Beijing Forest Studio

北京理工大学信息系统及安全对抗实验中心

面向恶意软件检测系统的对抗样本攻击

硕士研究生　张荣倩

2020年05月24日

内容提要

- 背景简介
- 基本概念
- 算法原理
- 优劣分析
- 应用总结
- 参考文献

背景简介

背景简介

- 预期收获
  - 1. 了解面向恶意软件检测系统，研究对抗样本攻击的目的
  - 2. 了解该领域对抗样本攻击的经典模型和算法
  - 3. 明确同面向图像领域进行对抗样本攻击的区别
  - 4. 了解对抗样本攻击日后发展方向和应用领域

背景简介

- 恶意软件检测
  - 分类
    - 签名 signatures
    - 启发式技术
      - 静态分析 static analysis
      - 动态分析 dynamic analysis
  - 检测质量：取决于所提供的特征
  - 评价指标
    - 正确（accuracy）
    - 漏报（false positive）
    - 误报（false negative）

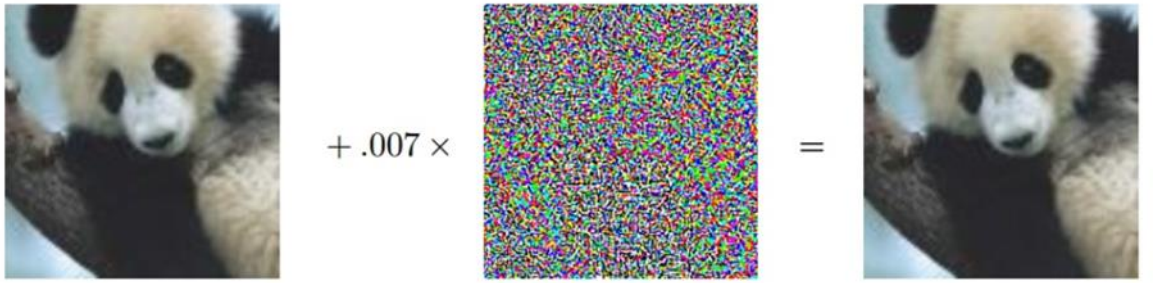
背景简介

- 最主要的问题：如何有效地提高恶意软件检测系统的鲁棒性?

- 研究目的
  - 评估分类模型的抗攻击性
  - 优化模型，让分类模型达到更加高的分类准确率

- 对抗样本攻击
  - 对输入样本故意添加一些人无法察觉的细微的干扰，导致模型以高置信度给出一个错误的输出。

背景简介



$\boldsymbol{x}$
"panda"
57.7% confidence

$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence

$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

- 分类
  - 白盒攻击&黑盒攻击
  - 目标攻击&非目标攻击

- 从最初的样本x制作对抗性样本x*

$$\boldsymbol{x}^* = \boldsymbol{x} + \delta_{\boldsymbol{x}} = \boldsymbol{x} + \min \|\boldsymbol{z}\| \text{ s.t. } \mathbf{F}(\boldsymbol{x} + \boldsymbol{z}) \neq \mathbf{F}(\boldsymbol{x})$$

背景简介

$$x^* = x + \delta_x = x + \min \|z\| \text{ s.t. } \mathbf{F}(x+z) \neq \mathbf{F}(x)$$

- 问题：非线性非凸性，很难找到闭式解
- 解决方法
  - One Pixel Attack
  - FGSM
  - JSMA
  - DeepFool
  - Papernot Method
  - CW（The Carlini and Wagner）

基本概念

基本概念

- 对抗样本可转移性
  - Delving into Transferable Adversarial Examples and Black-box Attacks
  - 针对一个模型制作的对抗样本，也可能对其他模型有效
  - 方法
    - 攻击者训练替代模型，该模型同原始模型具有相似性，然后对替代模型执行白盒攻击。
    - 使用目标DNN的置信度得分直接来估算它的梯度，而不是使用替代模型的梯度生成对抗样本。（黑盒情况下不可行）

