IA lingua lunga: Un'Indagine Sull'Overfitting e le Rivelazioni dei Dati

Francesco Lupo, Alessio Di Rubbo, and Michele Daniele

Department of Biosciences and Territory, University of Molise, Italy
{f.lupo, a.dirubbo, m.daniele2}@studenti.unimol.it

Abstract. L'avvento dell'intelligenza artificiale (IA) e del machine learning (ML) ha inaugurato un'era di trasformazioni radicali in settori cruciali come la medicina personalizzata, la finanza predittiva e la sicurezza nazionale. Tuttavia, la crescente dipendenza da enormi volumi di dati sensibili per l'addestramento di questi modelli introduce un paradosso fondamentale: maggiore è l'accuratezza del modello, maggiore è il rischio che esso possa involontariamente rivelare informazioni private.
Questo documento esplora le vulnerabilità introdotte dagli attacchi di inferenza, distinguendo tra Attacchi di Inferenza di Appartenenza (MIA), volti a determinare la presenza di un dato nel set di addestramento, e Attacchi di Inferenza di Attributi (AIA), mirati a estrarre informazioni sensibili mancanti. La presente relazione si propone di illustrare e confrontare le scoperte chiave delineate nei paper "On the (In)Feasibility of Attribute Inference Attacks on Machine Learning Models" e "Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting".
I risultati indicano che i modelli overfittati sono significativamente più vulnerabili ai MIA, memorizzando dettagli specifici del training set. Per gli AIA, la relazione con l'overfitting è più sfumata e dipende dalla specificità dell'attributo e del contesto. La ricerca sottolinea l'urgenza di adottare principi di "privacy by design" nello sviluppo del ML, garantendo che i sistemi di IA siano non solo potenti ed efficaci, ma anche rigorosamente custodi della privacy dei dati.

Keywords: Artificial Intelligence, Machine Learning, Data Privacy, Membership Inference Attacks, Attribute Inference Attacks, Overfitting, Cybersecurity, Networking Security, Model Vulnerabilities, Privacy Protection

1 Introduzione: L'Ombra sui Dati nell'Era dell'Intelligenza Artificiale

L'alba dell'intelligenza artificiale (IA) ha inaugurato un'era di trasformazioni senza precedenti. Modelli di machine learning (ML), alimentati da volumi colossali di dati, stanno ridefinendo settori cruciali quali la medicina personalizzata, la finanza predittiva, i sistemi di giustizia e la sicurezza nazionale. La loro capacità di discernere pattern complessi, automatizzare decisioni e generare intuizioni inimmaginabili fino a pochi anni fa, li rende strumenti indispensabili

nel panorama tecnologico moderno. Tuttavia, questa potenza computazionale nasconde una delicata vulnerabilità: la crescente tensione tra l'eccezionale utilità di questi modelli e la stringente necessità etica e legale di salvaguardare la privacy dei dati sensibili su cui sono addestrati. Un paradosso si manifesta: **più un modello diventa "intelligente" e accurato grazie all'esposizione a dati dettagliati e personali, maggiore è il rischio che possa, involontariamente o meno, rivelare proprio quelle informazioni che dovrebbe proteggere.**

Se in passato la sicurezza delle informazioni si concentrava prevalentemente sulla fortificazione delle "casseforti" digitali – database e canali di trasmissione protetti da robuste crittografie – l'avvento dell'intelligenza artificiale ha radicalmente spostato il campo di battaglia. Oggi, la minaccia non si annida più solo nei luoghi di conservazione, ma si proietta sui modelli stessi che apprendono da quei dati. Immaginate un sistema di apprendimento automatico non come un mero strumento, ma come un raffinato testimone silenzioso: anche quando opera come una "black box", accessibile solo tramite le sue previsioni API apparentemente innocue, esso può inavvertitamente tramutarsi in un'insidiosa porta aperta. Un aggressore astuto non ha bisogno di irrompere nel database originale; è sufficiente decodificare i sottili sussurri dell'algoritmo, leggendo tra le righe delle sue risposte, per inferire non solo tendenze aggregate, ma anche dettagli specifici e riservati inerenti alle informazioni di addestramento. Questa profonda metamorfosi nel panorama delle minacce riscrive le regole del gioco e impone un imperativo chiaro: la "privacy by design" non è più un'opzione, ma un principio fondante da intrecciare in ogni singola fibra del ciclo di vita del ML, dall'acquisizione meticolosa dei dati alla loro sicura messa in opera e al costante affinamento del modello.

In questo contesto dinamico della sicurezza del ML, gli attacchi di inferenza emergono come una delle minacce più significative e contemporanee alla privacy degli individui. Due categorie principali dominano il panorama della ricerca e della preoccupazione pratica: gli Attacchi di Inferenza di Appartenenza (**Membership Inference Attacks - MIA**) e gli Attacchi di Inferenza di Attributi (**Attribute Inference Attacks - AIA**).

I MIA consistono nella determinazione se un dato record è stato incluso nel set di addestramento di un modello di machine learning, dato l'accesso al modello e al record in questione [11]. Questi attacchi sfruttano le minime variazioni nel comportamento del modello, quali la confidenza delle predizioni, i valori di perdita residui o le sfumature interne dei gradienti, al fine di discernere la "memoria" del modello tra i dati osservati durante l'addestramento e quelli a esso sconosciuti. La gravità di tale intrusione si manifesta in contesti sensibili: ad esempio, un modello diagnostico addestrato su cartelle cliniche private, se vulnerabile a un MIA, potrebbe rivelare la partecipazione di un individuo a uno studio su una malattia rara o la presenza del suo profilo medico dettagliato in un database specifico. Tali rivelazioni possono comportare implicazioni significative sul piano sociale, professionale e personale, estendendosi oltre la mera conferma

di un fatto e potendo impattare sulla reputazione, sull'accesso a servizi o sulla sicurezza individuale.

Parallelamente, gli Attacchi di Inferenza di Attributi (Attribute Inference Attacks - AIA) si concentrano sull'estrazione di informazioni specifiche e altamente riservate. Qui, un avversario, in possesso di conoscenze parziali su un record di destinazione (ad esempio, età e sesso di un individuo), sfrutta l'accesso al modello addestrato per dedurre attributi mancanti o sensibili (come la condizione medica preesistente, le preferenze politiche o l'orientamento sessuale) [15]. Per esempio, un AIA potrebbe essere impiegato per inferire il reddito di un individuo basandosi sulle sue abitudini di spesa note e sulle risposte di un modello finanziario, o per dedurre l'affiliazione etnica da un dataset demografico. La natura dell'output del modello gioca un ruolo cruciale: la ricerca ha dimostrato che i modelli di regressione, producendo output continui e più granulari, sono intrinsecamente più vulnerabili agli AIA esatti rispetto ai modelli di classificazione, le cui risposte discrete offrono meno segnali per l'inferenza [13]. Tali attacchi possono compromettere gravemente l'anonimato e la riservatezza degli individui, esponendoli a rischi di profilazione indesiderata, discriminazione sistemica, e persino furto d'identità o estorsione. Ma sarà veramente così? Lo scopriremo nelle prossime sezioni.

La pervasività e l'attualità di queste minacce non sono mere speculazioni teoriche, ma sono state ampiamente riconosciute e dimostrate con forza nella letteratura scientifica. Numerosi studi, hanno rivelato la sorprendente vulnerabilità dei modelli moderni, anche quelli basati su architetture complesse come le reti neurali profonde. Sorprendentemente, questa suscettibilità persiste anche quando i modelli sono addestrati con tecniche di regolarizzazione standard (pensate per prevenire l'overfitting) o su set di dati apparentemente anonimizzati, suggerendo che le misure di privacy tradizionali sono spesso insufficienti. In particolare, il fenomeno dell'overfitting – dove il modello "memorizza" troppo i dati di addestramento invece di generalizzare – è stato identificato come un fattore significativo che aggrava la vulnerabilità ai MIA e, in determinate condizioni, anche agli AIA [13]. Questa lacuna tra la percezione di sicurezza e la realtà operativa dei modelli ML rappresenta una sfida pressante.

L'impatto di tali vulnerabilità trascende la mera sfera accademica, proiettandosi con prepotenza nel tessuto normativo, sociale ed etico. Esse non si limitano a costituire una minaccia latente, ma rappresentano un rischio tangibile per la conformità a legislazioni sempre più stringenti sulla protezione dei dati, come il Regolamento Generale sulla Protezione dei Dati (GDPR) in Europa. Ciò espone le organizzazioni non solo a onerose sanzioni pecuniarie, ma anche a un devastante danno reputazionale, erodendo la fiducia del pubblico in tecnologie che promettono progresso ma rischiano di tradire la riservatezza. Tale erosione di fiducia non è un mero inconveniente, bensì un ostacolo critico alla loro adozione responsabile ed etica in applicazioni sensibili, dove l'accettazione sociale è fondamentale. Pertanto, la comprensione profonda e articolata dei meccanismi sottostanti a MIA e AIA, l'identificazione meticolosa delle condizioni che rendono i modelli suscettibili a tali attacchi, e lo sviluppo proattivo di contromisure efficaci

non sono più confinati all'ambito della ricerca accademica. Essi si configurano piuttosto come imperativi categorici per la progettazione di sistemi di intelligenza artificiale che siano intrinsecamente robusti, incondizionatamente affidabili e, soprattutto, custodi intransigenti della privacy. Questo articolo si propone di illuminare queste aree critiche, tracciando un quadro esaustivo del panorama attuale degli attacchi di inferenza e delineando le sfide che ci attendono nella costruzione di un futuro dell'IA che sia tanto potente quanto inviolabilmente privato.

Tutti i codici sorgente, i notebook eseguibili e i dataset utilizzati per la sperimentazione descritta nel presente elaborato sono disponibili pubblicamente nella seguente repository GitHub:
https://github.com/MicheleDaniele/Privacy-LLM-Networking_Security.
La repository contiene anche istruzioni dettagliate per l'esecuzione su Google Colab.

2 Background

2.1 Paradigmi dei Modelli di Machine Learning

I modelli di machine learning si differenziano primariamente in base alla natura del problema che sono progettati per risolvere e alla tipologia di output che generano. I due paradigmi fondamentali in questo ambito sono la Classificazione e la Regressione.

