

4/25-26

XKungfoo 2018 信息安全交流大会

当黑客之刃不再锋利
基于深度学习的Webshell检测

王泉



Who am I?

- 王泉
- 哈尔滨工程大学在读
- 杭州默安科技有限公司，影武者实验室，安全数据分析实习生
- 安全算法、安全大数据，机器学习在安全中的应用

人工智能应用广泛

- 自动驾驶
- 医疗
- 安防
- 教育
- ……
- 安全?



机器学习是如何解决安全问题的？

- 当你融资的时候，这是AI业务
- 当你招人的时候，你说你需要机器学习工程师
- 当你实现具体功能的时候，变成了线性回归
- 当你最终上线产品的时候，你用的还是规则和关键字匹配
-



Motivation

- 我们是不是真的需要机器学习？
- 机器学习如何在安全中落地？怎样收集样本？如何解决误报漏报？
- 使用机器学习是否会引入新的安全问题？机器学习本身是否安全可靠？



目录

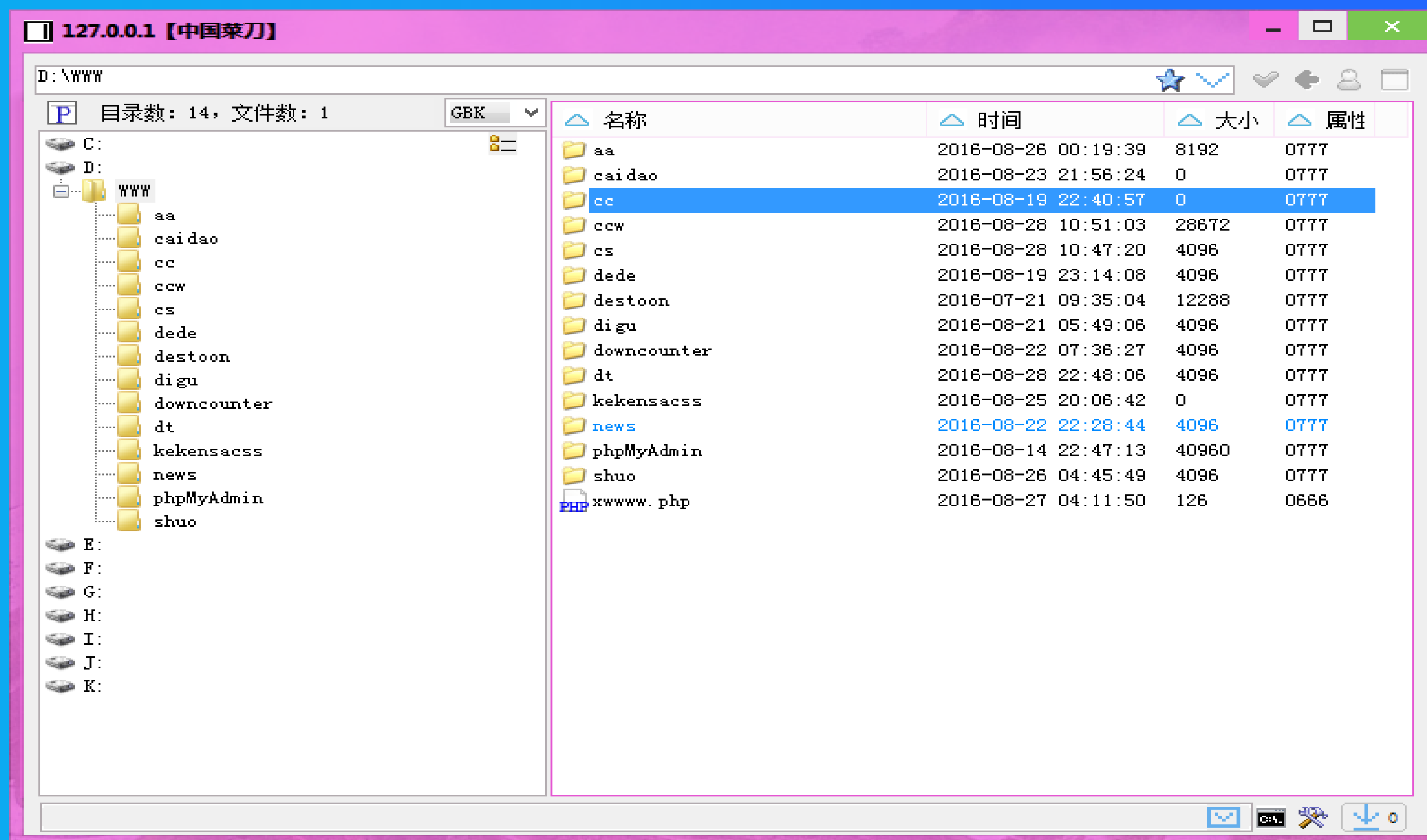
1. 菜刀与Webshell

2. 检测模型

3. 绕过机器学习检测

0x01 菜刀与Webshell

黑客之刃--菜刀



检测场景：检测文件还是检测流量？

- 检测文件：对用户上传文件或者网站目录下文件进行检测，检测的是Webshell文件本身。
- 检测流量：对所有HTTP请求进行检测，检测的是Webshell通信行为。

