Stealing Machine Learning Models via Prediction APIs

- 通过对APIs进行多次请求, 从而得到一个能够复制被请求模型的近乎一样能力的模型
- MLaaS : Machine Learning as a Service. 指允许用户上传数据集训练模型, 然后对其他使用这个模型进行请求的人提供服务, 并且收取一定的费用
- Cloud-based ML services通常允许模型的拥有者提供服务, 这就导致了一个问题 : 对模型的请求是widely accessible, 但模型本身和训练数据可能是有隐私的
- 本文考虑一个攻击者能够请求一个API, 并且获得对应于输入的预测. 在这个过程中, 这个模型是被视作一个黑盒, 并且攻击者可能不是知道模型的类型(比如logistic regression/ decision tree/ neural network等), 也不知道训练数据的分布. 攻击者的目标是抽取一个相等或近似相等的模型
- 本文实验表明, 对于广泛的ML模型, 本文方法可以取得和被攻击模型非常相似的模型. 甚至在某些情况下, 本文攻击方法能够抽取出被攻击模型的原本参数(比如linear classifier的coefficients). 对于攻击者不知道模型类型, 参数和特征的情况, 本文也使用了reverse-engineering的方法先来获取这些模型的特征.
- 类似与Google Amazon Microsoft BigML这些MLaaS都是返回最高的confidence和对应的类别标签.
- 对于一个d维的输入, 攻击者能够大概率解决d+1次参数W和b

Service	Model Type	Data set	Queries	Time (s)
Amazon	Logistic Regression	Digits	650	70
	Logistic Regression	Adult	1,485	149
BigML	Decision Tree	German Credit	1,150	631
	Decision Tree	Steak Survey	4,013	2,088

Table 1: **Results of model extraction attacks on ML services.** For each target model, we report the number of prediction queries made to the ML API in an attack that extracts a 100% equivalent model. The attack time is primarily influenced by the service's prediction latency ($\approx 100\text{ms/query}$ for Amazon and $\approx 500\text{ms/query}$ for BigML).

- 本文还讨论了一种简单的方法去防止模型抽取攻击, 即忽略confidence, 只输出模型的标签

Model Extraction Attacks

攻击者的目标是使用尽可能少的queries, 去获得近似的$\hat{f}$对于原本被攻击模型的f. 本文中近似的概念被两个指标定义:

(1) $Test\ error\ R_{test}$: 在测试集上的平均错误率, 越小意味着$\hat{f}$和f在输入分布上更加匹配

$$R_{\text{test}}(f, \hat{f}) = \sum_{(\mathbf{x},y)\in D} d(f(\mathbf{x}), \hat{f}(\mathbf{x}))/|D|.$$

(2)$Uniform\ error\ R_{unif}$: 对于统一从X中选择的数据集U, 计算它的平均错误率. 因此这个指标可以评估整体的错误

$$R_{\text{unif}}(f,\hat{f}) = \sum_{\mathbf{x}\in U} d(f(\mathbf{x}),\hat{f}(\mathbf{x}))/|U|.$$

本文中定义extraction accuracy为 [1-$R_{test}(f,\hat{f})$] + [1-$R_{unif}(f,\hat{f})$]

Equation-Solving Attacks for Regression Type (NN/LR)

绝大多数API是通过接受一个直接请求, 并计算类别概率, 最后返回相关的类别和概率值. 再这样的情况中, 攻击者可以把样本(x, f(x))视为一个未知模型参数的equation并去求解.

以LR为例, 给出一个样本(x, f(x)), 有如下的等式 $\mathbf{w}\cdot\mathbf{x}+\beta = \sigma^{-1}(f_1(\mathbf{x}))$.

因此d+1个样本是充要条件去恢复ω和β

Conclusion

这篇文章没看出什么来, 提出了一种equation-solving进行模型抽取的方法, 可以对NN/LR等回归类模型进行抽取, 而且几乎能够完全复制模型. 但问题在于是通过计算求解? 全文不像一个传统AI风格的文章, 看起来挺吃力. 个人期望使用一个模型的输入和API的输出做数据集进行训练得到一个imitate模型, 也许需要看看teacher-student的东西. 同时本文还讨论了在抽取的模型中进行model inversion的问题, 他们的实验也表明了可以获得原始数据的特征

一句话总结, 部分证明了retrospection的可行性, 但本文的方法好像帮助不大