- **Regressione**: Contrariamente ai modelli di classificazione, i modelli di regressione sono finalizzati alla previsione di valori numerici continui basati su variabili di input. L'output non è una categoria, bensì un valore numerico all'interno di un intervallo specificato. Applicazioni comuni includono la previsione dei prezzi di mercato, la stima di parametri ambientali o la determinazione di dosaggi precisi.
- **Classificazione**: I modelli di classificazione sono concepiti per assegnare etichette discrete o categorie a dati di input. L'obiettivo è determinare la relazione di un'istanza a una classe predefinita. Esempi tipici includono la categorizzazione di immagini (e.g., identificazione di "cane" o "gatto"), la rilevazione di messaggi di posta elettronica indesiderati (spam/non spam) o la diagnosi medica (presenza/assenza di una patologia). L'output di tali modelli è generalmente una probabilità o un punteggio di confidenza per ciascuna classe, culminante in una decisione discreta. La natura categorica di questi output può intrinsecamente limitare la granularità delle informazioni estraibili da un attaccante, rendendo gli attacchi di inferenza di attributi esatti (AIA) più complessi rispetto ai modelli di regressione.

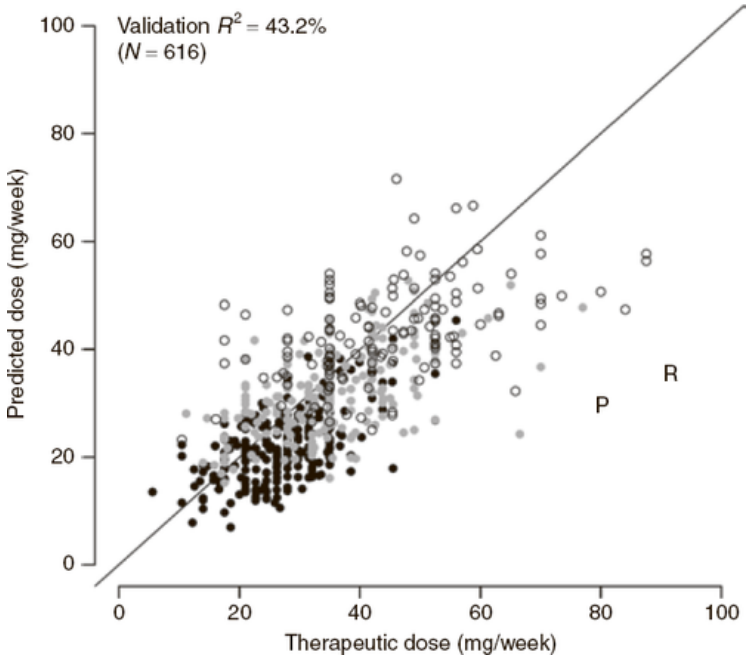


Fig. 1. Esempio di immagine di regressione. Fonte: https://www.datacamp.com/blog/classification-machine-learning

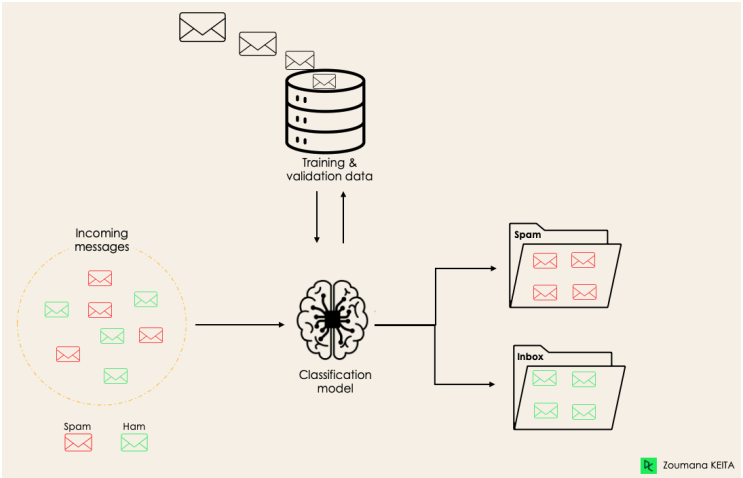


Fig. 2. Esempio di immagine di classificazione. Fonte: https://www.datacamp.com/blog/classification-machine-learning

2.2 Generalizzazione, Overfitting e Stabilità del Modello

La performance di un modello di machine learning su dati non precedentemente osservati è un indicatore cruciale della sua efficacia e affidabilità. Questo concetto è intrinsecamente connesso ai principi di Generalizzazione, Overfitting e Stabilità del Modello.

- **Generalizzazione**: La generalizzazione rappresenta la capacità di un modello di machine learning di applicare con successo le conoscenze e i pattern acquisiti dal suo "set di addestramento" a dati "inediti", mai incontrati prima, mantenendo un'elevata accuratezza predittiva. Un modello ben generalizzato non si limita a memorizzare gli esempi specifici presenti nel set di addestramento, ma piuttosto apprende i "pattern" e le "relazioni intrinseche" sottostanti ai dati, rendendoli applicabili a nuove situazioni. L'obiettivo primario di qualsiasi algoritmo di machine learning è ottimizzare questa capacità di generalizzazione, affinché il modello sia efficacemente utile nel mondo reale.
  Per valutare l'efficacia di un modello nella generalizzazione, si impiega **l'errore di generalizzazione**. Questo può essere inteso come la differenza tra le prestazioni del modello sui dati di addestramento e quelle sui dati inediti (o di test). Una differenza contenuta indica una robusta capacità di generalizzazione e un apprendimento efficace del modello.
- **Overfitting**: L'overfitting si manifesta quando un modello di machine learning si adatta in modo eccessivo ai dati di addestramento. Invece di cogliere i pattern generalizzabili, il modello tende a memorizzare dettagli specifici e persino il "rumore" presente nel set di addestramento. Di conseguenza, pur mostrando prestazioni eccezionali sui dati già visti, la sua performance si degrada significativamente quando incontra dati nuovi o leggermente diversi. Questo fenomeno costituisce un rilevante indicatore di rischio per la privacy, poiché implica che il modello ha "memorizzato" informazioni sensibili relative ai singoli punti dati di addestramento, rendendoli potenzialmente inferibili.
- **Stabilità del Modello**: La stabilità del modello è un attributo fondamentale che descrive quanto un modello di machine learning sia immune a variazioni significative nel suo comportamento o nelle sue previsioni, anche quando il set di dati di addestramento subisce piccole modifiche. Per analogia, si consideri un algoritmo crittografico robusto: la sua stabilità è dimostrata dalla sua capacità di mantenere la sicurezza e l'integrità delle operazioni (ad esempio, la crittografia o la generazione di hash) anche se i dati di input o i parametri subiscono alterazioni minime. Un sistema crittografico stabile non collassa o rivela informazioni sensibili a causa di piccole, sebbene non autorizzate, modifiche nel contesto operativo o nei dati trattati, garantendo che le sue proprietà fondamentali persistano. Più specificamente, la stabilità **On-Average-Replace-One (ARO)** serve a quantificare questa resistenza. Essa misura, in media, di quanto cambia l'output del modello nel momento in cui un singolo punto dato viene rimosso o sostituito all'interno del set di addestramento. Un algoritmo con un'elevata stabilità ARO è meno

propenso a "memorizzare" i dettagli fini di singoli dati, poiché l'influenza di ciascun dato sul comportamento complessivo del modello è limitata. Se una modifica minima a un singolo dato produce solo una variazione contenuta nell'output del modello, ciò implica che quel dato esercita un'influenza ridotta e non dominante, rendendo così più ardua la rivelazione di informazioni private ad esso associate. Questa intrinseca stabilità è strettamente correlata alla Privacy Differenziale [3]. La **Privacy Differenziale** è un concetto chiave che è possibile comprendere pensando a un insieme di dati, ad esempio, un database di utenti. L'idea è che, se togliamo o aggiungiamo il record di un singolo utente a questo database, il risultato di un'analisi o il comportamento di un modello di machine learning addestrato su quel database dovrebbe rimanere quasi identico. In altre parole, la Privacy Differenziale è un quadro matematicamente rigoroso che garantisce che la presenza o l'assenza del dato di una persona non alteri in modo significativo l'output finale del sistema. Questo rende estremamente difficile per un osservatore esterno capire se il dato di quella persona specifica sia stato utilizzato o meno. Praticamente, un modello che rispetta la Privacy Differenziale ha un alto grado di stabilità perché è progettato per essere insensibile ai singoli dati: esercita una minore "memoria" su di essi e, di conseguenza, offre un livello superiore di protezione della privacy. Il documento sottolinea che la stabilità, rispetto al solo errore di generalizzazione, fornisce un'indicazione più diretta e affidabile di quanto il modello "memorizzi" i singoli dati. (Yeom et al.[13]) ha inoltre dimostrato che alcuni algoritmi, pur essendo considerati stabili (e quindi in grado di generalizzare efficacemente, cioè non overfittati), possono comunque essere manipolati per consentire la fuga di informazioni precise. Questo mette in evidenza come la stabilità sia un fattore critico, e talvolta sottovalutato, per la salvaguardia della privacy nel machine learning.

2.3 L'Overfitting: Un Indicatore di Rischio per la Privacy, ma Non l'Unico Fattore Determinante

Sebbene l'overfitting sia spesso considerato un indicatore primario del rischio per la privacy, associato alla "memorizzazione" dei dati di addestramento, è cruciale riconoscere che non costituisce l'unico fattore determinante della vulnerabilità. Affidarsi esclusivamente all'overfitting può indurre un falso senso di sicurezza, poiché le limitazioni alla privacy derivano anche da altri aspetti fondamentali.

3 Attacks

La presente sezione si propone di esaminare in dettaglio gli attacchi di inferenza di appartenenza (Membership Inference Attacks - MIA) e gli attacchi di inferenza di attributi (Attribute Inference Attacks - AIA), analizzando le metodologie e le conclusioni presentate nei lavori di Yeom et al. e Zhao et al.

3.1 Attacchi di Inferenza di Appartenenza (Membership Inference Attacks - MIA)

Il concetto di attacco di inferenza di appartenenza può essere concettualizzato come l'assunzione del ruolo di un investigatore digitale, la cui missione consiste nel discernere se un'entità specifica, rappresentata da un "punto dati", sia stata inclusa in un "archivio riservato" — ovvero il dataset di training — impiegato per l'addestramento di un modello di intelligenza artificiale. Questa rappresenta l'essenza concettuale di un attacco di inferenza di appartenenza.

