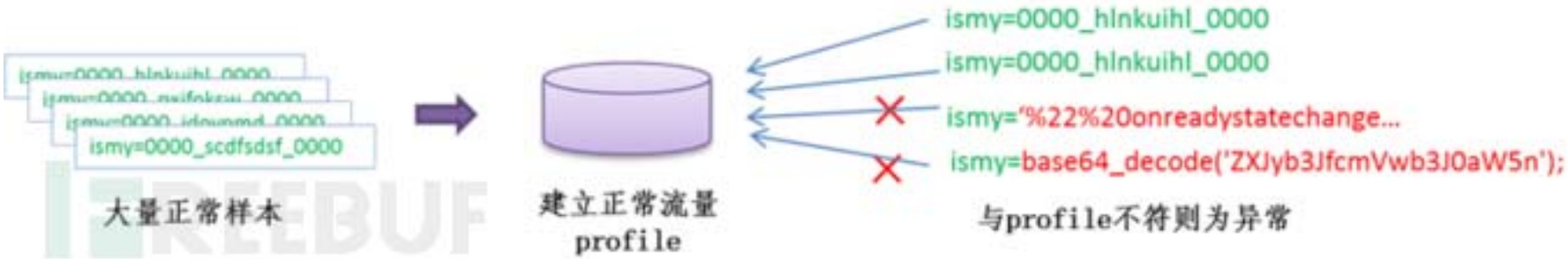
传统机器学习的web异常检测

背景

- 机器学习应用于web入侵检测也存在挑战，其中最大的困难就是标签数据的缺乏。尽管有大量的正常访问流量数据，但web入侵样本稀少，且变化多样，对模型的学习和训练造成困难。因此，目前大多数web入侵检测都是基于无监督的方法，针对大量正常日志建立模型(Profile)，而与正常流量不符的则被识别为异常。这个思路与拦截规则的构造恰恰相反。拦截规则意在识别入侵行为，因而需要在对抗中“随机应变”；而基于profile的方法旨在建模正常流量，在对抗中“以不变应万变”，且更难被绕过。

常规方法



基于异常检测的**web**入侵识别，训练阶段通常需要针对每个**url**，基于大量正常样本，抽象出能够描述样本集的统计学或机器学习模型**(Profile)**。检测阶段，通过判断**web**访问是否与**Profile**相符，来识别异常。

Profile的几种思路

1. 基于统计学习模型

基于统计学习的web异常检测，通常需要对正常流量进行数值化的特征提取和分析。特征例如，URL参数个数、参数值长度的均值和方差、参数字符分布、URL的访问频率等等。接着，通过对大量样本进行特征分布统计，建立数学模型，进而通过统计学方法进行异常检测。

2. 基于文本分析的机器学习模型

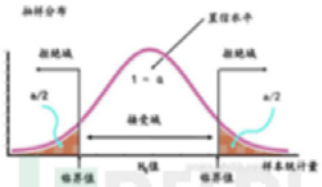
Web异常检测归根结底还是基于日志文本的分析，因而可以借鉴NLP中的一些方法思路，进行文本分析建模。这其中，比较成功的是基于隐马尔科夫模型(HMM)的参数值异常检测。

3. 基于单分类模型

由于web入侵黑样本稀少，传统监督学习方法难以训练。基于白样本的异常检测，可以通过非监督或单分类模型进行样本学习，构造能够充分表达白样本的最小模型作为Profile，实现异常检测。

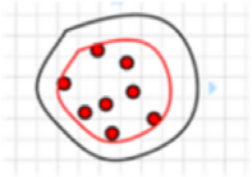
4. 基于聚类模型

通常正常流量是大量重复性存在的，而入侵行为则极为稀少。因此，通过web访问的聚类分析，可以识别大量正常行为之外，小撮的异常行为，进行入侵发现。



统计模型

文本序列HMM



单分类模型

聚类模型

基于统计学习的方法

- 基于统计学习模型的方法，首先要对数据建立特征集，然后对每个特征进行统计建模。对于测试样本，首先计算每个特征的异常程度，再通过模型对异常值进行融合打分，作为最终异常检测判断依据。
- 特征1：参数值value长度
- 特征2：字符分布
- 特征3：参数缺失
- 特征4：参数顺序
- 特征5：访问频率（单ip的访问频率，总访问频率）
- 特征6：访问时间间隔

http://localhost/login?
page=http%3A%2F%2Flocalhost%2Fcheckout%0D%0A%0D%0A%3Cscript%3Ealert%28%27hello%27%29%3C
%2Fscript%3E

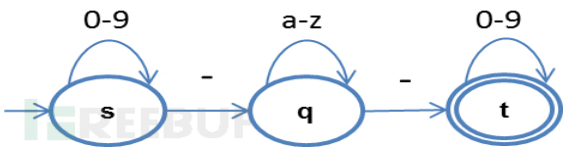
Location: http://localhost/checkout<CRLF>
<CRLF>
<script>alert('hello')</script>

基于文本分析的机器学习模型

- URL参数输入的背后，是后台代码的解析，通常来说，每个参数的取值都有一个范围，其允许的输入也具有一定模式。比如下面这个例子：

```
https://somedomain.com/alibaba/report?mid=6492_abc_7756
https://somedomain.com/alibaba/report?mid=1234_feagada_7680
https://somedomain.com/alibaba/report?mid=2345_hlnkl_9000
https://somedomain.com/alibaba/report?mid=base64_decode
```

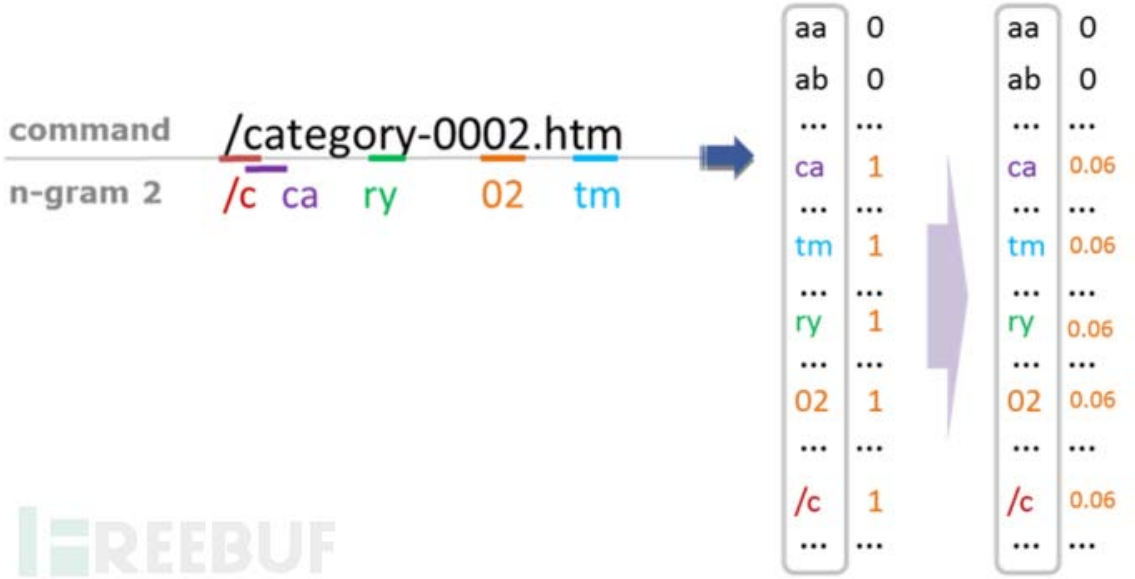
- 例子中，绿色的代表正常流量，红色的代表异常流量。由于异常流量和正常流量在参数、取值长度、字符分布上都很相似，基于上述特征统计的方式难以识别。进一步看，正常流量尽管每个都不相同，但有共同的模式，而异常流量并不符合。在这个例子中，符合取值的样本模式为：数字_字母_数字，我们可以用一个状态机来表达合法的取值范围：