检测流量

捕捉攻击者连接Webshell，通过Webshell进行操作，并获取返回结果的行为。

优点：

- 实时检测，迅速阻断打击
- 直接发现后门路径、连接口令
- 检测准确，存在连接操作并有返回的都是失陷主机

一句话木马通信经过大量编码、混淆

```
Wireshark · 追踪 HTTP 流 (tcp.stream eq 35) · caidao-done

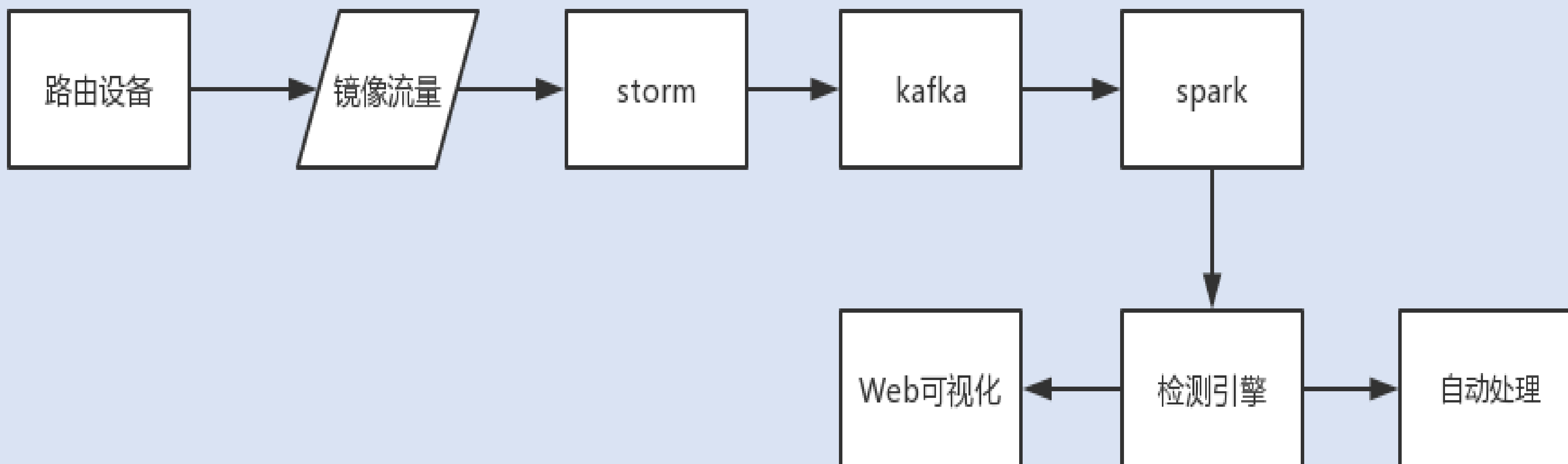
POST /-7.php HTTP/1.1
X-Forwarded-For: 180.76.63.42
Content-Type: application/x-www-form-urlencoded
Referer: http://192.168.199.248/
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Host: 192.168.199.248
Content-Length: 495
Connection: Close
Cache-Control: no-cache

-7=%40eval%01%28base64_decode%28%24_POST%5Bz0%5D
%29%29%3B&z0=QGluaV9zZXQoImRpc3BsYXlfZXJyb3JzIiwIMCIP00BzZXRfdGltZV9saW1pdCgwKTtAc2
V0X21hZ2ljX3F1b3Rlc19ydW50aW1lKDAp02VjaG8oIi0%2BfCIP0zskRj1nZXRfbWFnaWNfcXVvdGVzX2d
wYygpP3N0cmIwc2xhc2hlcygkX1BPU1RbInoxIl0p0iRfUE9TVFsiejEiXTskZnA9QGZvcGVuKCRGLCJyIi
k7aWYoQGZnZXRjKCRmcCkpe0BmY2xvc2UoJGZwKTtAcmVhZGZpbGUoJEYp031lbHNle2VjaG8oIkVSUk9S0
i8vIENhbiB0b3QgUmVhZCIp0307ZWNoYgIfDwtIik7ZGllKCK7&z1=%2Fhome%2Fwwwroot%2Fdefault
%2Fsql-labs%2FLess-1%2Findex.phpHTTP/1.1 200 OK
Server: nginx
```