Nel contesto di un attacco di inferenza di appartenenza, l'avversario mira a determinare la presenza o l'assenza di un punto dati specifico all'interno del dataset utilizzato per l'addestramento di un determinato modello. All'investigatore vengono forniti i seguenti elementi chiave:

- Il punto dati oggetto di indagine $z = (x, y)$: Questo elemento costituisce l'indizio primario, la cui appartenenza al database di training deve essere accertata.
- L'accesso al modello (A_S): Il modello opera come un **oracolo** che fornisce risposte. Questa interazione può essere paragonata all'interrogazione di una funzione crittografica che elabora input e produce output.
- La dimensione del set di training ($S = n$): Questo parametro indica il numero totale di entità contenute nell'"archivio riservato" di addestramento.
- La distribuzione da cui è stato estratto il training set (D): Rappresenta la "popolazione" generale da cui provengono tutti i possibili individui.

Munito di tali informazioni, l'investigatore è chiamato a formulare un giudizio conclusivo: il punto dati z è stato utilizzato nel set di training ($z \in S$) oppure no?

3.2 La Prospettiva di Yeom et al.

Yeom et al. [13] approfondiscono significativamente la correlazione tra l'overfitting di un modello e la sua vulnerabilità agli attacchi MIA. Il loro lavoro non si limita a identificare questa connessione, ma la formalizza ed esplora attraverso diverse strategie d'attacco. Per formalizzare questa "procedura investigativa", viene proposto l'Esperimento 1 (Membership experiment $Exp^M(\mathcal{A}, A, n, D)$. Questo esperimento costituisce una procedura di test rigorosa, paragonabile a una "sfida distinguibile" in crittografia, ove un avversario tenta di discriminare tra due scenari probabilistici:

1. Preparazione del "Covo Segreto": Un nuovo dataset S viene campionato dalla distribuzione D, e su di esso viene addestrato il modello $A_S = A(S)$. Questa fase simula la creazione di un sistema basato su dati potenzialmente sensibili.
2. Un valore binario b (0 o 1) viene scelto uniformemente a caso. Questo b rappresenta il "segreto" che l'investigatore è chiamato a scoprire, analogamente a una chiave binaria nascosta in un protocollo crittografico.

3. L'Indizio Fasullo o Reale:
   - Se $b = 1$, il punto dati z viene campionato dalla "popolazione" generale D. Qui, z potrebbe essere un membro o meno, senza alcuna garanzia di appartenenza al training set.
   - Se $b = 0$, il punto dati z viene campionato direttamente dal "covo segreto" S. In questo caso, z è indubbiamente un membro del training
4. La Sfida dell'Investigatore: L'avversario $\mathcal{A}$ riceve il punto dati z, l'accesso al modello A_s, la dimensione n del training set e la distribuzione D. Basandosi su queste informazioni, $\mathcal{A}$ deve annunciare la sua ipotesi b' (0 o 1) sul valore di b. L'esperimento $Exp^M(\mathcal{A}, A, n, D)$ restituisce 1 se l'investigatore indovina correttamente il valore di b, altrimenti restituisce 0.

L'avversario non possiede un vantaggio a priori sulla probabilità che z sia in S o in D basandosi unicamente sulla sua distribuzione, ma deve sfruttare le "fughe" di informazione dal modello A_S. In termini crittografici, è come se il modello fosse una funzione hash parzialmente "compromessa", dove l'output, pur non rivelando direttamente l'input, consente di inferire una sua proprietà nascosta.

Definizione 4: Misurare il "Vantaggio di Distinzione" Il vantaggio di distinzione ($Adv^M(\mathcal{A}, A, n, D)$) è definito come:

$$Adv^M(\mathcal{A}, A, n, D) = 2Pr[Exp^M(\mathcal{A}, A, n, D) = 1] - 1$$

Questa formula può essere interpretata anche come la differenza tra il "vero positivo" e il "falso positivo" dell'attaccante. Più specificamente, è equivalente a:

$$\text{Adv}^M = \Pr[\mathcal{A} \text{ predice } 0|b = 0] - \Pr[\mathcal{A} \text{ predice } 0|b = 1]$$

dove:

- $\Pr[\mathcal{A} \text{ predice } 0|b = 0]$ è la probabilità che l'attaccante indovini correttamente che z proviene dal training set ($b = 0$). Questo è il tasso di veri positivi (True Positive Rate - TPR).
- $\Pr[\mathcal{A} \text{ predice } 0|b = 1]$ è la probabilità che l'attaccante sbagli, classificando erroneamente un dato dalla distribuzione generale come proveniente dal training set ($b = 1$). Questo è il tasso di falsi positivi (False Positive Rate - FPR).

In questo contesto, il vantaggio Adv^M (una pratica scorciatoia per $\text{Adv}^M(\mathcal{A}, A, n, D)$) è la misura chiave che quantifica l'abilità di un attaccante nel distinguere tra un punto dati originario dal training set ($z \sim S$) e un punto dati proveniente dalla distribuzione generale ($z \sim D$), avendo a disposizione il modello.

Questa impostazione differisce sottilmente ma significativamente da quella di altri lavori precedenti [7], [11], i quali potrebbero distinguere tra un dato del training set e un dato che è *garantito* non essere nel training set ($z \sim D \setminus S$). La ragione di tale scelta da parte di Yeom et al. è fondamentale: il loro interesse primario consiste nel misurare in che misura il modello A_S stesso rivela l'appartenenza, piuttosto che quanto l'attaccante possa inferire da una conoscenza preesistente di S o D.

Se l'esperimento prevedesse il campionamento di z da $D \setminus S$ invece che da D, l'avversario potrebbe ottenere un vantaggio sfruttando semplicemente la conoscenza di quali punti dati sono più probabili essere stati campionati in un set S estratto da D^n. Questo tipo di vantaggio non rifletterebbe la "permeabilità" intrinseca del modello, e la Definizione 4 è stata formulata proprio per escluderlo.

Ciò implica che, nell'esperimento di appartenenza $\text{Exp}^M(\mathcal{A}, A, n, D)$, l'unico modo per l'avversario $\mathcal{A}$ di guadagnare un vantaggio significativo è attraverso l'interazione con il modello A_S. Gli altri input, n (la dimensione del training set) e D (la distribuzione generale), non dipendono dal valore di b (se il dato è membro o meno). Per chiarire ulteriormente questo punto, la seguente relazione probabilistica descrive la situazione prima che il modello venga interrogato:

$$\Pr[b = 0|z] = \Pr_{S\sim D^n}[z] \cdot \frac{\Pr[b = 0]}{\Pr[z]} = \Pr_{z\sim D}[z] \cdot \frac{\Pr[b = 1]}{\Pr[z]} = \Pr[b = 1|z].$$

Questa formula può essere compresa nel seguente modo: la probabilità condizionata $\Pr[b = 0|z]$ rappresenta la probabilità che il punto dati z provenga dal training set (corrispondente a $b = 0$), dato il punto z stesso. Similmente, $\Pr[b = 1|z]$ è la probabilità che z provenga dalla distribuzione generale D (corrispondente a $b = 1$), sempre dato z. La formula afferma che, se non si ha accesso al modello A_S, e ci si basa unicamente sul punto dati z, **la probabilità che z sia un membro del training set è identica alla probabilità che z provenga dalla distribuzione generale**. In altre parole, z di per sé non fornisce alcun indizio sulla sua appartenenza prima dell'interrogazione del modello. Qualsiasi successo dell'attaccante deve quindi derivare dal comportamento del modello.

È importante notare che la Definizione 4 non accredita all'avversario la capacità di predire correttamente che un punto campionato da D (cioè, quando $b = 1$), che per pura coincidenza si trova anche in S, sia un membro di S. Di conseguenza, il vantaggio massimo che un avversario può sperare di raggiungere è $1 - \mu(n, D)$, dove $\mu(n, D) = \Pr_{S\sim D^n, z\sim D}[z \in S]$ è la probabilità di ricampionare un individuo dal training set nella popolazione generale. In contesti reali con dataset di grandi dimensioni, $\mu(n, D)$ è probabilmente estremamente piccolo, rendendo questa limitazione trascurabile nella pratica.

Un valore di Adv^M pari a 0 significa che l'attaccante non ha alcun vantaggio rispetto a un lancio di moneta casuale. Un valore vicino a 1 indica un attacco altamente efficace, in cui l'attaccante può quasi sempre determinare correttamente l'appartenenza al training set.

Il lavoro di Yeom et al. sottolinea che un vantaggio significativo in un attacco MIA è spesso dovuto al fatto che i non-membri del training set che l'attaccante incontra sono "lontani" dai membri. In altre parole, c'è una chiara differenza nel comportamento del modello tra un dato che ha visto durante l'addestramento e un dato che non ha visto e che è significativamente diverso da quelli visti.

Tecniche di Attacco e Fattori di Rischio per MIA Le analisi precedenti hanno stabilito che l'overfitting è una condizione sufficiente per la vulnerabilità agli attacchi di inferenza di appartenenza. Questa sezione approfondisce le

diverse tecniche di attacco MIA e i fattori che ne influenzano l'efficacia, dimostrando come il comportamento di generalizzazione del modello sia un forte predittore di tale vulnerabilità.

Limiti dalla Privacy Differenziale Un aspetto fondamentale affrontato da Yeom et al. è il legame tra la privacy differenziale e la limitazione degli attacchi MIA. La privacy differenziale [**?**] è un robusto standard crittografico per la protezione della privacy che impone vincoli rigorosi su quanto la presenza o l'assenza di un singolo punto dati nel training set possa influenzare l'output di un algoritmo. È ampiamente riconosciuto che i modelli addestrati con garanzie di privacy differenziale dovrebbero, per loro natura, limitare gli attacchi di inferenza di appartenenza.