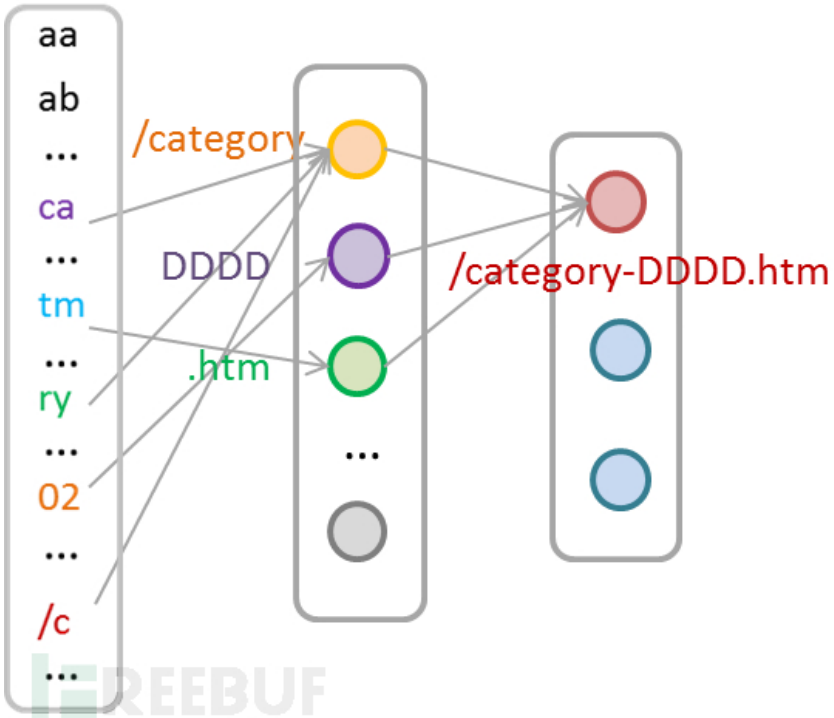
深度学习在Web异常检测中的应用

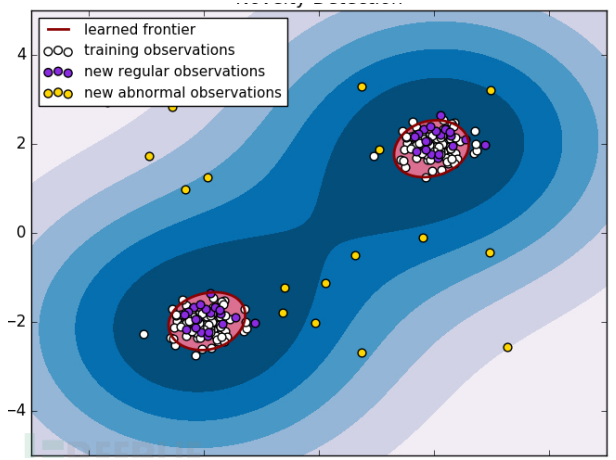
Step1: N-Gram 将文本数据向量化

http://abc.com/test?path=/category-0001.htm
http://abc.com/test?path=/category-0002.htm



直接将向量输入到深度自编码模型，进行训练。测试阶段，通过计算重建误差作为异常检测的标准。





深度学习同机器学习方法一样，深度学习方法也有监督学习与无监督学习之分：

卷积神经网络(**Convolutional neural networks**，简称**CNNs**)

深度置信网络(**Deep Belief Nets**，简称**DBNs**)

深度学习与网络安全

标志性的事件：全球第一个基于深度学习提供商业化网络空间安全解决方案的公司（**Deep Instinct**）于**2015**年**11**月在旧金山成立，该公司宣称其安全解决方案能够抵御未知攻击，能够即时地检测**0-day**漏洞的威胁和**APT**攻击。

2016年**1**月，**Symantec**公司也宣布采用深度学习技术检测利用**0-day**漏洞的病毒工具。

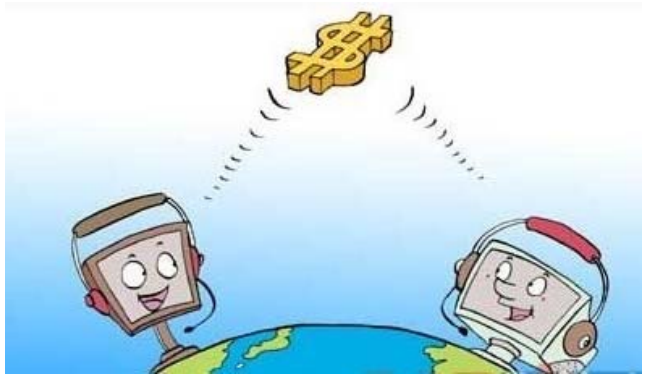
在公共网络语音监管中应用

1.语音犯罪行为增加

2.语音量巨大

3.不同于文本分析（敏感关键词）

4.现有语音识别系统误差率太高



公共网络语音监管原理

首先：将大量语音样本输入到深度学习神经网络中，通过样本得到语音样本的抽象化特征，从而得到语音特征信息库。

然后：将语音信息输入深度学习神经网络，得到语音信息的抽象化表示，与与语音特征库进行比对。

最后：通过一个分类器，可将正常信息与可疑信息区分开来。

正常信息直接忽略，可疑信息在通过语音识别为文本信息在通过人工方式甄别，从而实现对语音信息的分析和预警

基于计算机操作码恶意代码监测

恶意代码是指个人或组织有意编写的对计算机或者网络存在安全隐患的计算机代码，通常包含恶意共享软件、广告软件、木马、病毒、蠕虫等，每一种恶意代码又有不同种类的变种

深度置信网络模型主要分为3大模块：

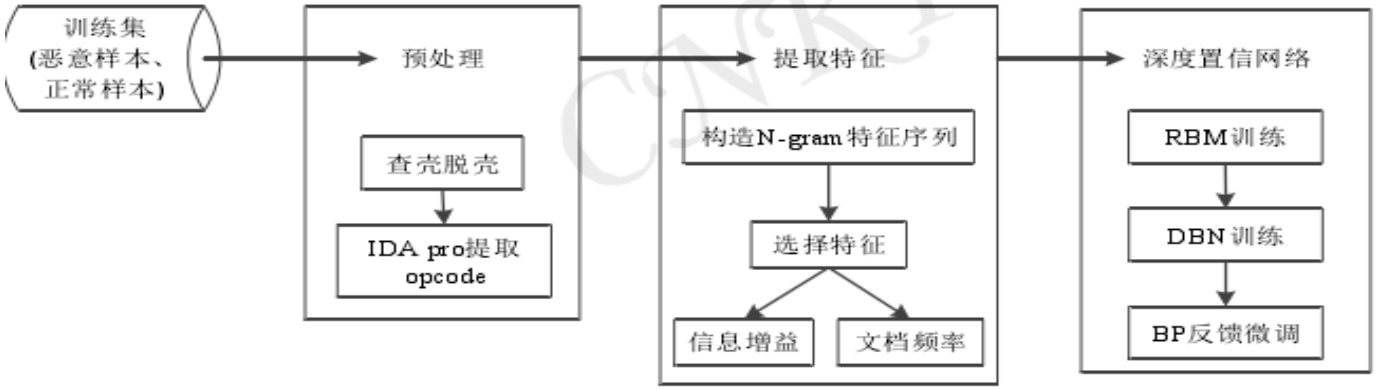
1.数据预处理

2.操作码特征提取

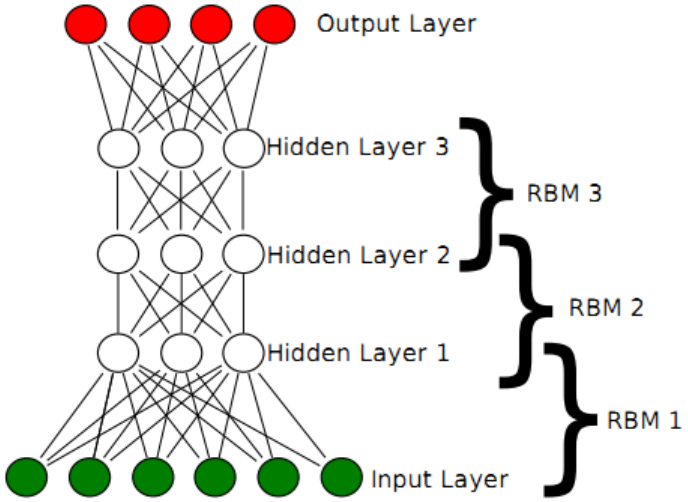
3.深度置信网络模块

1.RBM 预训练，即首先利用大量无类标样本进行受限玻尔兹曼机（RBM）的重构训练，受限玻尔兹曼机的调节过程是自下而上的各个层间的调节过程，以这种方式来初始化整个深度模型的权值；

深度置信网络模块



2. 深度信念网络（**DBN**）无监督的反馈调节，即首先进行自下而上的识别模型转换，然后再进行自上而下的生成模型转换，最后通过不同层次之间的不断调节，使生成模型可以重构出具有较低误差的原样本，这样就得到了此样本的本质特征，即深度模型的最高抽象表示形式；