0x02 检测模型

检测模型

检测系统架构：基于SPARK流式计算



为什么使用机器学习：

优点：

- 自动化，减少对规则的人力维护成本
- 智能化，发现未知威胁，关联分析

缺点：

- 实施难度大
- 性能影响大

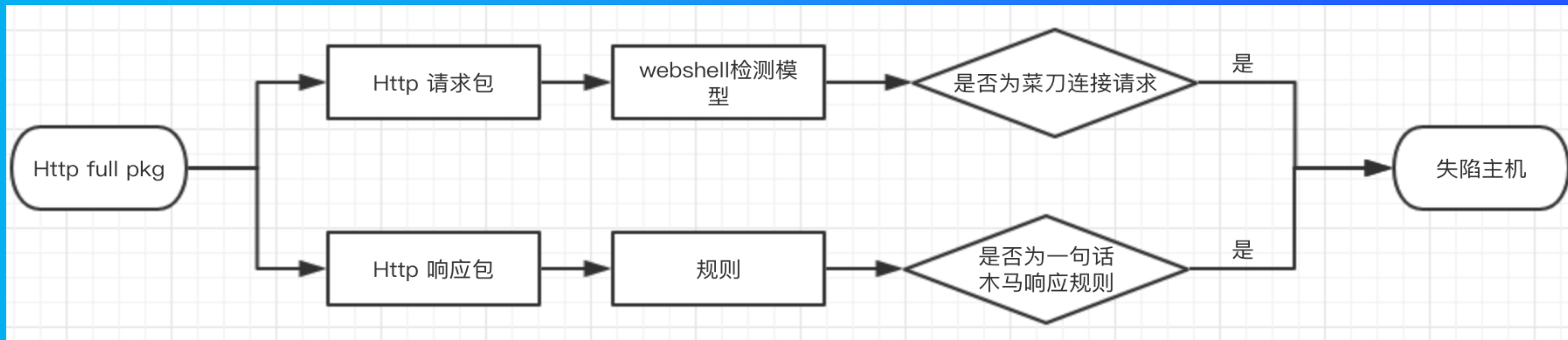
检测模型

Webshell检测流程

同时对请求包和响应包进行检测，同时使用机器学习和规则。

为什么需要响应包：

- Webshell扫描爆破
- 连接不存在的后门



检测模型

机器学习实施过程中的难点：

- 缺乏大规模高质量的训练样本
- 如何处理误报，模型怎样优化

检测模型

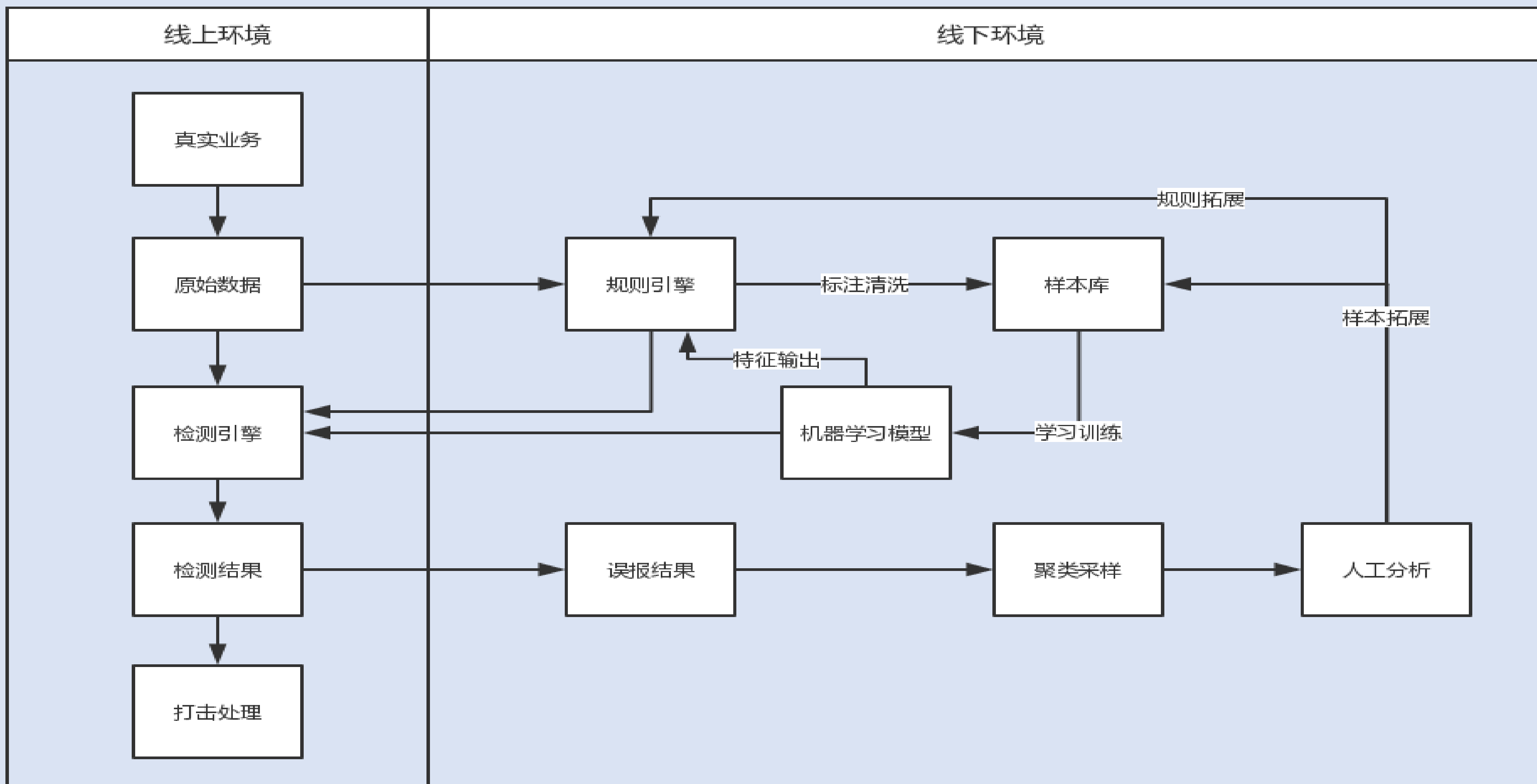
机器学习实施：

- 数据积累
- 算法方案
- 效果评估

算法模型只占很小一部分！

规则用于数据积累，模型用于预测。

检测模型



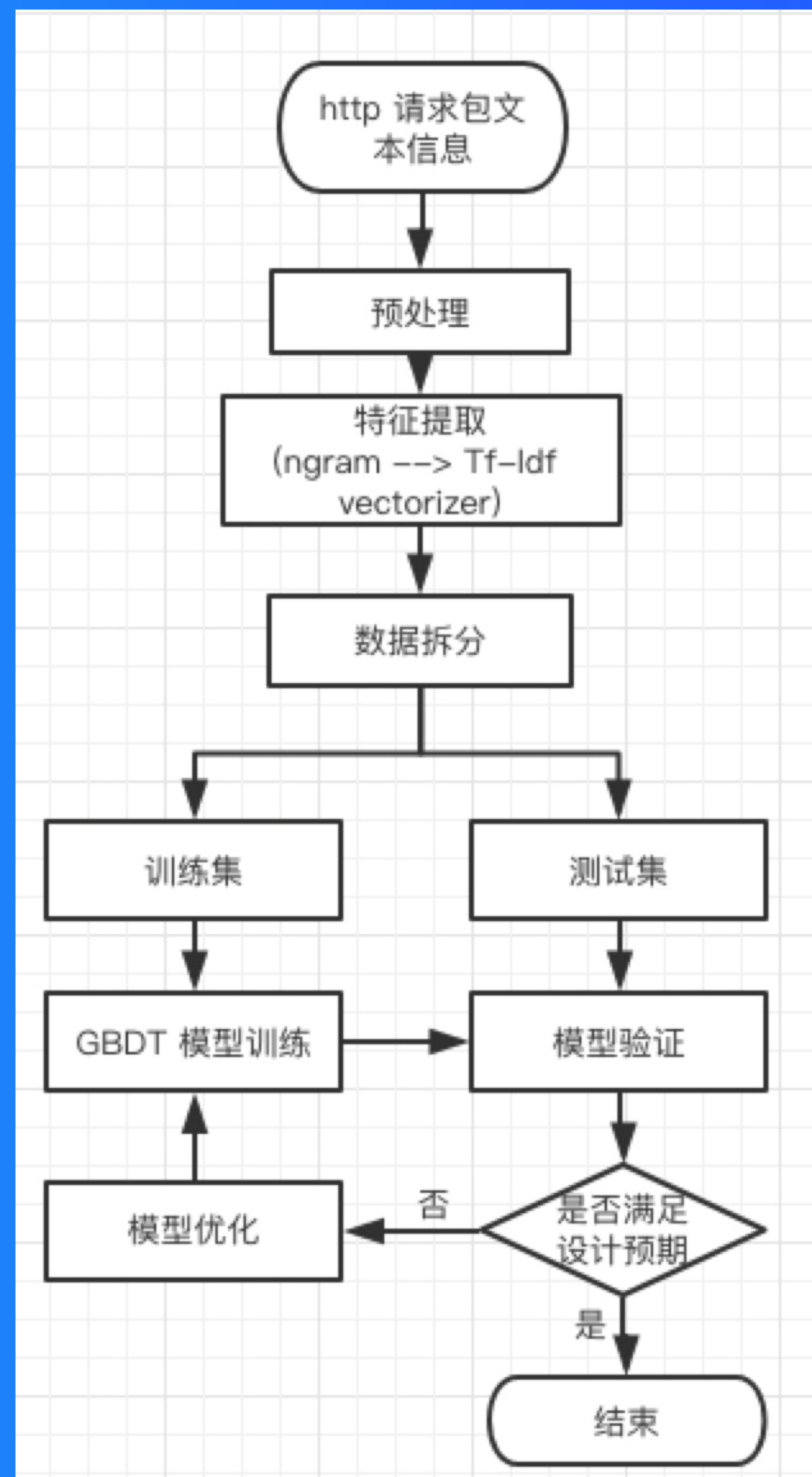
规则与模型

- 规则与模型不是对立的。
- 规则用于打标，模型用于检测。
- 模型使规则更强，规则使数据更干净。

检测模型

Webshell检测统计学习模型

1. 原始请求预处理
2. 文本特征抽取
3. 训练分类器
4. 验证评估



检测模型

原始请求预处理：减小计算量，去除对模型预测造成影响的字符。