Teorema 1: Sia A un algoritmo di apprendimento ϵ-differenzialmente privato e $\mathcal{A}$ un avversario di inferenza di appartenenza. Allora si ha:

$$\mathrm{Adv}^M(\mathcal{A}, A, n, D) \leq e^\epsilon - 1.$$

Dimostrazione: La dimostrazione di questo teorema si fonda sulla definizione cardine della ϵ-privacy differenziale. Tale definizione stabilisce che per qualsiasi coppia di dataset adiacenti S e S' (che differiscono per un solo elemento, ad esempio S' è S con un elemento sostituito, o con un elemento aggiunto/rimosso), e per qualsiasi output o di un algoritmo randomizzato M:

$$\Pr[M(S) = o] \leq e^\epsilon \cdot \Pr[M(S') = o].$$

Nel contesto del Teorema 1, l'algoritmo di apprendimento A è ϵ-differenzialmente privato. Ciò significa che la sua "reazione" alla presenza o assenza di un dato specifico nel training set è controllata. Consideriamo un set di addestramento $S = (z_1, \ldots, z_n)$ campionato da D^n. Definiamo un dataset adiacente $S^{(i)}$ come S in cui il i-esimo punto z_i è stato sostituito con un nuovo punto z' campionato da D. Poiché A è ϵ-differenzialmente privato, la distribuzione del modello A_S (ottenuto addestrando A su S) è "vicina" alla distribuzione del modello $A_{S^{(i)}}$ (ottenuto addestrando A su $S^{(i)}$). Questa vicinanza è quantificata dal parametro ϵ. Nel contesto dell'esperimento di inferenza di appartenenza, l'avversario $\mathcal{A}$ cerca di distinguere tra due scenari (quello in cui z è nel training set e quello in cui non lo è) analizzando il comportamento del modello. Se il modello è differenzialmente privato, il suo comportamento non dovrebbe cambiare in modo significativo anche se un singolo punto dati è presente o assente dal training set. La prova formale mostra che il vantaggio dell'avversario Adv^M è limitato da una funzione di ϵ. Nello specifico, l'espressione $e^\epsilon - 1$ quantifica la massima "distinguibilità" che l'avversario può raggiungere. Quando ϵ è molto piccolo (indicando una privacy elevata), $e^\epsilon - 1$ si avvicina a ϵ, limitando drasticamente il vantaggio dell'attaccante. Ciò significa che il modello non "trapela" abbastanza informazioni su singoli punti dati per permettere un'inferenza efficace della loro appartenenza. In pratica, se il modello non cambia in modo percettibile a seguito della modifica di un singolo dato, l'avversario avrà difficoltà estreme a

distinguere la presenza o meno di quel dato nel training set, preservando così la privacy.

Questo teorema evidenzia come un algoritmo differenzialmente privato, caratterizzato da un piccolo valore di ϵ, possa efficacemente limitare il vantaggio di un attaccante. In combinazione con il Teorema 1, ciò fornisce un limite teorico sul vantaggio di appartenenza quando la funzione di perdita soddisfa tali proprietà.

Attacchi di Appartenenza e Generalizzazione Il comportamento di generalizzazione del modello è un forte predittore della sua vulnerabilità agli attacchi MIA. L'overfitting, in particolare, è una condizione sufficiente per tale vulnerabilità.

Attacchi Basati sulla Funzione di Perdita Limitata: Il primo approccio di attacco MIA considerato è relativamente semplice e si basa sull'assunto che la funzione di perdita del modello sia limitata da una costante B.

Avversario 1 (Funzione di Perdita Limitata): Si supponga che la funzione di perdita $l(A_S, z)$ sia sempre minore o uguale a una costante B per qualsiasi set S e punto z. Dato un punto $z = (x, y)$, il modello A_S, la dimensione n e la distribuzione D, l'avversario $\mathcal{A}$ procede come segue:

1. Interroga il modello per ottenere la previsione $A_S(x)$.
2. Calcola la perdita $l(A_S, z)$. La perdita è una misura di quanto "male" il modello si comporta su quel particolare punto dati.
3. Genera un output 1 (indicando che z non è nel training set) con probabilità $l(A_S, z)/B$. Altrimenti, genera 0 (indicando che z è nel training set). Questa è la regola di decisione dell'avversario: se la perdita è bassa, è più probabile che il dato fosse nel training set (il modello lo "conosce bene" e fa pochi errori); se la perdita è alta, è più probabile che il dato non fosse nel training set.

Teorema 2: Il vantaggio dell'Avversario 1 è $\mathrm{R}_{gen}(A)/B$. *Dimostrazione:* La dimostrazione di questo teorema stabilisce una relazione diretta tra il vantaggio dell'attaccante e l'errore di generalizzazione dell'algoritmo di apprendimento. L'errore di generalizzazione $\mathrm{R}_{gen}(A)$ è definito come la differenza tra la perdita media del modello su un punto dati campionato dalla distribuzione generale ($z \sim D$) e la perdita media del modello su un punto dati campionato dal training set ($z \sim S$):

$$\mathrm{R}_{gen}(A) = E_{S\sim D^n, z\sim D}[l(A_S, z)] - E_{S\sim D^n, z\sim S}[l(A_S, z)].$$

- Il termine $E_{S\sim D^n, z\sim D}[l(A_S, z)]$ rappresenta la **perdita attesa su dati non visti** (o la perdita di test/generalizzazione). Questa è la media della perdita del modello su punti dati presi dalla popolazione generale.
- Il termine $E_{S\sim D^n, z\sim S}[l(A_S, z)]$ rappresenta la **perdita attesa su dati di addestramento** (o la perdita di training). Questa è la media della perdita del modello sui punti dati su cui è stato addestrato.

Quando un modello è in overfitting, la perdita sui dati di addestramento è significativamente inferiore rispetto alla perdita sui dati non visti. Pertanto, un $\mathrm{R}_{gen}(A)$ elevato indica un forte overfitting. Il teorema dimostra che il vantaggio dell'Avversario 1 è esattamente $\mathrm{R}_{gen}(A)/B$. Ciò significa che l'attaccante sfrutta direttamente la discrepanza tra quanto bene il modello "conosce" i dati di addestramento e quanto bene generalizza ai dati nuovi. Se il modello "memorizza" i dati di addestramento (perdita bassa sul training set) ma fa molti errori sui dati nuovi (perdita alta sul test set), questa differenza crea un segnale che l'attaccante può utilizzare per inferire l'appartenenza. In particolare, per le funzioni di perdita 0-1 (comuni nei problemi di classificazione), dove $B = 1$, il vantaggio dell'attaccante è esattamente l'errore di generalizzazione. Questo risultato è cruciale perché stabilisce una connessione formale e quantitativa: **un elevato errore di generalizzazione (cioè, un forte overfitting) si traduce inevitabilmente in una perdita di privacy per i modelli di classificazione, rendendoli vulnerabili agli attacchi MIA.**

Errore Gaussiano: Quando l'avversario ha una conoscenza precisa della distribuzione degli errori del modello, può adottare una strategia più sofisticata. I modelli di regressione lineare, ad esempio, assumono implicitamente una distribuzione normale (gaussiana) per gli errori.

Avversario 2 (Soglia): Si supponga che le funzioni di densità di probabilità condizionate dell'errore, $f(\epsilon|b = 0)$ (errore se z è membro) e $f(\epsilon|b = 1)$ (errore se z non è membro), siano note a priori. Dati $z = (x, y)$, A_S, la dimensione n e la distribuzione D, l'avversario $\mathcal{A}$ procede come segue:

1. Interroga il modello per ottenere la previsione $A_S(x)$.
2. Calcola l'errore $\epsilon = y - A_S(x)$. Questo è l'errore effettivo del modello sul punto z.
3. Restituisce il valore $b \in \{0, 1\}$ che massimizza la probabilità $f(\epsilon|b)$. L'avversario decide se z è un membro o meno scegliendo lo scenario ($b = 0$ o $b = 1$) che rende più probabile l'errore ϵ osservato.

In problemi di regressione che impiegano la funzione di perdita a errore quadratico, la grandezza dell'errore di generalizzazione è influenzata dalla scala della variabile di risposta y. Per tale ragione, si ricorre al rapporto σ_D/σ_S per quantificare l'errore di generalizzazione. Qui, σ_S rappresenta la deviazione standard degli errori del modello sui dati di training (membri), mentre σ_D è la deviazione standard degli errori sui dati campionati dalla popolazione generale (non-membri).

Teorema 3: Si supponga che σ_S e σ_D siano noti a priori, tali che $\epsilon \sim N(0, \sigma_S^2)$ quando $b = 0$ (l'errore su un membro segue una distribuzione normale con varianza σ_S^2) e $\epsilon \sim N(0, \sigma_D^2)$ quando $b = 1$ (l'errore su un non-membro segue una distribuzione normale con varianza σ_D^2). Allora, il vantaggio dell'Avversario 2 di Inferenza di Appartenenza è:

$$\operatorname{erf}\left(\frac{\sigma_D}{\sigma_S}\sqrt{\frac{\ln(\sigma_D/\sigma_S)}{(\sigma_D/\sigma_S)^2 - 1}}\right) - \operatorname{erf}\left(\sqrt{\frac{\ln(\sigma_D/\sigma_S)}{(\sigma_D/\sigma_S)^2 - 1}}\right).$$

Dimostrazione: La dimostrazione approfondisce l'analisi delle funzioni di densità di probabilità (PDF) normali degli errori, $f(\epsilon|b = 0)$ e $f(\epsilon|b = 1)$. Poiché i

membri del training set sono tipicamente "meglio appresi" dal modello rispetto ai non-membri, ci si aspetta che gli errori sui membri siano, in media, più piccoli e meno variabili, il che si riflette in una σ_S inferiore a σ_D. L'avversario sfrutta questa differenza per distinguere l'appartenenza. La strategia ottimale dell'avversario è determinare una "soglia di decisione" ϵ_{eq} (errore di equilibrio) tale che, se l'errore osservato $|\epsilon|$ è inferiore a ϵ_{eq}, l'attaccante predice che il dato è un membro (più probabile che provenga da $b = 0$); altrimenti, predice che non lo è (più probabile che provenga da $b = 1$). Questa soglia ϵ_{eq} è calcolata risolvendo l'equazione $f(\epsilon|b = 0) = f(\epsilon|b = 1)$, ovvero il punto in cui le due distribuzioni di probabilità condizionate si intersecano. L'espressione per ϵ_{eq} è:

$$\epsilon_{eq} = \sigma_D \sqrt{\frac{2\ln(\sigma_D/\sigma_S)}{(\sigma_D/\sigma_S)^2 - 1}}.$$

Il vantaggio dell'attaccante è quindi la differenza tra la probabilità che l'errore di un membro rientri nella regione di decisione ($\Pr[|\epsilon| < \epsilon_{eq}|b = 0]$) e la probabilità che l'errore di un non-membro rientri in quella stessa regione ($\Pr[|\epsilon| < \epsilon_{eq}|b = 1]$). La funzione $\mathrm{erf}(\cdot)$ (funzione errore di Gauss) è utilizzata per calcolare queste probabilità cumulative per le distribuzioni normali. Il teorema dimostra che il vantaggio è nullo quando $\sigma_S = \sigma_D$ (indicando assenza di overfitting, poiché il modello si comporta allo stesso modo su membri e non-membri) e si avvicina a 1 (massimo vantaggio) quando $\sigma_D/\sigma_S \to \infty$ (indicando un overfitting estremo, dove gli errori sui non-membri sono molto più grandi e variabili di quelli sui membri).