3. BP 反馈调节，以样本原始类标和目标输出之间的误差进行 **BP** 反馈微调，调节整个网络层数的权值。

入侵检测是一种预防性安全机制，通过对智能手机状态、网络行为等的监控，来发现是否发生用户越权行为或是入侵行为。

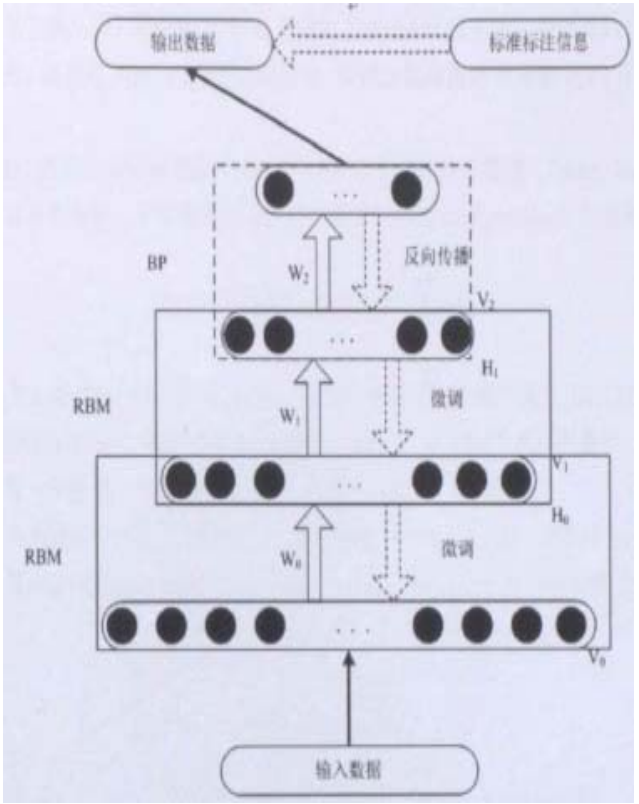
智能手机入侵检测

深度学习入侵检测主要有两个方向：
一是发现入侵的规则、模式，与训练模型匹配对比；
二是用于异常检测，找出用户正常行为，创建用户的正常行为库。

深度信念网络主要由限制玻尔兹曼机模型（**RBM**）和**BP**神经网络两部分

1.针对输入数据进行处理，然后使用**RBM**进行无监督的训练，使得每一层输出的特征较为显著，确保特征向里映射到不同特征空间时，都尽可多地保留特征信息，形成训练模型获取到了较为明显的特征信息；

2.最后一层运用**BP**神经网络来进行分类。**BP**网络层接收**RBM**层输出的特征向量作为它的输入数据，且该层的训练过程是有监督的训练。在进行了设定层数的训练后，将该分类参数设定为**2**，得到最后的误差值，计算出对应准确率。

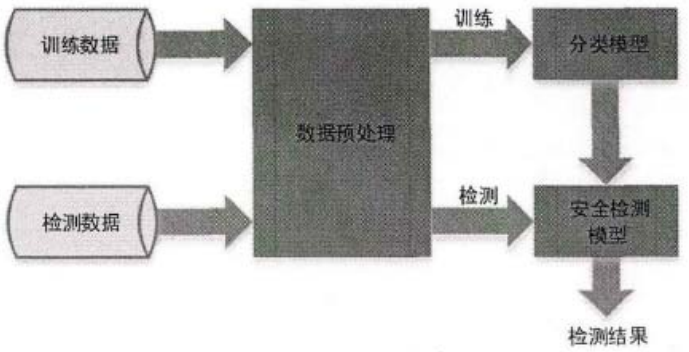


4.1 基于http恶意特征分析

随着**web**应用的发展，**http**协议的使用范围进一步扩大，同时也开始成为网络恶意行为的主要载体，因此请求数据中体现了很多恶意行为特征。

有很多**web**攻击如**SQL**注入、跨站脚本攻击、**cookie**篡改等的恶意行为都在**http**请求中有体现，且请求的攻击方式多变，恶意特征不是只体现在某个特定的地方，还有很多恶意特征集中在路径或者其他部位

首先对**http**请求格式和恶意特征分析，根据数据特点从结构、长度、字符三方面设计了大量特征而且还基于自动生成了一个敏感词库，统计请求内容中的敏感词数目，并将其也作为描述请求的特征之一。



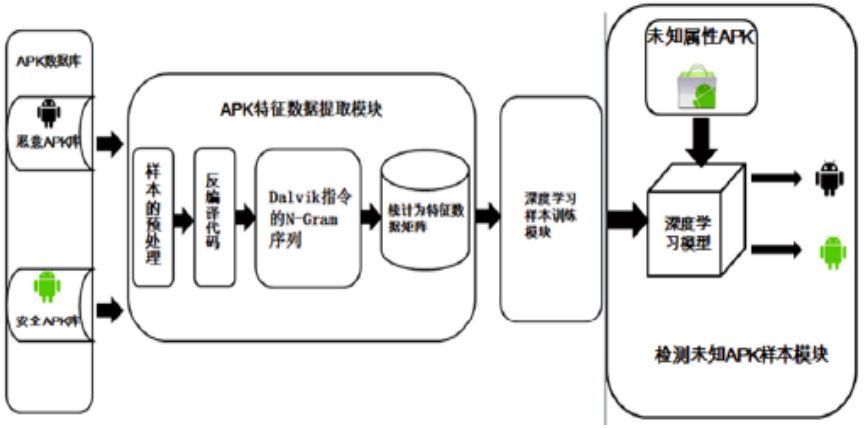
然后采用基于信息熵的特征选择算法，从设计的特征中选择出具有区分度的特征，用这些选择出的特征来向量化请求。在用分类算法训练好安全检测模型之后，就可以用来检测数据的类别了。

手机恶意apk代码监测

1.Andriod 平台的移动智能设备数量和用户数据流量呈指数爆炸式增长

2.基于深度学习的 **Andriod** 恶意应用程序检测系统，突破传统算法效率低的技术屏障，不仅理论上取得了可行性证明，实际验证也获得了较好的检测效果。

1.APK 代码特征的提取模块
2. “训练”模块
3. 检测未知 APK 样本模块



1.采用静态代码分析技术提取 Android应用的多类行为特征数据，

2.将特征数据转化为样本特征矩阵，

3.再用卷积神经网络算法文件来对样本特征矩阵进行训练。

4.最后批量下载未参与训练深度神经网络的 Andriod 应用程序，然后对其APK 执行系统步骤，得到未知样本 APK 的相关预测报告。

深度学习在流量识别中的应用案例分析

流量识别的传统方法（一）

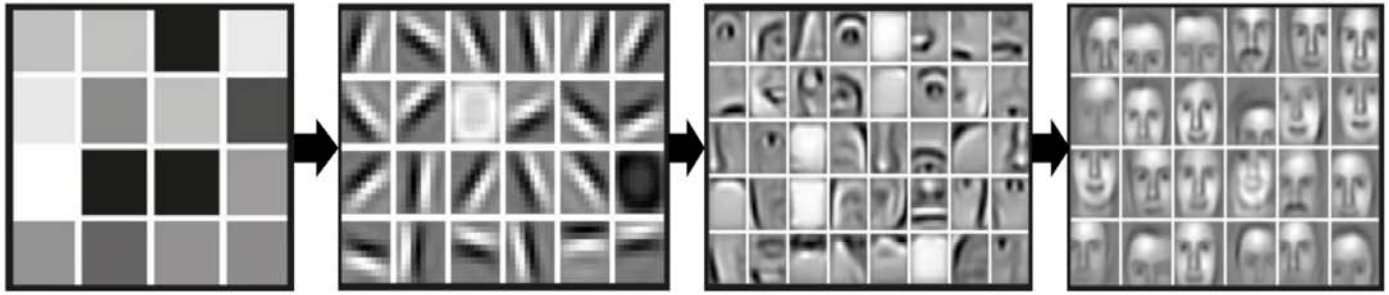
- 将流量准确地映射到某种协议或应用
  - 是网络安全的基础
  - 对异常检测、安全管理作用重大
- 基于预定义或特殊端口
  - 标准HTTP端口：80
  - 默认SSL端口：443
  - 缺点：非标准端口或新定义的端口不适用
- 基于DPI和统计特征的流量识别
  - 根据经验和规则确定的特征字/指纹/序列
  - 缺点：既耗时又耗力

流量识别的传统方法（二）

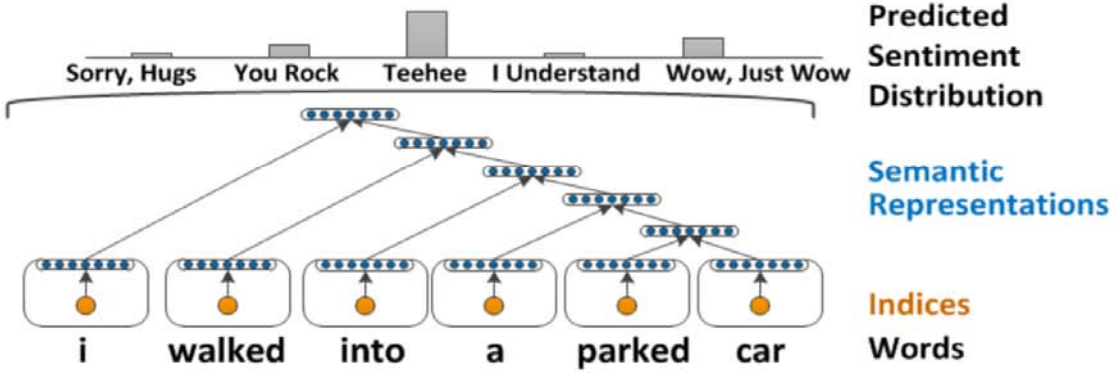
- 基于行为特征和机器学习
  - 优点：建模和识别过程自动化
  - 难点：特征抽取和选择依赖于如何选择特征？
- 有没有不依赖于专家的方法？
- 非监督的特征学习是否可行？
- 答案
  - 人工智能领域的深度学习技术

