API1= 1/1=1
- Api1->api3=1/3

使用机器学习算法进行分类

- 记得那个 5 5 5 0 排列吗?
- 编号为0-5549，每个的置信度作为值。没有的置为0.
- 对一个apk 进行反汇编，计算出每种的置信度，作为训练数据送入机器学习分类算法处理。
- 对数据集分出训练集和测试集， 对训练集进行处理，训练出模型。再用此模型对测试集进行测试。

集成学习算法之BAGGING

- Bagging（套袋法）bagging的算法过程如下：
- 从原始样本集中使用Bootstraping方法随机抽取n个训练样本，共进行k轮抽取，得到k个训练集。（k个训练集之间相互独立，元素可以有重复）对于k个训练集，我们训练k个模型（这k个模型可以根据具体问题而定，比如决策树，knn等）对于分类问题：由投票表决产生分类结果；对于回归问题：由k个模型预测结果的均值作为最后预测结果。（所有模型的重要性相同）

随机森林分类器

随机森林是一种重要的基于Bagging的集成学习方法，可以用来做分类、回归等问题。

具有极高的准确率

随机性的引入，使得随机森林不容易过拟合

随机性的引入，使得随机森林有很好的抗噪声能力

能处理很高维度的数据，并且不用做特征选择

既能处理离散型数据，也能处理连续型数据，数据集无需规范化

训练速度快，可以得到变量重要性排序

容易实现并行化

效果:对SOTA 的MAMADROID等比较

不需要生成 call graphs, 速度快， 3秒与40秒的差异（在某一特定平台下）

由于是调用语义关系，可以对抗恶软的部分进化 ，更强的泛化能力。

准确率高： Drebin(benign 5.9K and malware5.6K) and AMD(benign 20.5K and malware 20.8K), our system has achieved 96% and 98% detection results both in Accuracy and F-measure

总结

存在问题

展望

总结之存在问题：只能辅助

- APK预处理非常费时，反编译费时间
- 生成Call Graph 要10秒级时间
- 不用Call Graph 时的本方法也是秒级，实时检测太慢
- DNN算法也慢，秒级

- 服务器端辅助人工进行自动分类，作为参考
- 手机端实时检测（暂时不要想了）

总结之展望

- Knowledge Graph+ Call Graph + Graph Neural Networks
- 本质 ,图+图卷积 分类
- RNN 生成可预测Feature Map (代码序列的少量变化)

致谢

- 本分享的后半部分成果来源于本人的师弟 张寒青 的研究成果：
- 《**An efficient Android malware detection system based on method-level behavioral semantic analysis**》

声明

我不是Android专家, 只是略懂一点机器及深度学习。

代表师弟给大家分享下

研究方向：基于知识图谱的语义相似性计算

End and not to be continued.

再见