Errore Standard Sconosciuto: In pratica, i modelli spesso pubblicano un solo valore per l'errore standard. Questo pone la sfida di non conoscere il rapporto σ_D/σ_S. Una soluzione pratica è assumere che $\sigma_S \approx \sigma_D$, implicando che il modello non sia gravemente in overfitting. In tal caso, la soglia di decisione per l'errore $|\epsilon|$ viene impostata a σ_S. Alternativamente, se l'avversario conosce l'algoritmo di apprendimento utilizzato, può campionare ripetutamente set di addestramento da D, addestrare il modello A_S e misurare l'errore per ottenere stime affidabili di σ_S e σ_D.

Altre Fonti di Vantaggio nell'Inferenza di Appartenenza Le analisi precedenti mostrano che l'overfitting è una condizione sufficiente per l'inferenza di appartenenza. Tuttavia, i modelli possono rivelare informazioni sul training set anche in altri modi, implicando che l'overfitting non è una condizione necessaria per tale vantaggio. Ad esempio, una regola di apprendimento potrebbe semplicemente produrre un modello che funge da codifica lossless del training set. Sebbene tale esempio possa sembrare semplicistico, esistono minacce più sofisticate.

Il concetto di **Algorithm Substitution Attack (ASA)** emerge qui. In un ASA, un algoritmo "legittimo" (ad esempio, un software a sorgente chiuso che addestra modelli) viene segretamente alterato da un fornitore malevolo. Questa alterazione permette a un avversario colluso di violare la privacy degli utenti dell'algoritmo, il tutto in modo quasi impossibile da rilevare.

Teorema 4: Siano $d = \log|X|$ e $m = \log|Y|$, l una funzione di perdita limitata da una costante B, e A una regola di apprendimento ARO-stabile con tasso $\epsilon_{stable}(n)$. Si supponga che x determini univocamente il punto (x, y) in D. Allora, per ogni intero $k > 0$, esiste una regola di apprendimento ARO-stabile A^k con tasso massimo $\epsilon_{stable}(n) + knB2^{-d} + \mu(n, D)$ e un avversario $\mathcal{A}$ tale che:

$$\text{Adv}^M(\mathcal{A}, A^k, n, D) = 1 - \mu(n, D) - 2^{-mk}.$$

Dimostrazione: La prova di questo teorema è particolarmente sofisticata in quanto dimostra la possibilità di compromettere la privacy anche in modelli che non presentano overfitting, attraverso una "collusione" tra l'algoritmo di addestramento e l'avversario. Il cuore della dimostrazione è la costruzione di una regola di apprendimento modificata A^k e di un avversario $\mathcal{A}$ che operano in tandem. Il modello A^k viene progettato per incorporare informazioni sull'appartenenza del training set in un modo "nascosto" ma recuperabile. Questo si ottiene utilizzando un insieme di k coppie di funzioni pseudocasuali con chiave (F_K, G_K). Una **funzione pseudocasuale (PRF)** è un algoritmo deterministico che, a meno di conoscere una chiave segreta, si comporta come una vera funzione casuale. L'algoritmo di addestramento A^k "supplementa" il training set originale S con punti dati generati utilizzando queste PRF e le chiavi segrete. Questi punti supplementari sono costruiti in modo tale che, per un membro (x_i, y_i) del training set, A^k "memorizzi" una relazione specifica tra $F_{K_j}(x_i)$ e $G_{K_j}(x_i)$ per ogni chiave K_j. Il modello A^k è configurato per dare una risposta specifica (cioè $G_{K_j}(x)$) quando interrogato con un input della forma $F_{K_j}(x)$. In tutti gli altri casi, si comporta come il modello originale A. L'avversario $\mathcal{A}$ è a conoscenza delle stesse chiavi $K_1, \ldots, K_k$. Quando vuole testare un punto dati $z = (x, y)$, l'avversario interroga il modello A^k con gli input "speciali" $F_{K_j}(x)$ per ogni chiave K_j. Se le risposte del modello $A^k(F_{K_j}(x))$ corrispondono esattamente a $G_{K_j}(x)$ per tutte le k chiavi, l'avversario deduce che x (e quindi z) era un membro del training set. Se anche una sola corrispondenza fallisce, si conclude che non è un membro. Il termine 2^{-mk} nell'espressione del vantaggio deriva dalla probabilità che l'attaccante ottenga per pura casualità la corrispondenza $y'_j = G_{K_j}(x)$ per tutte le k chiavi, anche se z non fosse un membro. Poiché G_K mappa su uno spazio di 2^m possibili valori e ci sono k tali test, la probabilità di una corrispondenza casuale su tutti i k test è $(2^{-m})^k = 2^{-mk}$. Questo termine diventa **negligibile** (ovvero estremamente piccolo, virtualmente nullo) all'aumentare del numero di chiavi k. Il teorema dimostra che, nonostante A^k sia approssimativamente ARO-stabile (cioè, la sua capacità di generalizzazione è vicina a quella del modello originale A), l'avversario colluso può quasi perfettamente recuperare le informazioni di appartenenza. Questo risultato è cruciale perché rivela che la privacy può essere compromessa anche da algoritmi che non mostrano overfitting (e che quindi sembrerebbero "sicuri" secondo le metriche tradizionali), attraverso meccanismi di fuga di informazione intenzionali e discreti.

A titolo illustrativo, si presentano un algoritmo di addestramento colludente e un avversario colludente che incarnano i principi di tale attacco:

Algoritmo 1 (Algoritmo di Addestramento Colludente A^C): Siano $F_K : X \mapsto X$ e $G_K : X \mapsto Y$ funzioni pseudocasuali con chiave, $K_1, ..., K_k$

chiavi scelte uniformemente, e A un algoritmo di addestramento. Alla ricezione di un training set S, A^C procede come segue:

1. Integra S utilizzando F, G: per ogni $(x_i, y_i) \in S$ e per ogni $j \in [k]$, si definisca $z'_{i,j} = (F_{K_j}(x_i), G_{K_j}(x_i))$, e si ponga $S' = S \cup \{z'_{i,j} | i \in [n], j \in [k]\}$.
2. Restituisce $A_{S'} = A(S')$.

Avversario 3 (Avversario Colludente $\mathcal{A}^C$): Siano $F_K : X \mapsto X$, $G_K : X \mapsto Y$ le funzioni e $K_1, ..., K_k$ le chiavi utilizzate da A^C. Sia $A_{S'}$ il modello risultante dall'addestramento con A^C e tali chiavi. Dato un input $z = (x, y)$, l'avversario $\mathcal{A}^C$ procede come segue:

1. Per ogni $j \in [k]$, si calcoli $y'_j \leftarrow A_{S'}(F_{K_j}(x))$.
2. Restituisce 0 (membro) se $y'_j = G_{K_j}(x)$ per ogni $j \in [k]$. Altrimenti, restituisce 1 (non-membro).

L'Algoritmo 1, sebbene formalmente dimostrato, potrebbe non essere pratico per tutte le classi di modelli, in quanto la loro capacità di memorizzare informazioni di appartenenza e generalizzare è variabile. Tuttavia, studi empirici [14] hanno mostrato che le reti neurali convoluzionali profonde (CNN) possiedono tale capacità, suggerendo che il fenomeno della "memorizzazione" può avere serie implicazioni per la privacy, anche in modelli che apparentemente generalizzano bene. Ciò sottolinea che il rischio di privacy non è confinato ai modelli che esibiscono overfitting.

La Prospettiva di Yeom et al. sugli Attacchi AIA Gli attacchi di inferenza di attributi (AIA) rappresentano una minaccia alla privacy più specifica rispetto ai MIA. In un attacco AIA, un avversario tenta di inferire un attributo mancante (o sconosciuto) di un record parzialmente noto che è stato utilizzato nel set di addestramento, accedendo al modello di machine learning come oracolo [15]. Ad esempio, dato un record con alcuni attributi conosciuti (es. età, sesso) ma con un attributo sensibile mancante (es. una specifica condizione medica), l'attaccante tenta di dedurre quest'ultimo interrogando il modello.

La fattibilità degli attacchi AIA è strettamente correlata alla capacità dell'attaccante di distinguere tra un membro del training set e un non-membro *molto simile*, differendo solo nell'attributo target. Questo è precisamente ciò che il concetto di Strong Membership Inference (SMI) affronta.

Come evidenziato da Zhao et al. [15], anche se un modello di classificazione può essere vulnerabile agli attacchi MIA, è spesso *improbabile* che sia suscettibile agli attacchi AIA. La ragione principale di questa infeasibility risiede nel fatto che gli attacchi MIA standard non sono sufficienti per distinguere un membro da un non-membro che gli è estremamente vicino nello spazio dei dati. Per un attacco AIA di successo, l'avversario dovrebbe essere in grado di discernere una differenza nel comportamento del modello tra, ad esempio, un record (x_0, y_0) nel training set e un record (x_0, y'_0) dove y'_0 è un attributo diverso per lo stesso x_0, e (x_0, y'_0) non è nel training set. Questa capacità richiede una "discriminazione fine" che va oltre la semplice inferenza di appartenenza generale.

Zhao et al. argomentano che se un algoritmo di apprendimento non è vulnerabile a Strong Membership Inference (cioè, l'attaccante ha un vantaggio Adv_{SM} basso), allora è anche resistente agli attacchi di inferenza di attributi. Ciò implica che, per proteggersi dagli AIA, la difesa più efficace potrebbe non essere semplicemente prevenire l'overfitting (che è sufficiente per MIA), ma piuttosto garantire che il modello non "memorizzi" i dettagli così finemente da distinguere tra dati molto simili. In altre parole, la capacità di generalizzare bene a dati *simili* è cruciale per la resistenza agli AIA.

Tuttavia, Yeom et al. [13] notano che, sebbene l'overfitting sia un fattore chiave per la suscettibilità a MIA, gli attacchi AIA possono essere fattibili quando l'attributo target soddisfa "certe condizioni sulla sua influenza". Ciò suggerisce che, in alcuni scenari specifici in cui un attributo ha un impatto molto significativo sul modello, anche senza una SMI forte, l'inferenza di tale attributo potrebbe diventare possibile. La ricerca continua a esplorare le complesse interazioni tra overfitting, generalizzazione, stabilità e le diverse tipologie di attacchi alla privacy.