深度学习的热度

- 图像



- 自然语言处理



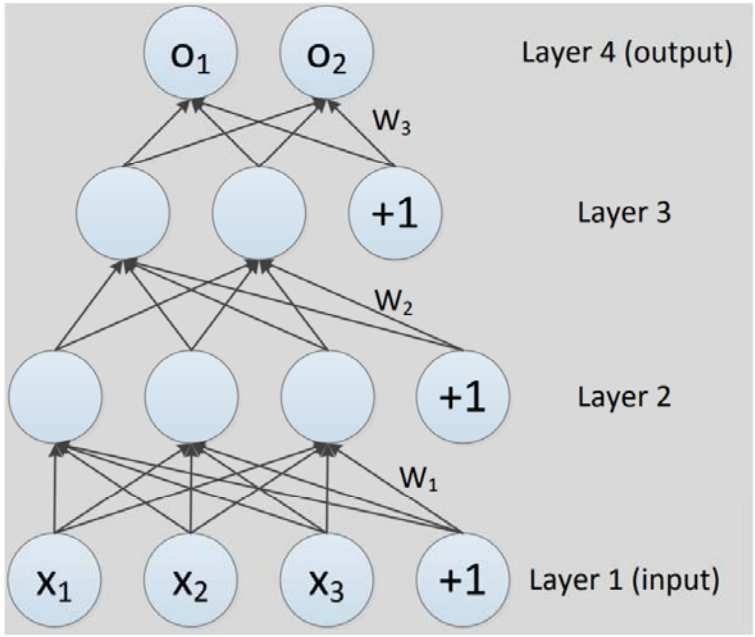
- 语音

深度学习技术的应用

- Gatys, L. A. (2015). A Neural Algorithm of Artistic Style. arXiv preprint arXiv:1508.06576.

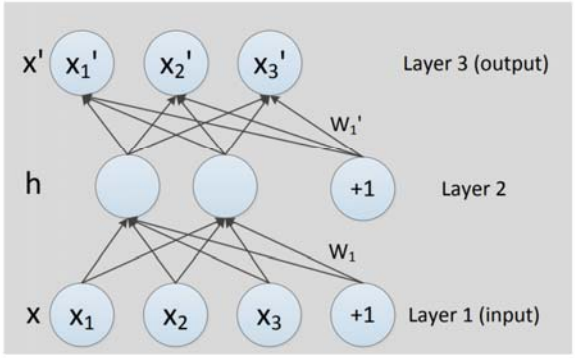
神经网络

- 人工神经网络
- 基本单元
  - 神经元
- 结构
  - 输入层
  - 隐藏层
  - 输出层
- 相邻层的神经元 彼此相连
- 同层的神经元 不直接相连



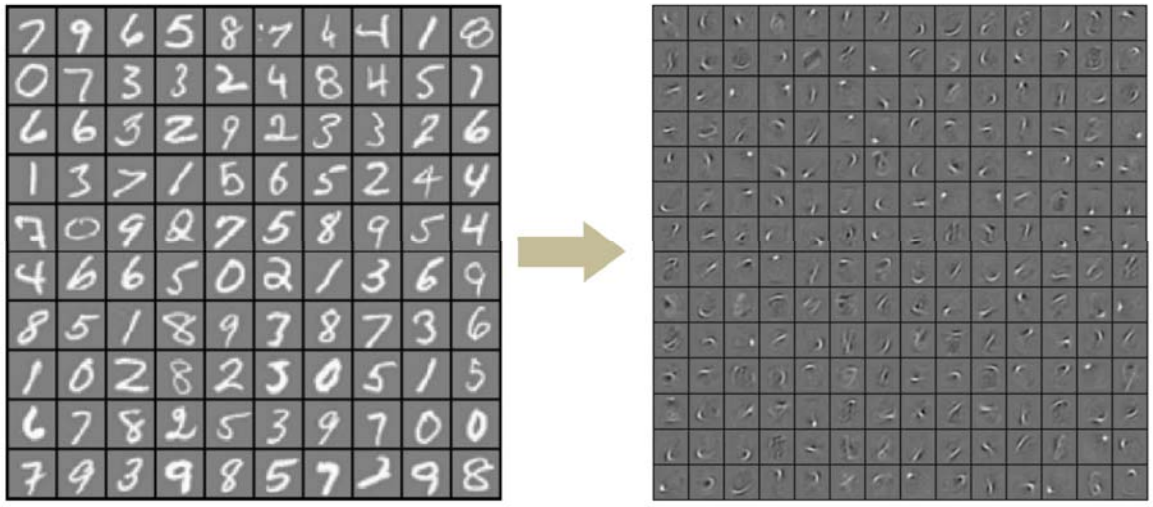
自编码(Auto-Encoder)网络

- 一种特殊的神经网络
- 只有一个隐藏层
- 输出层与输入层完全相同！



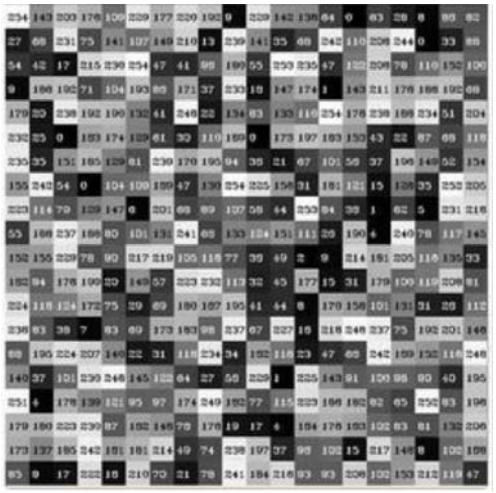
自编码在图像识别中的应用

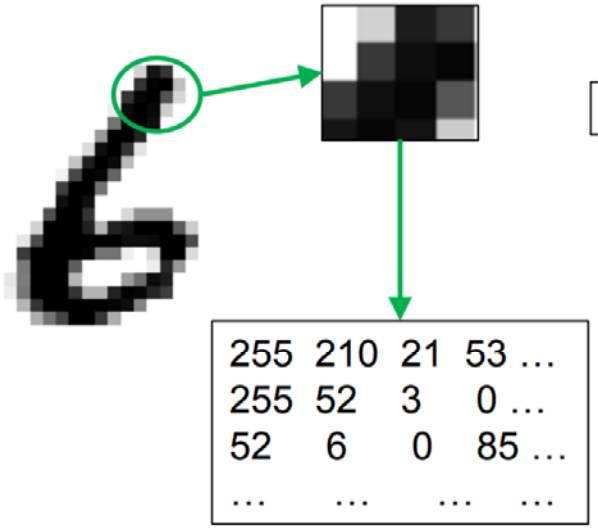
- 手写体数字识别



图像 VS Payload数据

- 是否有相似之处？





TCP flow Payloads

474554206874……727665720020……732048545450……33a31353a323……

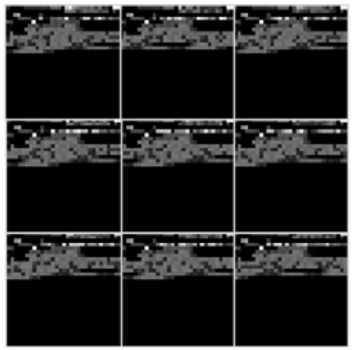
732048545450……33a31353a323……

115 32 72 84 84 80……51 163 19 83 163 35……

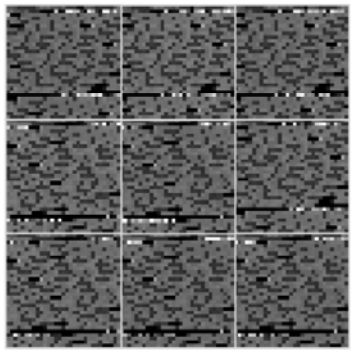
数值范围相同：[0,255]
256个数字!

