Testo

M.I.A e I.I.A

I modelli di machine learning possono presentare dei rischi per la privacy nel caso in cui modelli facciano trasparire delle informazioni sensibili sui dati di training attraverso il loro output.
Si parla di Membership Inference Attack se è possibile con una certa accuratezza determinare se un certo dato fosse presente nei dati di training di un modello.
Si parla invece di Identity Inference Attack se è possibile determinare se nel training dataset fosse presente qualche dato corrispondente ad una certa persona avendo a disposizione un altro dato della stessa persona (ad esempio a partire da una foto di un individuo determinare se nel training dataset fosse presente un'altra foto dello stesso individuo).

Black Box e White Box

Per gli inference attacks possiamo usare due modelli diversi.

- White box (scatola bianca) ovvero l'attaccante conosce informazioni sul modello da attaccare come, tipo di modello utilizzato, numero di layer, parametri.
- Black box (scatola nera) ovvero scatola nera ovvero l'attaccante non ha alcuna informazione sul funzionamento interno del modello e può solo utilizzarlo osservando gli output corrispondenti agli input che gli fornisce.
  Il modello black box può essere visto come una API a cui è possibile fare delle richieste con degli input scelti dall'attaccante e ottenere le risposte da utilizzare per l'attacco.
  Ci concentreremo su questo secondo modello in quanto è più simile ad un caso reale in cui un attaccante effettui un M.I.A. su un modello a cui non ha accesso direttamente.

M.I.A. su classificatori

Detto $\mathcal{T}$ il modello target su cui vogliamo svolgere l'attacco.
Detto $\mathcal{D}_{train,\mathcal{T}}$ il dataset di training di $\mathcal{T}$.

Nei modelli di machine learning di classificazione, il modello determina a quale tra k classi è più probabile appartenga l'input.
Il classificatore da in output un vettore lungo k dove ogni componente rappresenta la probabilità che l'input appartenga alla corrispondente classe. Ad esempio

$$\begin{bmatrix} cane \\ gatto \\ orso \\ volpe \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.1 \\ 0.1 \\ 0.2 \end{bmatrix}$$

L'intuizione su cui ci basiamo è che il modello classificherà in maniera diversa input che erano già presenti nei dati di training ($\mathcal{D}_{train,\mathcal{T}}$), ovvero con una confidenza maggiore ad esempio:

$$\begin{bmatrix} cane \\ gatto \\ orso \\ volpe \end{bmatrix} = \begin{bmatrix} 0.9 \\ 0.025 \\ 0.025 \\ 0.05 \end{bmatrix}$$

Come mostra nel Paper di REZA(aggiungi link TODO) l'overfitting del modello da attaccare $\mathcal{T}$ rende maggiori le differenze nella confidenza della classificazione tra dati nuovi e dati già visti dal modello durante il training.

Possiamo quindi addestrare un nuovo modello di machine learning $M_{inference}$ (un classificatore binario)

che a partire da queste differenze nell'output tra dati in $\mathcal{D}_{train,\mathcal{T}}$ e in $\overline{\mathcal{D}_{train,\mathcal{T}}}$
determini se l'input $\in \mathcal{D}_{train,\mathcal{T}}$ o no.

Per poter addestrare $M_{inference}$ avremmo bisogno dei vettori di predizione (e.g. $\begin{bmatrix} cane \\ gatto \\ orso \\ volpe \end{bmatrix}$) con la corrispondente label `in` o `out` in base all'appartenenza a $\mathcal{D}_{train,\mathcal{T}}$.

Non abbiamo a disposizione questi dati per il modello $\mathcal{T}$ quindi creiamo una serie di "shadow models" $\mathcal{S}_i$ che andranno a imitare $\mathcal{T}$.
Siccome questi $\mathcal{S}_i$ sono creati da noi possiamo controllarne il training set $\mathcal{D}_{train,\mathcal{S}_i}$(creazione dataset shadow) e quindi abbiamo a disposizione dei vettori di predizione con la corrispondente label `in` e `out`.
Possiamo quindi addestrare il classificatore binario $M_{inference}$ in modo che capisca se un certo input appartenga a $\mathcal{D}_{train,\mathcal{S}_i}$ oppure no in base al vettore di predizione corrispondente.
L'idea è che se gli shadow models $\mathcal{S}_i$

---

creazione dataset shadow

come creiamo i dataset shadow TODO se serve