Dettagli Approfonditi sugli Attacchi di Inferenza di Attributi (AIA) Si approfondisce ora la trattazione degli attacchi di inferenza di attributi (AIA), dove l'obiettivo dell'avversario è **indovinare il valore di una caratteristica sensibile** di un punto dati. L'avversario possiede solo alcune informazioni "pubbliche" su tale punto e l'accesso al modello di machine learning. Per chiarire, in questa sezione si considera che i punti dati sono formati da tre parti: $z = (v, t, y)$, dove (v, t) sono le caratteristiche del dato (input del modello) e y è la sua etichetta (output del modello). L'elemento t rappresenta la caratteristica **sensibile** che l'attaccante cerca di scoprire. Esiste una funzione ϕ che descrive le informazioni che l'avversario conosce sul punto dati, che tipicamente **nasconde l'attributo sensibile** t. L'insieme T contiene tutti i possibili valori che l'attributo sensibile t può assumere nella distribuzione D. La funzione $\pi(z)$ estrae specificamente il valore dell'attributo sensibile t dal punto dati z.

L'inferenza di attributi è formalizzata tramite un test, chiamato **Esperimento 2 (Attribute experiment $\mathbf{Exp}^A(\mathcal{A}, A, n, D)$)**. Questo esperimento è strutturato in modo molto simile all'Esperimento 1 per gli attacchi di inferenza di appartenenza, ma con una differenza fondamentale: all'avversario viene fornita **solo una parte dell'informazione** sul punto dati di sfida z, rappresentata da $\phi(z)$, ovvero la versione "parziale" del dato in cui l'attributo sensibile è sconosciuto.

Le fasi dell'Esperimento 2 sono le seguenti:

1. **Campionamento del Training Set:** Viene generato un set di dati S campionando n punti in modo indipendente dalla distribuzione generale D. Su questo set S viene poi addestrato il modello A_S.
2. **Scelta Casuale dello Scenario:** Un valore binario b (che può essere 0 o 1) viene scelto in modo completamente casuale. Questo valore decide la natura del punto dati di sfida per l'avversario.
3. **Generazione del Punto di Sfida:**

- Se $b = 0$, il punto dati di sfida z viene estratto direttamente dal training set S. In questo caso, z è un dato che il modello A_S ha "visto" durante l'addestramento.
- Se $b = 1$, il punto dati di sfida z viene estratto dalla distribuzione generale D. In questo caso, z è un dato che il modello A_S potrebbe non aver mai visto, ed è trattato come un "non-membro" per scopi di confronto.

4. **La Sfida dell'Avversario:** All'avversario $\mathcal{A}$ viene fornita la versione parziale del punto z (ovvero $\phi(z)$), l'accesso al modello A_S, la dimensione del training set n e la distribuzione D. L'avversario deve quindi **indovinare il valore dell'attributo sensibile** $\pi(z)$. L'esperimento restituisce 1 se l'ipotesi dell'avversario è corretta (cioè, la sua predizione corrisponde al vero attributo $\pi(z)$), e 0 altrimenti.

Lo scopo di questo esperimento è quantificare, attraverso la **Definizione 5 (Vantaggio di Attributo)**, la quantità di informazioni sull'attributo sensibile $\pi(z)$ che il modello A_S "rivela" specificamente quando il punto dati z proviene dal suo training set. La Definizione 5 confronta quanto è bravo l'avversario a indovinare l'attributo quando z è un membro ($b = 0$) rispetto a quando z non lo è ($b = 1$).

La **Definizione 5 (Vantaggio di Attributo)** è formulata come segue:

$$\text{Adv}^A(\mathcal{A}, A, n, D) = \Pr[\text{Exp}^A(\mathcal{A}, A, n, D) = 1|b = 0] - \Pr[\text{Exp}^A(\mathcal{A}, A, n, D) = 1|b = 1].$$

Le probabilità qui considerate tengono conto della casualità nelle scelte dell'avversario, nel campionamento del training set S e nella generazione del punto dati z. In termini più dettagliati, questa formula può essere espressa come:

$$\text{Adv}^A = \sum_{t_i \in T} \Pr_{z \sim D}[t = t_i](\Pr[\mathcal{A} = t_i|b = 0, t = t_i] - \Pr[\mathcal{A} = t_i|b = 1, t = t_i]), \quad (5)$$

dove $\mathcal{A}$ e Adv^A sono abbreviazioni per indicare, rispettivamente, l'output dell'avversario e il vantaggio di attributo completo. Questa formula mostra che il vantaggio dell'attaccante si basa sulla differenza tra la sua capacità di indovinare correttamente l'attributo t_i quando il punto z è un membro del training set ($b = 0$) e la sua capacità di indovinare lo stesso attributo quando il punto z proviene dalla distribuzione generale ($b = 1$). Se questa differenza è grande, significa che il modello sta rivelando informazioni sull'attributo sensibile specificamente per i dati di training.

Un aspetto critico della Definizione 5 è che potrebbe indurre l'avversario a "giocare sporco", ottenendo intenzionalmente prestazioni scarse quando sa che il punto non proviene dal training set ($b = 1$). Per affrontare questo problema, è stata proposta una definizione alternativa, che introduce un "simulatore Bayesiano ottimale".

Definizione 6 (Vantaggio di Attributo Alternativo): In questa definizione, il comportamento dell'avversario quando $b = 1$ viene confrontato con quello di un **simulatore ideale** $\mathcal{S}(\phi(z), n, D)$. Questo simulatore è il migliore possibile nell'indovinare l'attributo target $\pi(z)$ basandosi solo sulle informazioni pubbliche

$\phi(z)$, sulla dimensione n del training set e sulla distribuzione generale D, senza avere accesso al modello A_S. Il vantaggio alternativo è quindi:

$$\mathrm{Adv}^{A}_{\mathcal{S}}(\mathcal{A}, A, n, D) = \Pr[\mathcal{A}(\phi(z), A_S, n, D) = \pi(z)|b = 0] - \Pr[\mathcal{S}(\phi(z), n, D) = \pi(z)|b = 1].$$

Questa definizione mira a isolare meglio il rischio di privacy che il modello *specificamente* introduce per i dati di training. Tuttavia, essa presenta un inconveniente: una maggiore accuratezza generale del modello (cioè, il modello apprende bene anche le tendenze generali della distribuzione D) porterebbe comunque a un vantaggio di attributo più elevato, indipendentemente dal fatto che l'accuratezza derivi da overfitting o da una reale capacità di generalizzazione. Gli autori del paper hanno preferito mantenere la Definizione 5 per la loro analisi, poiché essa consente di distinguere più chiaramente il ruolo dell'overfitting rispetto all'apprendimento di tendenze generali. Sebbene gli avversari che "manipolano il sistema" (come consentito dalla Definizione 5) possano sembrare problematici, la loro efficacia indica comunque una perdita di privacy, poiché la loro esistenza implica la capacità di inferire l'appartenenza, come dimostrato dall'Avversario di Riduzione 5 nella Sezione 5.1.

Inversione, Generalizzazione e Influenza Un caso particolare di AIA è l'**inversione del modello** (model inversion) [4], dove la funzione ϕ nasconde solo l'attributo sensibile t, ovvero fornisce all'avversario il punto dati z meno l'attributo t. L'obiettivo è ricostruire t.

In questa sezione, si analizza l'attacco di inversione del modello proposto da Fredrikson et al. [4] utilizzando la Definizione 5 del vantaggio di attributo. Questa analisi è significativa perché il vantaggio è qui definito per riflettere quanto un attacco di inferenza di attributi riveli informazioni su **singoli individui specifici** presenti nel training set. L'obiettivo non è solo valutare l'accuratezza dell'attacco, ma comprendere i fattori che aumentano il rischio per la privacy dei membri del training set. A tal fine, si esaminerà la relazione tra il vantaggio dell'attaccante e l'errore di generalizzazione, nonché il ruolo dell'**influenza** di una caratteristica. L'influenza di una caratteristica t si riferisce a quanto una modifica di t possa alterare l'output del modello $A_S(x)$.

L'attacco di Fredrikson et al. si concentra sui modelli di regressione lineare e assume che gli errori del modello seguano una distribuzione Gaussiana (come discusso nella Sezione 3.2). In generale, se l'avversario può stimare bene la distribuzione dell'errore (ad esempio, assumendo una Gaussiana con deviazione standard nota), può ottenere un vantaggio provando tutti i possibili valori dell'attributo sensibile. L'approssimazione dell'avversario della distribuzione dell'errore è indicata con $f_{\mathcal{A}}$. L'attributo target t può assumere un numero finito di valori $t_1, \ldots, t_m$ con probabilità note nella distribuzione D. Le altre caratteristiche note all'avversario sono indicate con v. L'attacco è descritto dall'**Avversario 4**. Per ogni possibile valore t_i, l'avversario immagina che $t = t_i$ e calcola l'errore che il modello produrrebbe. Usa queste informazioni per aggiornare la sua conoscenza e scegliere il valore di t_i più probabile.

Avversario 4 (Generale): Sia $f_{\mathcal{A}}(\epsilon)$ la stima dell'avversario per la densità di probabilità dell'errore $\epsilon = y - A_S(x)$. Dati la parte pubblica del punto v, l'etichetta y, il modello A_S, la dimensione n del training set e la distribuzione D, l'avversario procede come segue:

1. Per ogni possibile valore t_i dell'attributo sensibile (da t_1 a t_m), l'avversario interroga il modello A_S con il punto (v, t_i) per ottenere la previsione $A_S(v, t_i)$.
2. Per ciascun t_i, calcola l'errore che si otterrebbe: $\epsilon(t_i) = y - A_S(v, t_i)$. Questo è l'errore tra la vera etichetta y e la predizione del modello, supponendo che l'attributo sensibile sia t_i.
3. L'avversario restituisce il valore t_i che massimizza il prodotto tra la probabilità a priori che $t = t_i$ nella distribuzione generale D ($\Pr_{z\sim D}[t = t_i]$) e la probabilità che si verifichi l'errore $\epsilon(t_i)$ data la sua stima $f_{\mathcal{A}}(\epsilon(t_i))$. In pratica, sceglie il t_i che è più probabile, considerando sia la sua frequenza generale che la compatibilità con l'output del modello.

Nell'analizzare l'Avversario 4, l'attenzione è rivolta all'impatto dell'errore di generalizzazione sul vantaggio. Ci si aspetta che un ampio errore di generalizzazione (overfitting) porti a un maggiore vantaggio. Tuttavia, l'influenza funzionale (definita come la probabilità che cambiare t causi un cambiamento nell'output $A_S(v, t)$) può giocare un ruolo cruciale. Per i modelli lineari, l'influenza di t corrisponde al valore assoluto del suo coefficiente normalizzato.