协议流量→图像

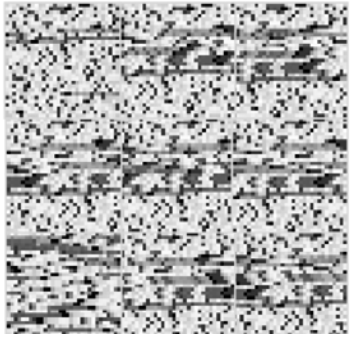
MySQL



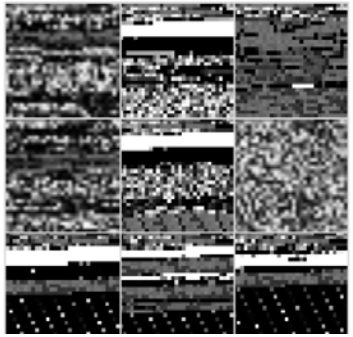
SSH



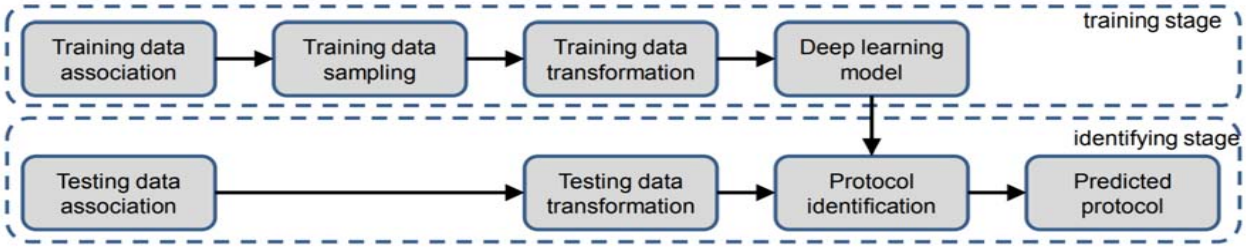
Whois-DAS



BitTorrent



- 实验环境
  - 框架1 - CPU集群: 2~10台服务器
  - 框架2 - CPU + 4GPU
  - 训练时间 - 天->分钟



协议分类结果

- 宏观准确率>99%
- 平均准确率97.9%

Protocol	Precision	Protocol	Precision
SMB	1.0000	RSYNC	0.9987
DCE_RPC	1.0000	Redis	0.9985
NetBIOS	1.0000	FTP_CONTROL	0.9970
TDS	1.0000	HTTP_Connect	0.9967
SSH	0.9996	SMTP	0.9949
Kerberos	0.9996	Whois-DAS	0.9943
LDAP	0.9996	IMAPS	0.9814
BitTorrent	0.9992	Apple	0.9640
MySQL	0.9989	SSL	0.9513
DNS	0.9989	HTTP_Proxy	0.9174

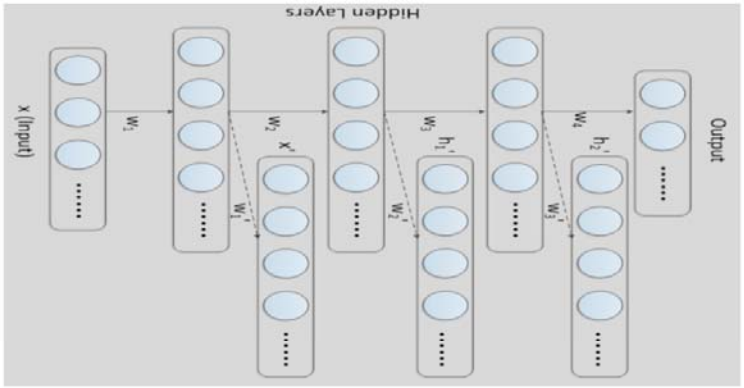
未知协议识别

- 随机选取10,000条被传统方法标记为“unknown”的记录
- 识别率：
- 0%
- 63.37%

	number	ratio
SSL	1956	29.12%
DCE_RPC	1454	21.65%
Skype	873	13.00%
Kerberos	517	7.70%
MSN	360	5.36%
Google	311	4.63%
DNS	260	3.87%
RTMP	234	3.48%
TDS	202	3.01%
H323	170	2.53%