算法原理

算法原理

- 目标
  - 利用生成的对抗样本，对恶意软件检测模型进行高效攻击

- 挑战

  与图像分类任务相比
  - 可接受扰动的范围大大减小：模型输入现在是具有离散值的一组特征。在离散且通常为二进制的输入域中运行，而以前的工作仅在连续且可微分的域中运行。-高效性
  - 保证恶意软件的功能性 –可行性

算法原理

- MFGSM
  - Adversarial examples for malware detection （2017）
  - 白盒

- 模型
  - 基于FGSM改进

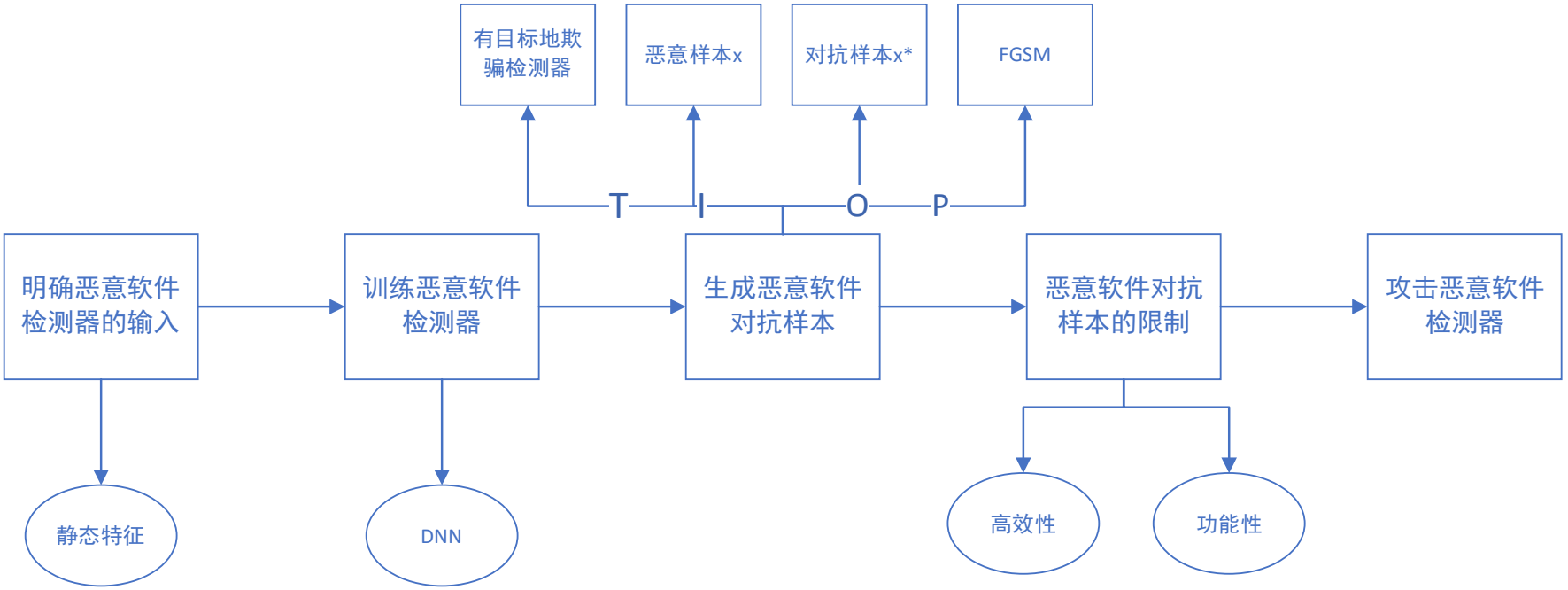
- 实验效果
  - 误导了所有恶意软件样本中63%的分类器

算法原理

T	生成对抗样本，训练出可以以高成功率欺骗恶意软件检测系统的攻击模型
I	恶意软件样本x
P	在原始样本中添加微小扰动δ，最大化误差函数，生成对抗样本x*
O	生成对抗样本攻击模型