- URL Decode
- 解码
- 替换汉字
- 替换乱码
- 去除特殊字符
-



```
Wireshark · 追踪 HTTP 流 (tcp.stream eq 35) · caidao-done

POST /-7.php HTTP/1.1
X-Forwarded-For: 180.76.63.42
Content-Type: application/x-www-form-urlencoded
Referer: http://192.168.199.248/
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Host: 192.168.199.248
Content-Length: 495
Connection: Close
Cache-Control: no-cache

-7=%40eval%01%28base64_decode%28%24_POST%5Bz0%5D
%29%29%3B&z0=QGluaV9zZXQoImRpc3BsYXlfZXJyb3JzIiwuMCIp00BzZXRfdGltZV9saW1pdCgwKTtAc2
V0X21hZ2ljX3F1b3Rlc19ydW50aW1lKDAp02VjaG8oIi0%2BfCIp0zskRj1nZXRfbWFnaWNfcXVvdGVzX2d
wYygpP3N0cm1wc2xhc2hlcygkX1BPU1RbInoxIl0p0iRfUE9TVFsiejEiXTskZnA9QGZvcGVuKCRGLCJyIi
k7aWYoQGZnZXRjKCRmcCkpe0BmY2xvc2UoJGZwKTtAcmVhZGZpbGUoJEYp031lbHNle2VjaG8oIkVSUk95O
i8vIENhbiB0b3QgUmVhZCIp0307ZWNoYgIfDwtIik7ZGl1KCK7&z1=%2Fhome%2Fwwwroot%2Fdefault
%2Fsqli-labs%2FLess-1%2Findex.phpHTTP/1.1 200 OK
Server: nginx
```

检测模型

利用N-Gram和TF-IDF从原始请求文本计算特征向量。

- N-Gram：对文本序列分词，构建一个长度为N的窗口，在文本上滑动，得到一系列短语。

字符级3-gram：

- 原始文本：hello xkungfoo
- 转换后：[hel, ell, llo, lo , o x, xk, xku, kun, ung, ngf, gfo, foo]

检测模型

TF-IDF向量化：计算每个词的重要程度。

如果某个词或短语在一篇文章中出现的频率TF高，并且在其他文章中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类。