特征的自动学习

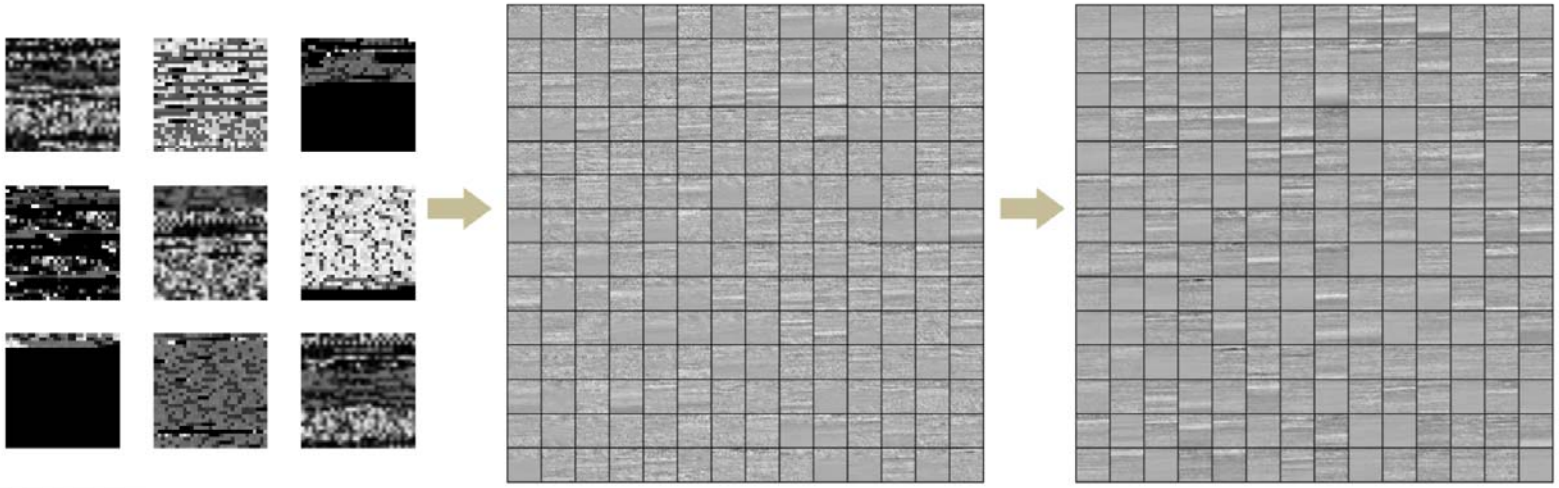
- 特征抽取



原始流量图像

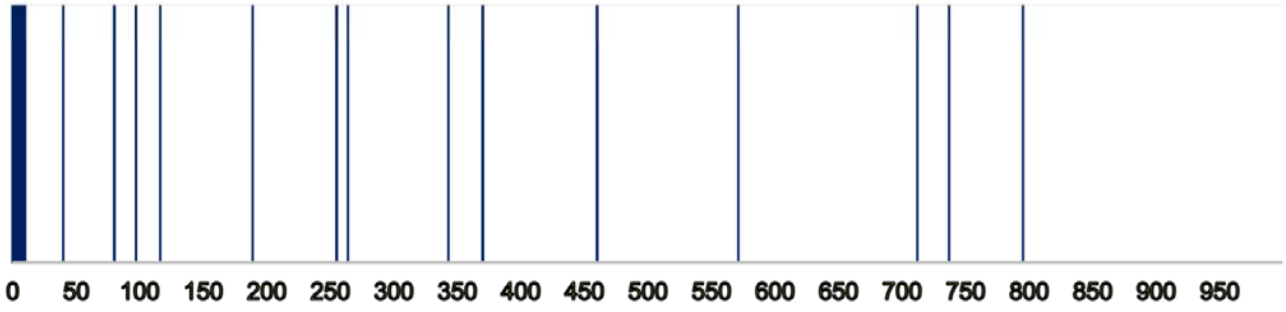
1层AE的特征

2层AE的特征



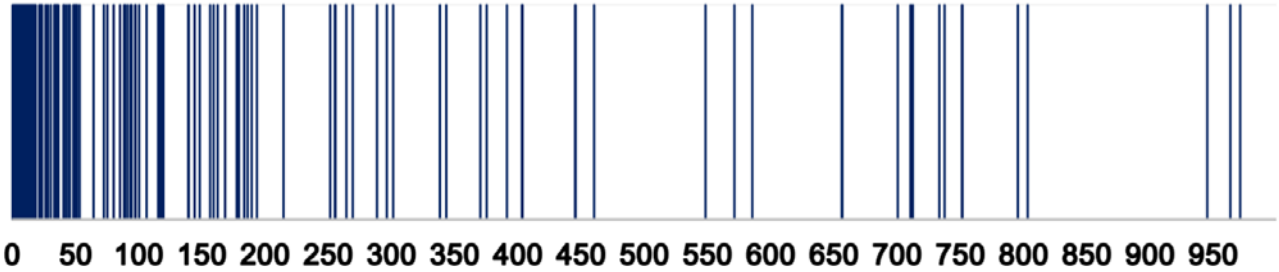
特征的自动学习

- 特征选择
  - A：最重要的25个字节



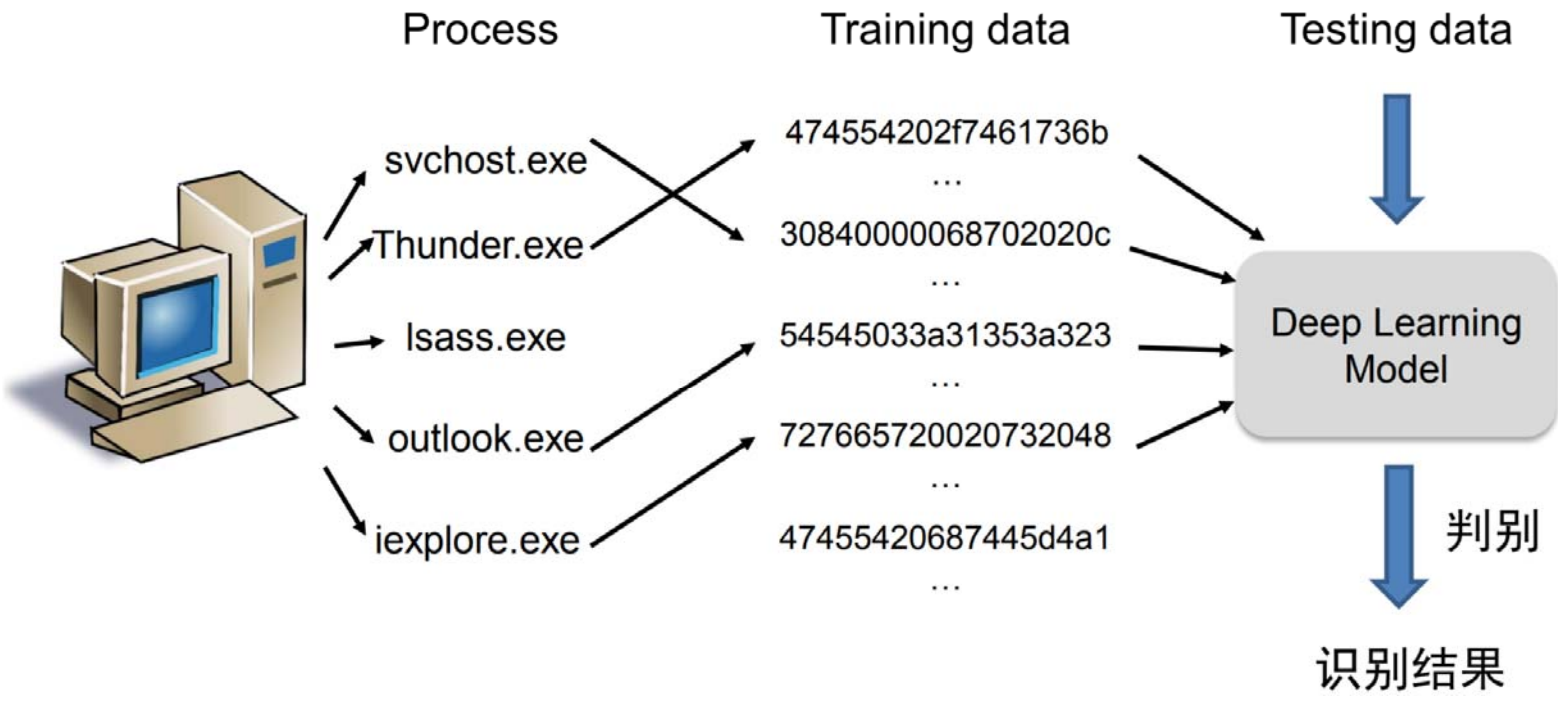
特征的自动学习

- 特征选择
  - B：最重要的100个字节



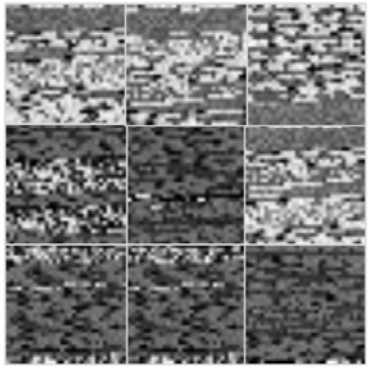
  - C：最不重要的300个字节

应用程序识别

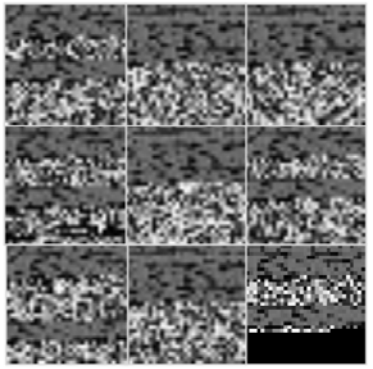

Process
Training data
Testing data
svchost.exe
Thunder.exe
lsass.exe
outlook.exe
iexplore.exe
474554202f7461736b
…
30840000068702020c
…
54545033a31353a323
…
727665720020732048
…
47455420687445d4a1
…
Deep Learning Model
判别
识别结果