Variabile Binaria con Prior Uniforme: Si consideri il caso più semplice in cui l'attributo sensibile t è binario (può assumere solo due valori, t_1 o t_2) e questi valori sono ugualmente probabili nella popolazione generale. Si assume che l'output del modello per t_1 e t_2 differisca di una quantità $\tau \geq 0$, dove τ è una misura dell'**influenza** di t. Un τ maggiore significa che t ha un impatto più significativo sull'output del modello.

Teorema 5: Se t è scelto uniformemente da $\{t_1, t_2\}$ e l'errore ϵ è Gaussiano con varianza σ_S^2 per i membri e σ_D^2 per i non-membri, allora il vantaggio dell'Avversario 4 è:

$$\frac{1}{2}\left(\operatorname{erf}\left(\frac{\tau}{2\sqrt{2}\sigma_S}\right) - \operatorname{erf}\left(\frac{\tau}{2\sqrt{2}\sigma_D}\right)\right).$$

Dimostrazione: L'Avversario 4, in questo scenario semplificato, sceglie il valore t_i che minimizza il valore assoluto dell'errore $|\epsilon(t_i)|$. L'intuizione è che un membro del training set dovrebbe avere un errore più piccolo rispetto a un non-membro. Se il vero attributo è t_1, l'avversario indovina correttamente se l'errore $\epsilon(t_1)$ è superiore a una certa soglia $(-\tau/2)$. Questa soglia deriva dal punto in cui la probabilità di osservare l'errore dato t_1 è uguale alla probabilità di osservare l'errore dato t_2. Il vantaggio dell'avversario, quando il vero attributo è t_1, è la differenza tra la probabilità che indovini correttamente per i membri ($b = 0$) e per i non-membri ($b = 1$). Poiché gli errori sono distribuiti normalmente, si usa la funzione d'errore (erf) per calcolare queste probabilità. La formula mostra che il vantaggio è nullo se $\sigma_S = \sigma_D$ (nessun overfitting), poiché il modello si comporta allo stesso modo su membri e non-membri. Il vantaggio aumenta

quando $\sigma_S \to 0$ (perfetta memorizzazione sui membri) e $\sigma_D \to \infty$ (grandi errori sui non-membri), il che indica un overfitting significativo. Il massimo vantaggio possibile in questo scenario è $1/2$. L'influenza τ è cruciale: se $\tau = 0$ (l'attributo non influenza il modello), l'attaccante non ha vantaggio. Se τ è molto grande, l'influenza dell'attributo maschera il rumore dell'errore, e l'avversario indovina quasi sempre correttamente, ma il vantaggio tende comunque a zero agli estremi (sia τ molto piccolo che molto grande) perché non c'è più distinzione basata sull'overfitting. L'efficacia dell'attacco è massima quando τ e il rapporto σ_D/σ_S sono in equilibrio.

Caso Generale (Prior non Uniforme): In situazioni più complesse, l'attributo t potrebbe non avere una distribuzione uniforme (ad esempio, una malattia rara). L'Avversario 4 tiene conto delle probabilità a priori dei valori di t quando prende una decisione. In questo caso, $\sigma_S = \sigma_D$ implica ancora zero vantaggio. Il massimo vantaggio si ottiene con overfitting estremo ($\sigma_S \to 0, \sigma_D \to \infty$), portando a un vantaggio di $1 - 1/m$, dove m è il numero di possibili valori di t. Il calcolo del vantaggio generale utilizza l'Equazione 5, determinando i confini di decisione tra i valori di t_i e convertendoli in probabilità usando le distribuzioni di errore.

Connessione tra MIA ed AIA Questa sezione esplora il legame tra MIA e AIA attraverso l'uso di **avversari di riduzione**. Questi avversari "riducono" un tipo di attacco all'altro, dimostrando se l'efficacia in un attacco implica l'efficacia nell'altro.

Dall'Inferenza di Appartenenza all'Attributo (e Viceversa) From Membership to Attribute (Avversario 5): Si consideri un avversario ($\mathcal{A}^{M \to A}$) il cui obiettivo è eseguire un'inferenza di **appartenenza**, ma che ha a disposizione un "oracolo" in grado di eseguire un'inferenza di **attributo**. **Avversario 5 (Membership → Attribute):** Dato un punto z, il modello A_S, la dimensione n e la distribuzione D:

1. L'avversario $\mathcal{A}^{M \to A}$ interroga l'oracolo di attributo ($\mathcal{A}^A$) fornendogli le informazioni parziali $\phi(z)$ e il modello A_S. L'oracolo restituisce una predizione t per l'attributo sensibile.
2. Se la predizione t è corretta (cioè, $t = \pi(z)$), l'avversario $\mathcal{A}^{M \to A}$ conclude che il punto z è un membro del training set (restituisce 0). Altrimenti, conclude che non lo è (restituisce 1).

Teorema 6: Questo teorema afferma che il vantaggio dell'Avversario 5 nel compiere un'inferenza di appartenenza è esattamente uguale al vantaggio dell'oracolo di attributo utilizzato.

$$\mathrm{Adv}^M(\mathcal{A}^{M \to A}, A, n, D) = \mathrm{Adv}^A(\mathcal{A}^A, A, n, D).$$

Dimostrazione: La dimostrazione è una conseguenza diretta delle definizioni di vantaggio di appartenenza e di attributo. Il vantaggio di appartenenza misura la differenza tra la probabilità che l'avversario indovini correttamente un membro

($b = 0$) e la probabilità che indovini correttamente un non-membro ($b = 1$). Se l'Avversario 5 predice "membro" solo quando l'oracolo di attributo è corretto, allora il suo successo nel rilevare l'appartenenza riflette direttamente l'accuratezza dell'oracolo di attributo nel distinguere i dati di training da quelli generali. Questo risultato suggerisce che l'inferenza di attributi è un compito almeno "difficile" quanto l'inferenza di appartenenza.

From Attribute to Membership (Avversario 6 e 7): Questa direzione è più complessa. L'avversario ha $\phi(z)$ (informazione parziale) e deve ricostruire il punto z per interrogare un oracolo di **appartenenza**. Si assume che l'avversario conosca una funzione ϕ^{-1} che può ricostruire un punto z' da $\phi(z)$, anche se z' potrebbe non essere identico a z a causa della perdita di informazione in ϕ.

Avversario 6 (Uniform attribute → membership): Questo avversario cerca di indovinare l'attributo sensibile t di un punto, usando un oracolo di appartenenza. Dato $\phi(z)$, il modello A_S, n e D:

1. Sceglie un valore t_i per l'attributo sensibile in modo completamente casuale tra tutti i possibili m valori.
2. Ricostruisce un punto z' usando $\phi^{-1}(\phi(z))$ e poi imposta l'attributo sensibile di z' al valore t_i scelto casualmente.
3. Interroga l'oracolo di appartenenza ($\mathcal{A}^M$) con il punto z'. L'oracolo risponde se z' è un membro (0) o meno (1).
4. Se l'oracolo di appartenenza risponde "membro" (0), l'avversario restituisce il valore t_i che aveva scelto. Altrimenti, non fornisce alcuna predizione.

La scelta casuale di t_i è una limitazione, poiché l'avversario non sa quale t_i è più probabile. Questo avversario ha successo solo se indovina t_i e l'oracolo di appartenenza conferma che il punto corrispondente è un membro. **Teorema 7**: Il vantaggio dell'Avversario 6 nell'inferenza di attributo è proporzionale al vantaggio modificato dell'oracolo di appartenenza, diviso per il numero di possibili valori di t (m).

$$\mathrm{Adv}^A(\mathcal{A}^U_{A\to M}, A, n, D) = \frac{1}{m}\mathrm{Adv}^M_*(\mathcal{A}^M, A, n, D, \phi, \phi^{-1}, \pi).$$

Dimostrazione: La probabilità che l'Avversario 6 indovini correttamente l'attributo t dipende da due fattori: la probabilità di scegliere il t_i corretto (che è $1/m$) e la probabilità che l'oracolo di appartenenza risponda che il punto ricostruito z' con quel t_i è un membro. La dimostrazione formale scompone queste probabilità e mostra che il vantaggio totale è la frazione $1/m$ del vantaggio dell'oracolo di appartenenza. Questo stabilisce un limite inferiore per l'efficacia degli AIA che usano oracoli di appartenenza. Sebbene non trasferisca tutto il vantaggio (a causa del fattore $1/m$), dimostra che è comunque possibile, in una certa misura, dedurre attributi usando un oracolo di appartenenza.

L'Avversario 6 è limitato dal dover indovinare t_i. L'**Avversario 7** cerca di migliorare questo aspetto effettuando più interrogazioni all'oracolo di appartenenza. **Avversario 7 (Multi-query attribute → membership):** Dato $\phi(z)$, il modello A_S, n e D:

1. Ricostruisce il punto z' come nell'Avversario 6.
2. Per ogni possibile valore t_i, crea una versione di z' con l'attributo sensibile impostato a t_i.
3. Interroga l'oracolo di appartenenza ($\mathcal{A}^M$) per ognuno di questi z'_i. Costruisce un insieme T di tutti i t_i per i quali l'oracolo di appartenenza ha risposto "membro".
4. Tra i valori t_i in questo insieme T, restituisce quello che ha la massima probabilità a priori nella distribuzione generale D. Se l'insieme T è vuoto (nessun t_i ha fatto sì che l'oracolo di appartenenza rispondesse "membro"), non fornisce alcuna predizione.

Questo avversario è più sofisticato e si prevede che sia più efficace, in quanto prova tutte le possibilità e sceglie la più probabile tra quelle che "sembrano" essere membri.

3.3 La Prospettiva di Zhao et al.

Zhao et al. iniziano la loro analisi riprendendo la definizione di inferenza di appartenenza (MI) da lavori precedenti, come quello di Yeom et al. [13]. Un attacco MI mira a determinare se un punto dati specifico è stato utilizzato nel dataset di addestramento di un modello di machine learning.

3.4 Definizione di Membership Inference (MI)

L'**Esperimento 1 (Membership Inference (MI) [13])** formalizza questo concetto:

1. Viene costruito un modello h_X su un dataset X campionando n elementi dalla distribuzione D.
2. Viene campionato un bit $b \leftarrow \{0, 1\}$ casualmente.
3. Se $b = 0$, viene campionato un punto $x \leftarrow D$.
4. Se $b = 1$, viene campionato un punto $x \leftarrow X$.
5. L'avversario $\mathcal{A}$ riceve x, la sua etichetta vera $c(x)$, e l'accesso all'oracolo h_X.
6. L'avversario $\mathcal{A}$ annuncia la sua ipotesi $b' \in \{0, 1\}$. Se $b' = b$, l'output è 1; altrimenti, è 0.