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

$$idf_i = \log \frac{|D|}{|\{j : t_i \in d_j\}|}$$

$$tfidf_{i,j} = tf_{i,j} \times idf_i$$

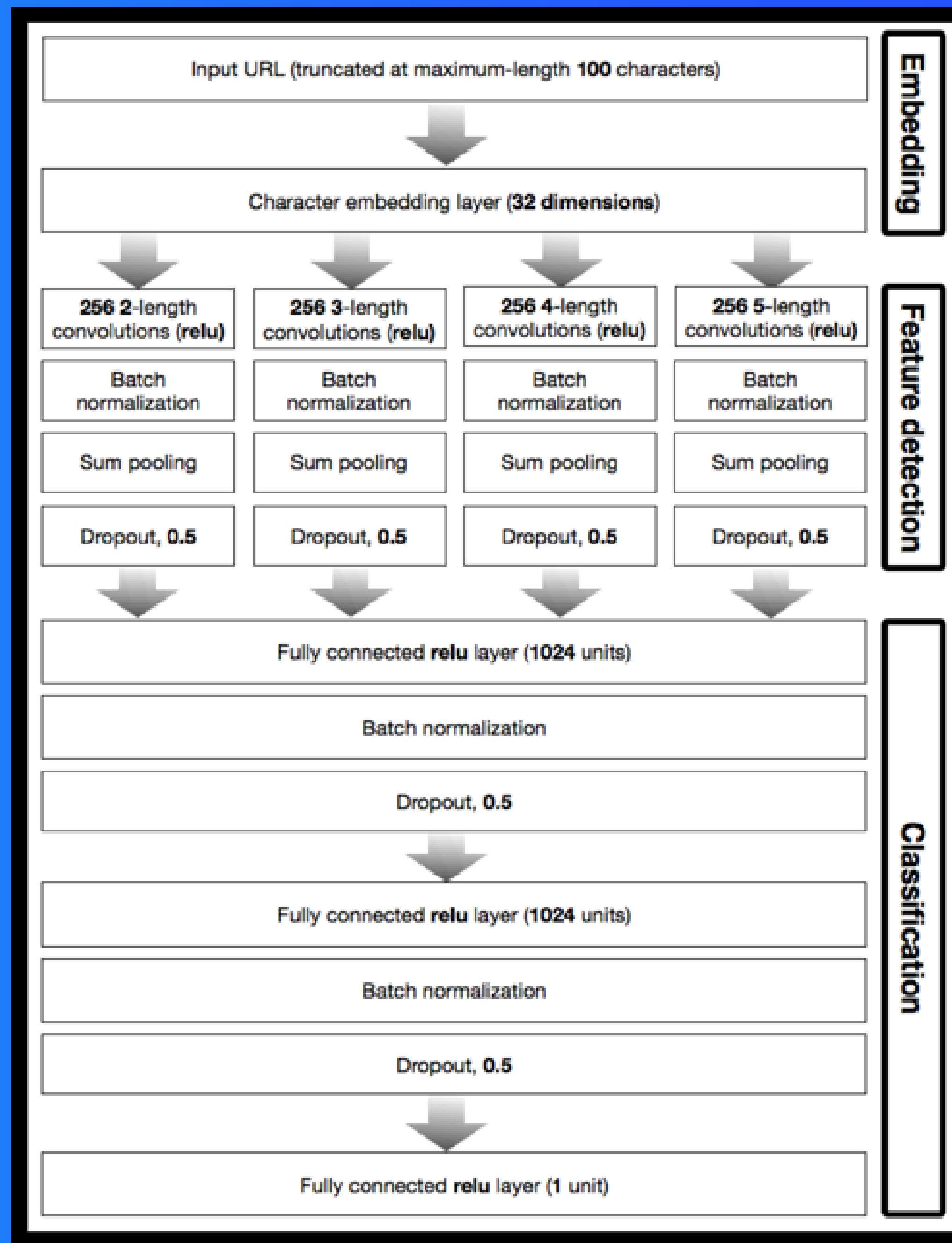
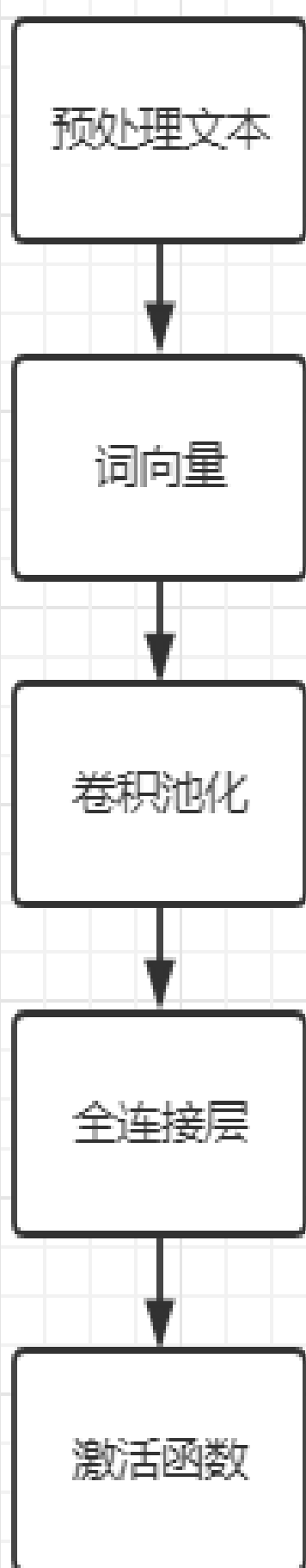
N-Gram + TF-IDF 可发现新特征。
权值高的词可作为关键字加入规则。