应用程序流量→图像

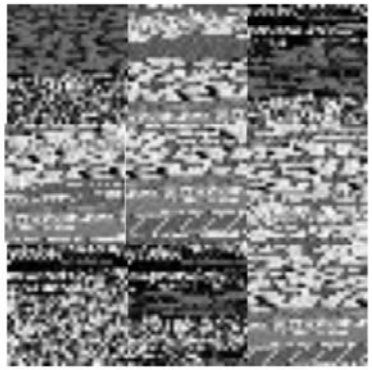
QQ.exe



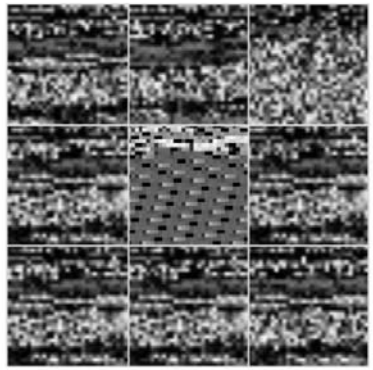
wechat.exe



iexplore.exe



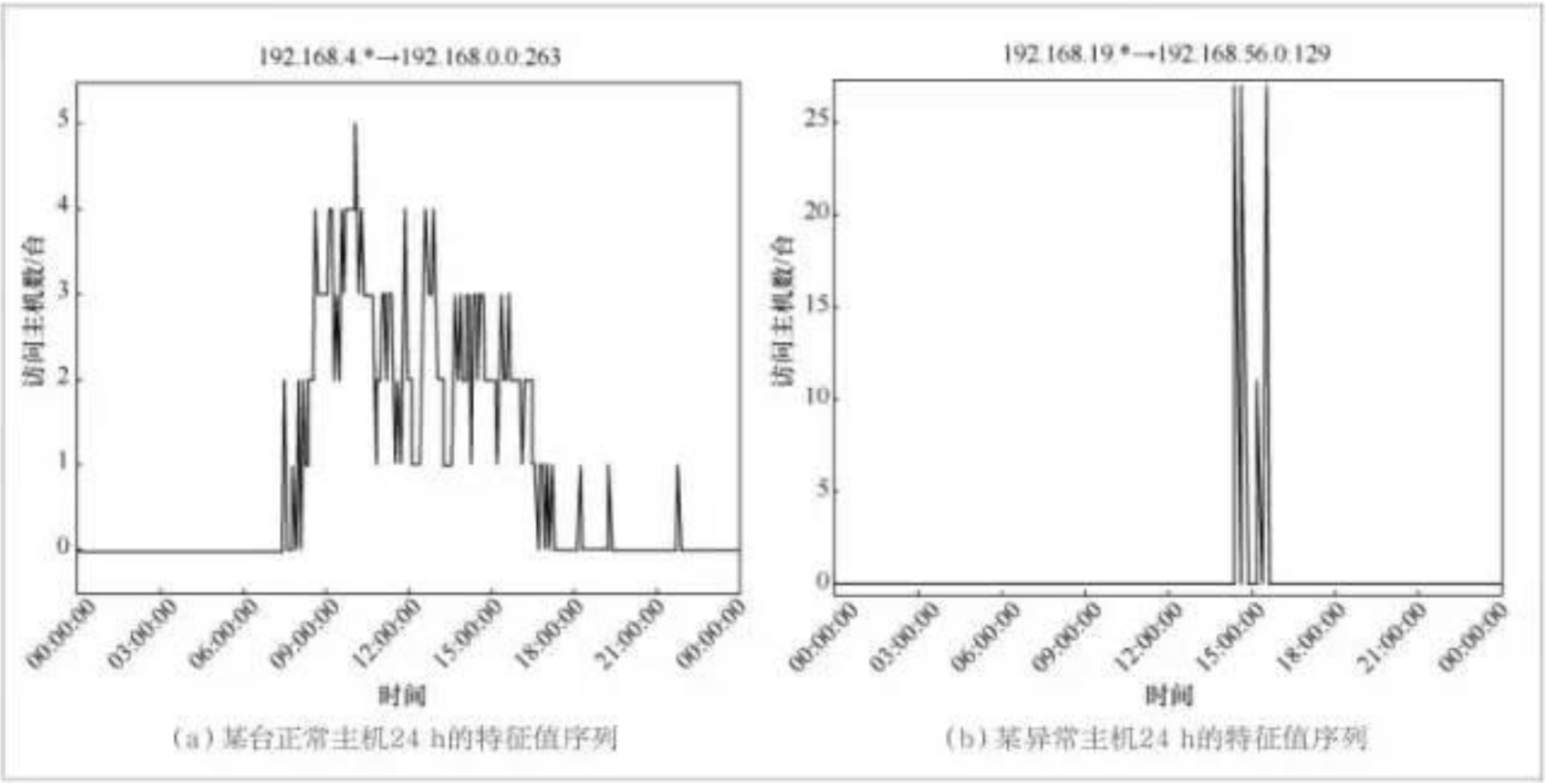
outlook.exe



识别结果

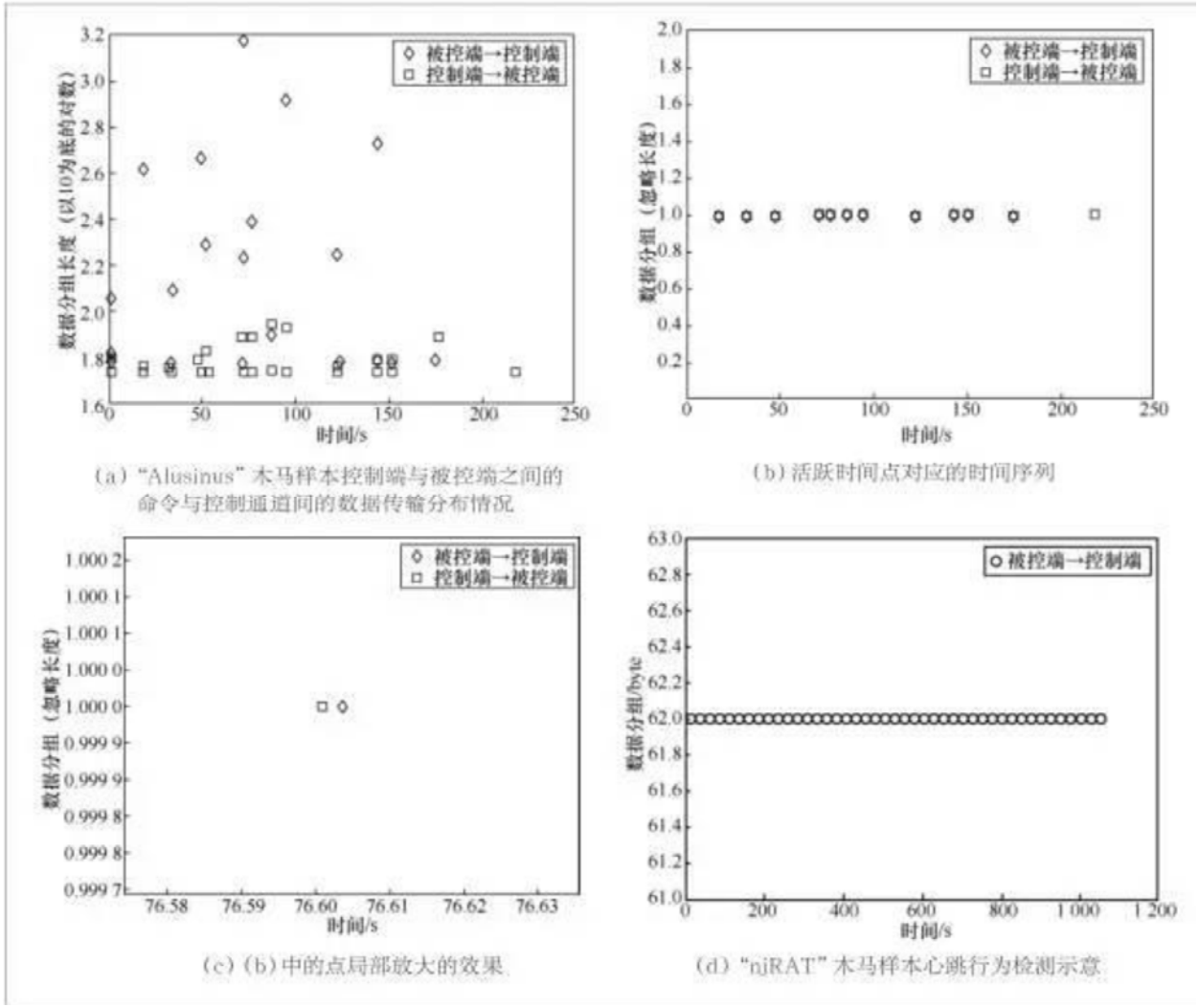
- 训练数据中包含几百种应用
- 宏观准确率>96%，平均准确率>90%

Application	Precision	Protocol	Precision
foxmail.exe	1.0000	xshell.exe	0.9813
wpservice.exe	1.0000	baidumusic.exe	0.9808
taobaoprotect.exe	0.9984	fetion.exe	0.9779
wechat.exe	0.9983	qqmusic.exe	0.9730
liebao.exe	0.9978	qqdownload.exe	0.9615
weibo2015.exe	0.9974	yodaodict.exe	0.9542
lsass.exe	0.9945	itunes.exe	0.9429
sogoucloud.exe	0.9897	outlook.exe	0.9219
qq.exe	0.9884	thunder.exe	0.9168
pplive.exe	0.9870	iexplore.exe	0.8860