P	尽可能达到较高的逃避率
C	保证恶意软件功能不被影响的同时，提高逃避率
D	如何将面向图像分类算法应用到恶意软件领域中
L	ESORICS CCF-B类会议

算法原理


有目标地欺骗检测器
恶意样本x
对抗样本x*
FGSM
T
I
O
P
明确恶意软件检测器的输入
训练恶意软件检测器
生成恶意软件对抗样本
恶意软件对抗样本的限制
攻击恶意软件检测器
静态特征
DNN
高效性
功能性

算法原理

- 算法模型（FGSM）

Algorithm 1 Crafting adversarial examples for Malware Detection

Input: x, y, F, k, I

$\mathbf{x}^* \leftarrow \mathbf{x}$

$\Gamma = \{1 \dots |\mathbf{x}|\}$

while $\arg\max_j \mathbf{F_j}(\mathbf{x}^*) \neq \mathbf{y}$ and $||\delta_\mathbf{X}|| < \mathbf{k}$ **do**

  Compute forward derivative $\nabla \mathbf{F}(\mathbf{x}^*)$

  $i_{max} = \arg\max_{j \in \Gamma \cap \mathbf{I}, X_j = 0} \frac{\partial \mathbf{F}_y(\mathbf{X})}{\partial \mathbf{X}_j}$

  if $i_{max} \leq 0$ **then**

    return Failure

  end if

  $\mathbf{x}^*_{i_{max}} = 1$

  $\delta_\mathbf{x} \leftarrow \mathbf{x}^* - \mathbf{x}$

end while

return $\mathbf{x}^*$

$x \in \{0.1\}^m$　X：特征的二进制向量　F：神经网络模型

x：原始样本　x*:对抗样本　X_i：是否具有特征i

1. 计算F相对于X的梯度，估计X产生的扰动所改变F输出的方向
2. 选择能产生最大正梯度的扰动δ
3. 重复1.2.过程，直到（a）达到了限制点（b）成功造成错误分类

算法原理

- 问题
  - 算法1特征的更改可能会导致所涉及的应用程序部分或全部失去其恶意软件的功能
  - 特征之间的相互依赖性

- 解决方法
  - 限制最大失真δ $\|\delta\| \leq k$ k=20
  - 仅通过添加对应单行代码的方式修改特征
  - 只更改特征“manifest”

算法原理

- MFGSM
  - 实现了针对Android静态分析恶意软件分类器的白盒规避技术。
  - 不足
    - 仅涉及静态特征，而未涉及RNN或动态特征
    - 没考虑可能影响多种分类器的通用攻击
    - 白盒的假设在现实中缺乏可行性

算法原理

- MAPI
  - Generic Black-Box End-to-End Attack Against State of the Art API Call Based Malware Classifiers (2018)
  - 黑盒