检测模型

深度学习方法

文本卷积神经网络：Text-CNN

- 字符级词向量（32维）
- dropout参数：0.5
- 两层全连接



检测模型

机器学习模型为什么能检测Webshell：

- 从请求文本中学习一些特殊关键词和符号
- 发现一些隐藏搭配模式
- 学习到请求长度、信息熵等其他信息

检测模型

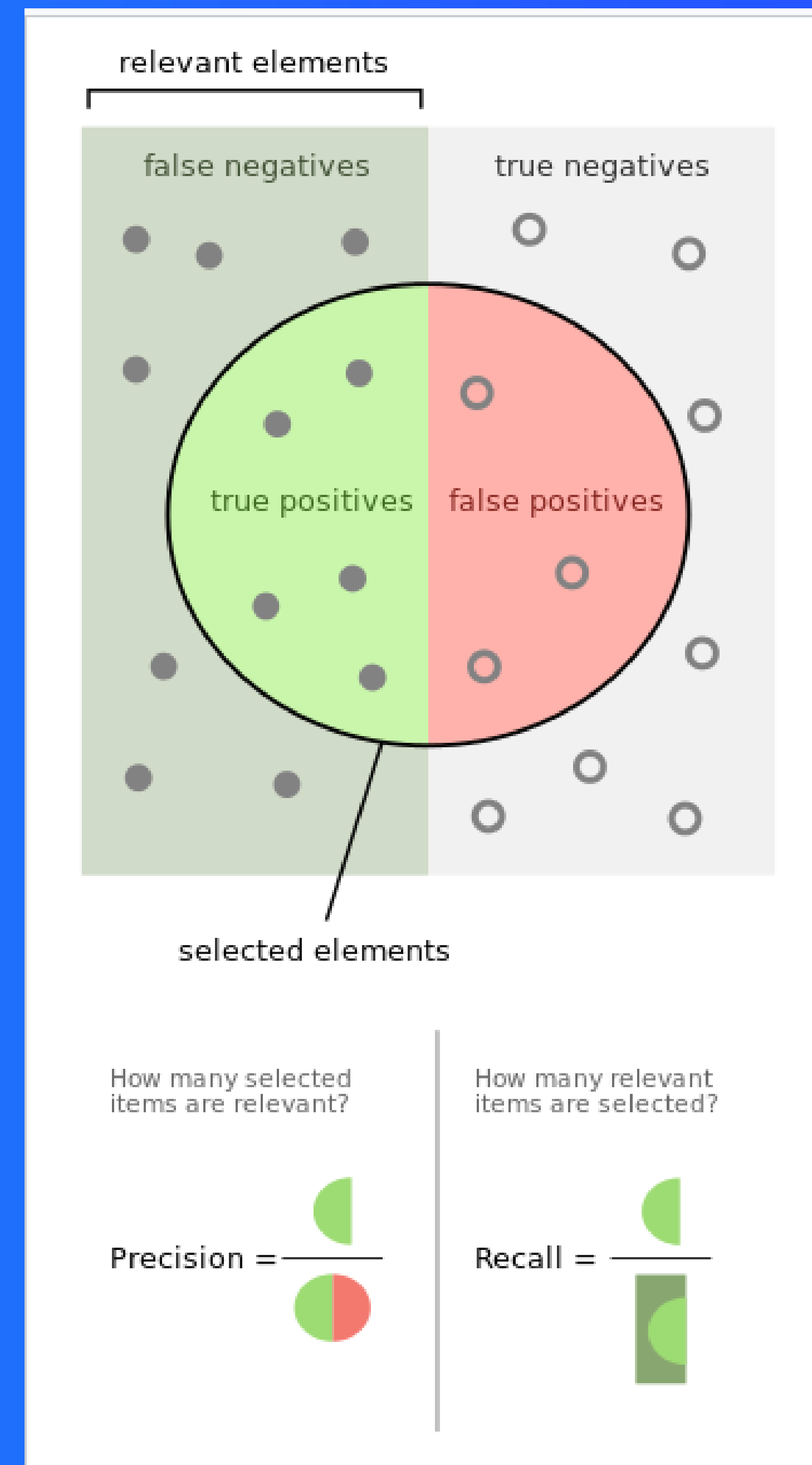
模型评估：

- 既要考虑漏报，又要考虑误报。
- 误报远远比漏报严重！

漏报可由多个安全产品共同解决，误报直接封禁。

$$F_1 = \frac{2}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$



如何处理误报

1. 收集线下训练和线上累积误报样本
2. 对误报样本进行聚类
3. 安全专家对每个簇抽样分析
4. 根据分析结果，将误报样本加入到训练集，扩充规则

0x03 机器学习的攻防对抗

一个安全产品的自我修养

安全产品除了解决安全问题，更重要的是：


- 稳定可靠，不影响业务
- 确保自身应该安全

机器学习检测模型是安全的吗？

绕过机器学习检测模型

对抗样本：对原始样本加入肉眼不可见的微小干扰，使机器学习系统发生误判。

Adversarial Examples



Timeline:

- “Adversarial Classification” Dalvi et al 2004: fool spam filter
- “Evasion Attacks Against Machine Learning at Test Time” Biggio 2013: fool neural nets
- Szegedy et al 2013: fool ImageNet classifiers imperceptibly
- Goodfellow et al **Another interesting thing is that**

(Goodfellow 2016)

对抗样本解释

- 模型没有学习到真正的决策边界
- 模型训练样本不足

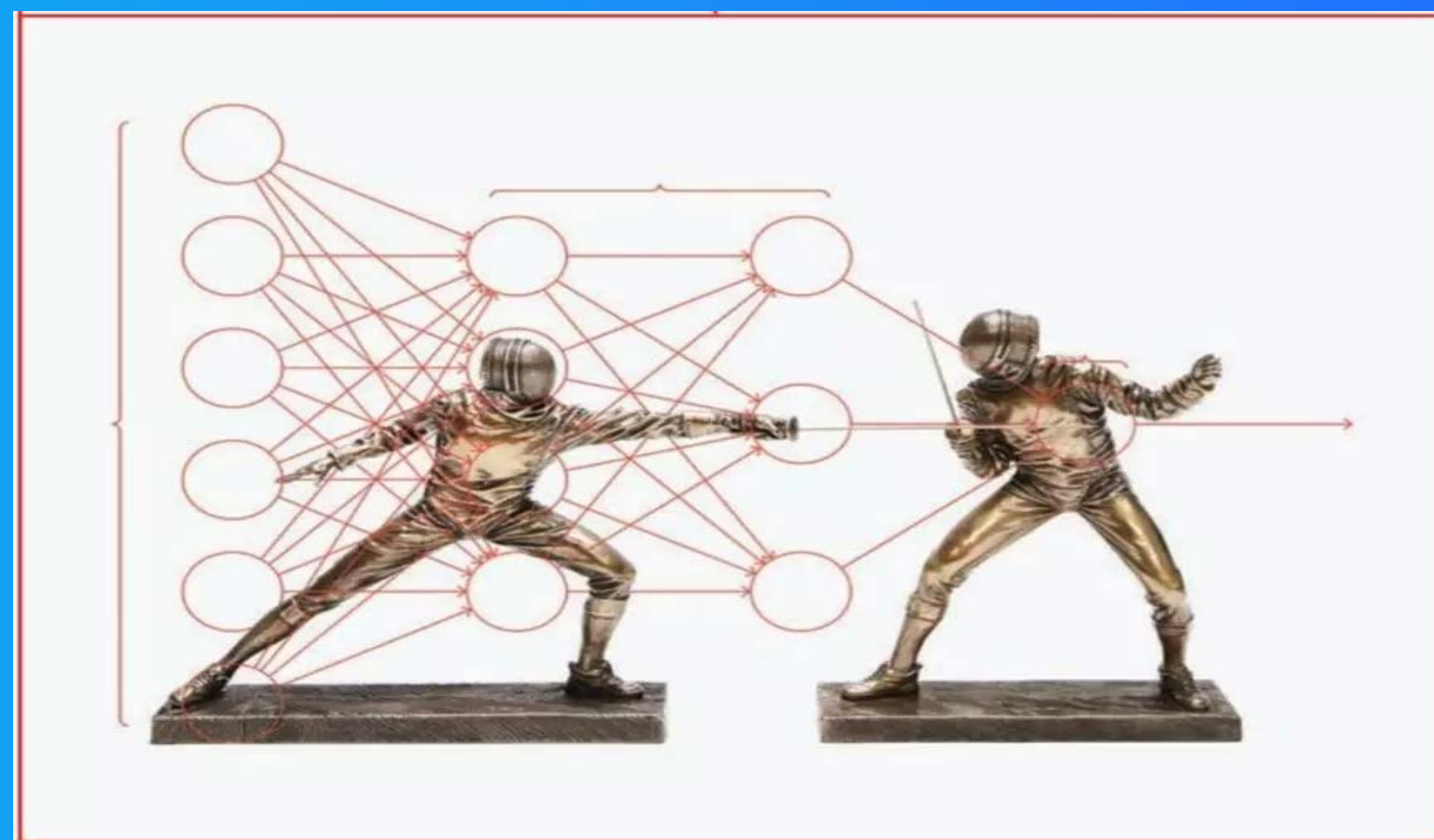
攻击方法：

- Fast gradient sign method
-

机器学习的攻防对抗

更多思考：利用对抗样本绕过机器学习检测引擎？

在请求中改变个别字符，欺骗模型，使机器发生误判，误以为是真正请求。





Thank You!