(a) 某台正常主机24 h的特征值序列

(b) 某异常主机24 h的特征值序列



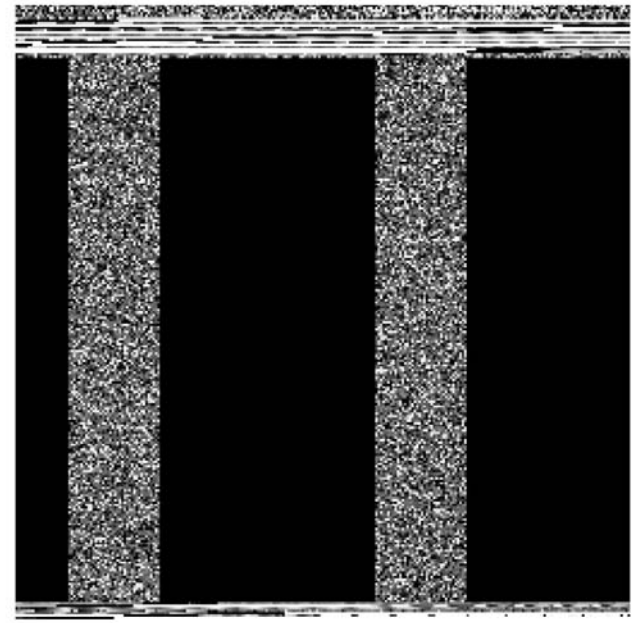
两次连接对应的特征

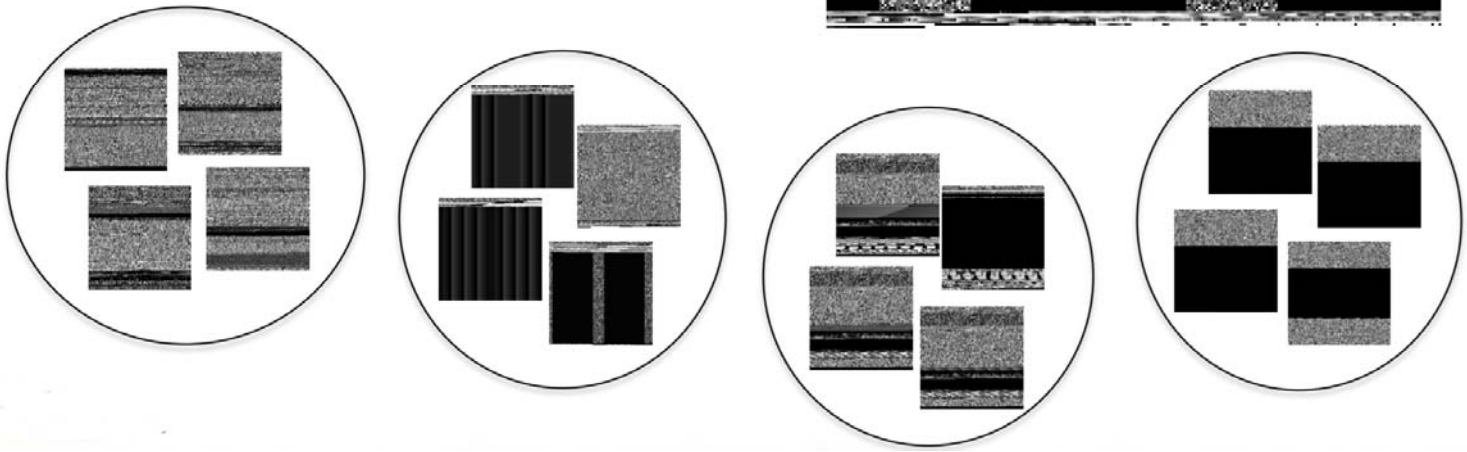
图（a）为“Alusinus”木马样本控制端与被控端之间的命令与控制通道间的数据传输分布情况。从图（a）无法直接观测出是否存在反向连接行为、心跳等行为的特征，但可以看到从被控制端到控制端的数据分组要比反向的数据分组大1个数量级以上，存在明显的上下行流量不对称问题。找出控制端和被控端之间两个方向的活跃时间点，并忽略数据分组大小的影响，可以得到活跃时间点对应的时间序列（如图（b）所示）。可以看到，两个方向的活跃时间点总是成对出现（除了最后一个由控制端到被控端的活跃时间点之外，经验证，该次数据传输的命令为断开连接），响应率为91.7%，激活率为100%。图（c）为将图（b）中的点局部放大的效果，可以看到控制端发送命令在前，被控端响应命令在后。这些都是典型的命令与控制通道的特征。

图（d）为“njRAT”木马样本心跳行为检测示意。本文根据定义5~定义6的计算方法，查找控制端与被控端之间平稳度最高的数据传输序列。从图（d）中可以看到，由被控端发往控制端，大小固定为62 byte的数据分组，平均每19 s发送一次，平稳度高达99.8%，从而被本文所述的检测算法识别为异常行为。

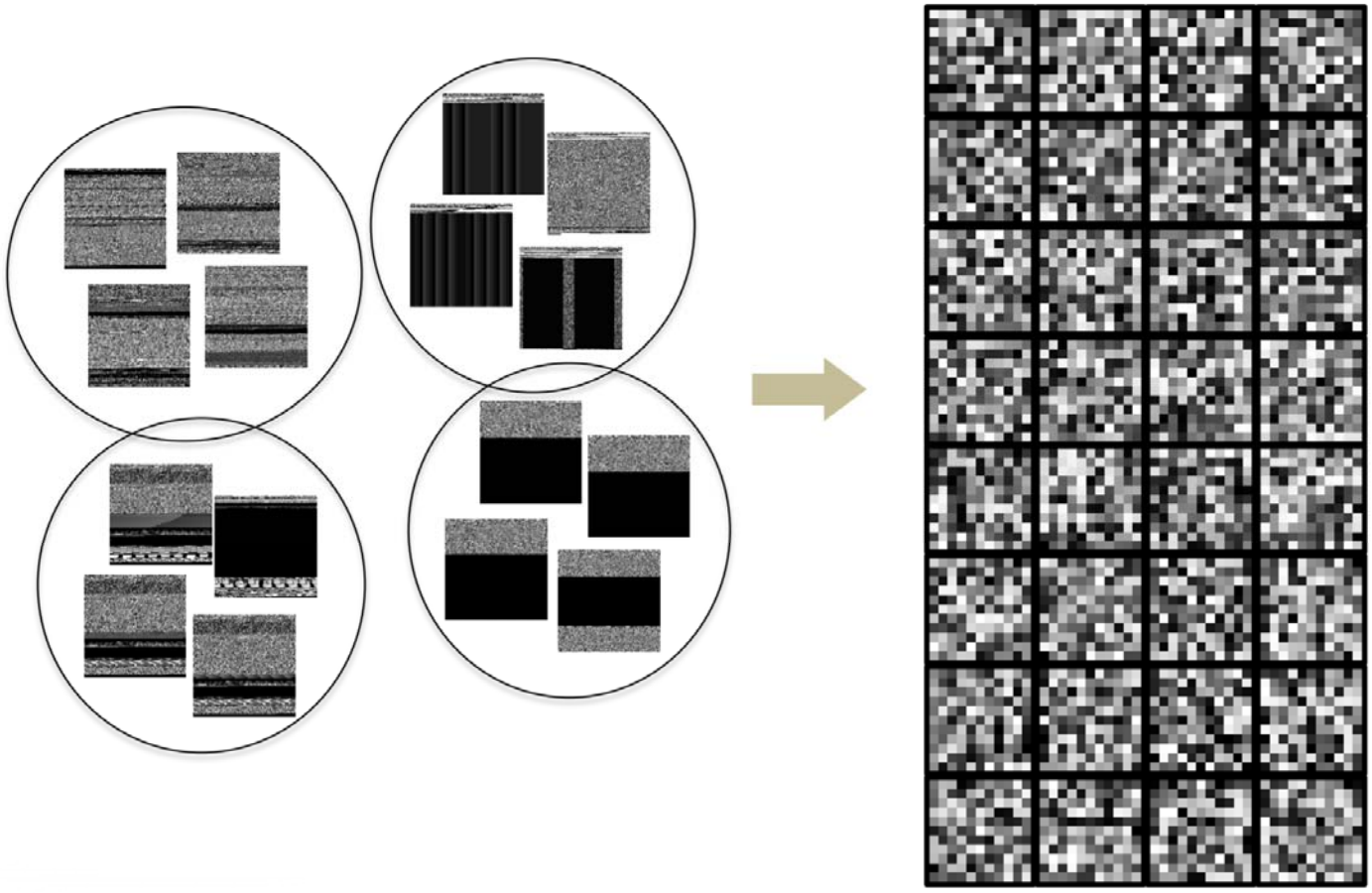
恶意代码样本→图像

```
00401020 60 60 00 33 C5 89 85 98 01 00 00 8B 85 A4 01 00
00401030 00 53 56 8B B5 AC 01 00 00 57 6A 31 89 75 60 A3
00401040 88 70 60 00 C7 05 7C 70 60 00 00 10 40 00 FF 15
00401050 68 90 5F 00 8B 0D 6C 70 60 00 51 FF 15 6C 90 5F
00401060 00 8D 55 38 52 8D 45 48 50 A1 6C 70 60 00 8D 4D
00401070 50 51 8D 55 40 52 50 FF 15 70 90 5F 00 8B 0D 6C
00401080 70 60 00 51 FF 15 74 90 5F 00 33 DB 53 53 FF 15
00401090 98 92 5F 00 8B 15 64 70 60 00 68 10 94 5F 00 68
004010A0 08 94 5F 00 52 FF 15 00 90 5F 00 68 64 70 60 00
004010B0 68 04 94 5F 00 68 01 00 00 80 FF 15 04 90 5F 00
004010C0 8B 45 8C 83 AD 74 FF FF FF 02 F7 D0 66 89 45 A4
004010D0 8B C6 89 5D 7C C7 45 6C 07 00 00 00 8D 50 01 90
004010E0 8A 08 40 84 C9 75 F9 66 8B 0D A2 72 60 00 FF 05
004010F0 4C 72 60 00 2B C2 66 F7 D1 66 89 0D 9E 72 60 00
00401100 3B C3 74 10 8A 06 3C 30 74 0A 80 7E 01 3A 74 04
00401110 3C 33 75 05 BB 01 00 00 00 66 8B 15 92 72 60 00
00401120 8B 0D 24 71 60 00 2B 0D E8 70 60 00 66 83 C2 35
00401130 6A 04 66 89 15 9A 72 60 00 8B 15 F4 70 60 00 23
00401140 15 F8 70 60 00 68 00 10 00 00 68 90 E3 1B 00 B8
00401150 DD 00 00 00 6A 00 C6 05 A6 72 60 00 B8 66 A3 CC
```





样本图像的深度学习（CNN）



小结

- 深度学习在流量识别中的应用
  - 通过网络流量识别协议、应用程序
  - 特征的自动学习
  - 解决大数据的并行计算问题
- 价值
  - 减轻人工负担
  - 精度高
- 应用于网络安全领域的难点——一头一尾
  - 输入：非传统的语音/图像/文本
  - 输出：安全领域往往要求更精准