- 方法
  - 通过插入API序列产生对抗样本

- 实验效果
  - 多组实验达到接近100%的逃逸率

算法原理

T	生成对抗样本，训练出可以以高成功率欺骗恶意软件检测系统的攻击模型
I	恶意软件样本
P	通过插入API序列生成对抗样本
O	生成对抗样本攻击模型

P	尽可能达到较高的逃避率
C	黑盒模型，不改变恶意软件功能性
D	如何在不知道检测模型结构的前提下，达到逃避恶意软件检测的目的
L	RAID CCF-B类会议

算法原理

- 算法模型
  - 使用目标分类器O创建其替代模型F ,从而近似黑盒模型O的决策边界（papernot method）

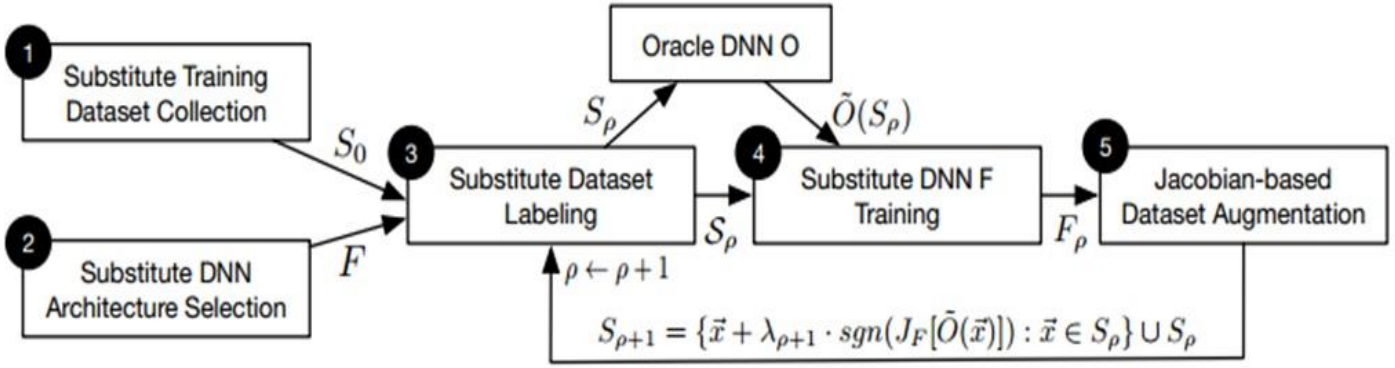


Figure 3: **Training of the substitute DNN** F: the attacker (1) collects an initial substitute training set S_0 and (2) selects an architecture F. Using oracle $\tilde{O}$, the attacker (3) labels S_0 and (4) trains substitute F. After (5) Jacobian-based dataset augmentation, steps (3) through (5) are repeated for several substitute epochs ρ.

  - 挑战
    - 不知道原始模型的结构信息
    - 为了更好处理，限制向原始模型询问（输入输出）的次数

算法原理

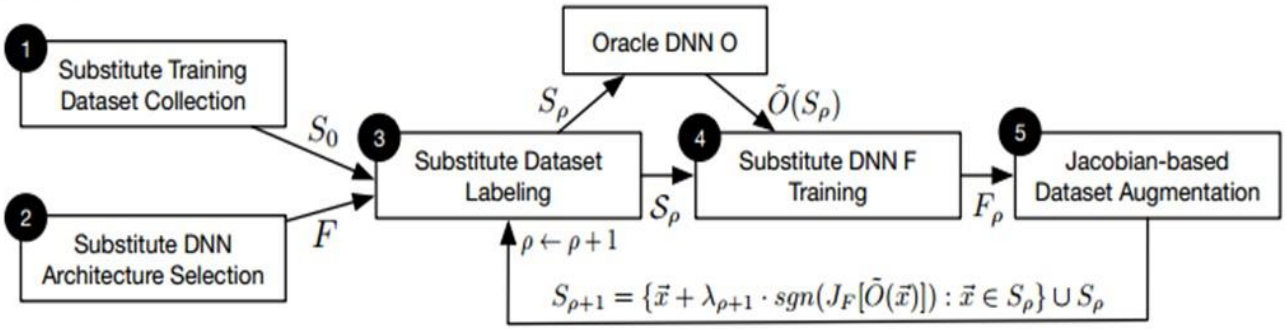


Figure 3: **Training of the substitute DNN** F: the attacker (1) collects an initial substitute training set S_0 and (2) selects an architecture F. Using oracle $\tilde{O}$, the attacker (3) labels S_0 and (4) trains substitute F. After (5) Jacobian-based dataset augmentation, steps (3) through (5) are repeated for several substitute epochs ρ.

(1) 收集输入数据 S_0

(2) 选择模型的结构 F

(3) 训练替代模型Fρ

- 标数据
- 训练模型F
- 数据集增强 Sρ

重复ρ次迭代，以提高F的准确率并让其决策边界和模型O更相似

算法原理

– 通过替代模型生成对抗序列

Algorithm 2 Adversarial Sequence Generation

Input: f (black-box model), $\hat{f}$ (surrogate model), $\mathbf{x}$ (malicious sequence to perturb, of length l), n (size of adversarial sliding window), D (vocabulary)

for each sliding window $\mathbf{w}_j$ of n API calls in $\mathbf{x}$:

  $\mathbf{w_j}^* = \mathbf{w_j}$

  while $f(\mathbf{w}_j^*) = malicious$:

    Randomly select an API's position i in $\mathbf{w}$

    # Insert a new adversarial API in position $i \in \{1..n\}$:

    $\mathbf{w}_j^*[i] = \arg\min_{api} ||sign(\mathbf{w_j}^* - \mathbf{w}_j^*[1 : i-1] \perp api \perp \mathbf{w}_j^*[i : n-1]) - sign(J_{\hat{f}}(\mathbf{w_j})[f(\boldsymbol{w_j})])||$

  Replace $\mathbf{w}_j$ (in $\mathbf{x}$) with $\mathbf{w_j}^*$

return (perturbed) $\boldsymbol{x}$

- API调用序列可能很长，不可能一次对整个序列进行训练
- 解决方法：滑动窗口方法
- x 长度为l的恶意扰动序列 n 对抗滑动窗口大小
- wj :滑动窗口 D:分类器记录的API调用

算法原理

Algorithm 2 Adversarial Sequence Generation

Input: f (black-box model), $\hat{f}$ (surrogate model), $\mathbf{x}$ (malicious sequence to perturb, of length l), n (size of adversarial sliding window), D (vocabulary)

for each sliding window $\mathbf{w}_j$ of n API calls in $\mathbf{x}$:

$\mathbf{w}_j{}^* = \mathbf{w}_j$

while $f(\mathbf{w}_j^*) = malicious$:

Randomly select an API's position i in $\mathbf{w}$

\# Insert a new adversarial API in position $i \in \{1..n\}$:

$\mathbf{w}_j^*[i] = \arg\min_{api} ||sign(\mathbf{w}_j{}^* - \mathbf{w}_j^*[1 : i-1] \perp api \perp \mathbf{w}_j^*[i : n-1]) - sign(J_{\hat{f}}(\mathbf{w}_j)[f(\mathbf{w}_j)])||$

Replace $\mathbf{w}_j$ (in $\mathbf{x}$) with $\mathbf{w}_j{}^*$

return (perturbed) $\boldsymbol{x}$

- 步骤
  - 在w中随机选择一个位置I
  - 在D中找API调用api （最接近雅可比矩阵所指示的方向）
  - 迭代的应用，直到发现一个对抗性输入序列被分类为良性

优劣分析

优劣分析

- FGSM vs MFGSM

	FGSM	MFGSM
应用领域	图像分类	恶意软件检测
输入数据	连续可微分	离散，二进制域
目标	相似性高	不危害恶意软件功能的前提下，提高逃避率

优劣分析

- MFGSM vs MAPI

	MFGSM	MAPI	影响
白盒/黑盒	白盒	黑盒	白盒在现实中缺乏可行性
逃逸率	63%	多组实验接近100%	尽可能提高逃逸率
应用检测器	DNN	多种分类器（RNN变体、DNN、传统机器学习分类器）	攻击可以绕过基于多功能特征的恶意软件分类器
验证功能性	未验证	验证	保证对抗软件的恶意软件功能

应用总结

应用总结

- 应用领域
  - 人脸识别系统(Face Recognition)
  - 语义分割网络(Semantic Segmentation)
  - 目标检测系统(Object Detection)
  - 自然语言处理(Natural Language Processing)
  - 恶意软件检测(Malware Detection)
  - 强化学习(Reinforcement Learning)

应用总结

- 未来发展
  - 对抗样本背后的因果关系的探索
  - 针对此类攻击研究其防御机制，设计更有效、更强大的攻击用来评估新兴的防御系统
    - 检测对抗样本
    - 使分类器具有对抗攻击能力
  - 实现物理世界中的对抗攻击（那些对抗性攻击是否会对物理世界形成真正威胁）

参考文献

[1] I. J. Goodfellow et al. Explaining and harnessing adversarial examples. In Proceedings of ICLR, 2015.

[2] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning. In ASIACCS, 2017.

[3] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel. Adversarial examples for malware detection. In ESORICS, 2017.

[4] Rosenberg, I., et al., Generic black-box end-to-end attack against state of the art API call based malware classifiers, in RAID. 2018, Springer. p. 490--510.

谢谢！

大成若缺，其用不弊。大盈若冲，其用不穷。大直若屈。大巧若拙。大辩若讷。静胜躁，寒胜热。清静为天下正。