La **Definizione 7 (Membership Inference Advantage)** quantifica il successo di un attacco MI:

$$\mathrm{Adv}_M(\mathcal{A}, h, n, \mathcal{D}) = \Pr[b' = 1|b = 1] - \Pr[b' = 1|b = 0] = \Pr[b' = 0|b = 0] - \Pr[b' = 0|b = 1]$$

Zhao et al. sottolineano che, nella loro tesi, un vantaggio significativo in un attacco MI è dovuto al fatto che i non-membri sono "distanti" nello spazio delle feature dai vettori membri. Se, invece, un vettore non-membro è molto "vicino" a un vettore membro, l'avversario potrebbe non essere in grado di distinguerli. Questa intuizione li porta a introdurre la nozione di Strong Membership Inference (SMI).

4 Strong Membership Inference (SMI)

Per affrontare la limitazione del MIA standard di distinguere tra membri e non-membri molto simili, Zhao et al. introducono il concetto di Strong Membership Inference (SMI).

4.1 Definizione di Strong Membership Inference (SMI)

L'**Esperimento 2 (r-Strong Membership Inference (SMI))** definisce questa sfida più stringente:

1. Viene costruito un modello h_X.
2. Viene campionato un bit $b \leftarrow \{0, 1\}$ casualmente.
3. Viene campionato un punto $x_0 \leftarrow X$ casualmente dal training set.
4. Se $b = 0$, il punto dati di sfida x viene campionato da $B_d(x_0, r)$ (una "sfera" di raggio r centrata in x_0 secondo una metrica di distanza d), in accordo con la distribuzione indotta da D. Ciò significa che x è un vicino di x_0 ma non necessariamente x_0 stesso, e non fa parte del training set.
5. Se $b = 1$, il punto dati di sfida x è semplicemente x_0.
6. L'avversario $\mathcal{A}$ riceve x, la sua etichetta vera $c(x)$, e l'accesso oracolare a h_X.
7. L'avversario $\mathcal{A}$ annuncia la sua ipotesi $b' \in \{0, 1\}$. Se $b' = b$, l'output è 1; altrimenti, è 0.

La **Definizione 8 (Strong Membership Inference Advantage)** quantifica il successo di un attacco SMI:

$$\mathrm{Adv}_{SM}(\mathcal{A}, h, r, n, \mathcal{D}) = \Pr[b' = 1|b = 1] - \Pr[b' = 1|b = 0] = \Pr[b' = 0|b = 0] - \Pr[b' = 0|b = 1]$$

4.2 Relazione tra MI e SMI

Zhao et al. dimostrano formalmente che la SMI è un requisito più stringente rispetto alla MIA. Il **Teorema 1** afferma che esiste una situazione in cui un avversario MI ottiene un vantaggio non trascurabile, mentre un avversario SMI ha un vantaggio trascurabile utilizzando lo stesso algoritmo. Questo implica che il successo di un attacco MI tradizionale non garantisce il successo di un attacco SMI, specialmente quando i non-membri sono molto vicini ai membri nel feature space.

Questo risultato si basa sull'assunto di **vicini indistinguibili (Definition 5)**: intorno a qualsiasi vettore x, esistono vettori campionati secondo la distribuzione indotta da D che sono indistinguibili da x per un dato algoritmo, con un vantaggio $\epsilon(r)$ che è trascurabile per piccoli r.

5 Attacchi di Inferenza di Attributi (Attribute Inference Attacks - AIA)

Gli attacchi di inferenza di attributi (AIA) mirano a inferire un attributo mancante (o sconosciuto) di un record parzialmente noto che è stato usato nel training set.

5.1 Definizione di Attribute Inference (AI)

L'**Esperimento 3 (Attribute Inference (AI) [13])** è definito come segue:

1. Viene costruito il modello h_X.
2. Viene campionato un bit $b \leftarrow \{0, 1\}$ casualmente.
3. Se $b = 0$, viene campionato un punto $x \leftarrow D$.
4. Se $b = 1$, viene campionato un punto $x \leftarrow X$.
5. Sia $x^* = \phi_S(x)$ una porzione di x (con un sottoinsieme S di feature sconosciute di cardinalità m').
6. L'avversario $\mathcal{A}$ riceve x^*, la sua etichetta vera $c(x)$, e l'accesso oracolare a h_X.
7. L'avversario $\mathcal{A}$ annuncia $x' \in D^m$. Se $x' = x$, l'output è 1; altrimenti, è 0.

La **Definizione 9 (Attribute Inference Advantage)** quantifica il successo di un attacco AI:

$$\mathrm{Adv}_{AI}(\mathcal{A}, h_X, m', n, \mathcal{D}) = \Pr[\mathrm{Exp}_{AI}(\mathcal{A}, h_X, m', n, \mathcal{D}) = 1 | b = 1] - \Pr[\mathrm{Exp}_{AI}(\mathcal{A}, h_X, m', n, \mathcal{D}) = 1 | b = 0]$$

Zhao et al. enfatizzano che questa definizione valuta se l'inferenza tramite il modello è più efficace dell'inferenza basata unicamente sulla distribuzione dei dati. Un attacco AI è vantaggioso solo se il modello rivela informazioni aggiuntive.

5.2 Relazione tra AI e SMI

Zhao et al. argomentano che la **SMI è una condizione necessaria per la fattibilità degli AIA**. Se un avversario non riesce a distinguere tra un membro del training set e un suo vicino non-membro, allora non potrà nemmeno inferire l'attributo mancante di un dato parzialmente noto. Il **Teorema 2** formalizza questa relazione:

Teorema 2: Sia $\mathcal{A}$ un avversario AI con vantaggio δ. Allora esiste un avversario SMI $\mathcal{B}$ con vantaggio $\delta + \epsilon(r)$, assumendo che $\epsilon(r)$, il vantaggio di distinguibilità dei vicini r, sia trascurabile per piccoli r. Questo teorema, insieme al risultato precedente (Theorem 1), mostra che $SMI \Leftrightarrow AI$ (SMI è equivalente a AI), a condizione che l'assunto di vicini indistinguibili sia valido. Di conseguenza, se un attacco SMI ha un vantaggio trascurabile, allora anche un attacco AI avrà un vantaggio trascurabile.

6 Approximate Attribute Inference (AAI)

Dato che gli AIA "esatti" risultano spesso infeasible per i modelli di classificazione, Zhao et al. introducono una nozione più rilassata: l'**Approximate Attribute Inference (AAI)**. Qui, l'avversario deve indovinare un valore che sia "sufficientemente vicino" all'attributo vero.

6.1 Definizione di Approximate Attribute Inference (AAI)

L'**Esperimento 4 (Approximate Attribute Inference (AAI))** è una variazione dell'Esperimento 3:

1. Vengono eseguiti i passaggi 1-6 dell'Esperimento 3 (costruzione del modello, scelta di b, campionamento di x, creazione della porzione x^*, e fornitura all'avversario).
2. L'avversario $\mathcal{A}$ annuncia $x' \in D^m$.
3. Se la distanza $d(x', x)$ è minore o uguale a un parametro α (cioè, $d(x', x) \leq \alpha$), l'output è 1; altrimenti, è 0.

La **Definizione 10 (Approx. Attribute Inference Advantage)** quantifica il successo di un AAI:

$$\text{Adv}_{AI}(\mathcal{A}, h_X, m', n, \alpha, \mathcal{D}) = \Pr[\text{Exp}_{AI}(\mathcal{A}, h_X, m', n, \alpha, \mathcal{D}) = 1 | b = 1] -$$

$$-\Pr[\text{Exp}_{AI}(\mathcal{A}, h_X, m', n, \alpha, \mathcal{D}) = 1 | b = 0].$$

Con $\alpha = 0$, l'AAI è equivalente all'AI esatto. Si osserva che $AI \Rightarrow AAI$, ma il contrario non è necessariamente vero.

7 Metodologie Specifiche di Attacco Esaminate da Zhao et al.

Zhao et al. non si limitano alle definizioni, ma valutano sperimentalmente diverse implementazioni concrete di attacchi MI e AI.

7.1 Attacchi di Inferenza di Appartenenza (MI) Specifici

Gli autori esaminano cinque attacchi MI preesistenti, tre "black-box" (che accedono al modello solo tramite le sue previsioni) e due "white-box" (che ottengono informazioni più approfondite sul modello):

- **Shadow MI [11]:** Questo attacco addestra un "modello di attacco" ausiliario per discernere l'appartenenza, sfruttando le previsioni (e la loro confidenza) di "modelli ombra" che replicano il comportamento del modello target.
- **Loss MI [13]:** Questo attacco valuta direttamente la funzione di perdita del vettore sul modello target, utilizzando la perdita di training come soglia per determinare l'appartenenza. I dati di training tendono ad avere una perdita inferiore.
- **Conf MI [9]:** Una versione più semplice del Loss MI, che utilizza solo il valore di confidenza della previsione più probabile per inferire l'appartenenza. Richiede meno informazioni.
- **Local White Box (WB) MI [8]:** Attacco "white-box" dove l'avversario ha accesso a informazioni interne del modello, come i gradienti degli strati finali e gli stati intermedi del processo di addestramento.
- **Global White Box (WB) MI [8]:** Simile al Local WB MI, ma in uno scenario in cui l'attaccante ha informazioni a livello di server e attacca singole parti che hanno contribuito all'addestramento.

7.2 Attacchi di Inferenza di Attributi (AI) Specifici

Gli attacchi AI esaminati da Zhao et al. utilizzano gli attacchi MI come "sottoprogrammi" o "oracoli". La procedura generale è: dato un punto dati con attributi mascherati, l'avversario genera tutte le possibili "variazioni" (siblings) modificando gli attributi mancanti. Ogni sibling viene quindi passato a un attacco MI, e il sibling con la "confidenza di appartenenza" più alta viene ritenuto la ricostruzione corretta.

- **Shadow AI:** Utilizza il modello di attacco di Shadow MI per valutare la confidenza di appartenenza di ciascun sibling.
- **Loss AI [13]:** Basato sulla proposta di Yeom et al., questo attacco seleziona il sibling che produce una perdita più vicina alla perdita media del training set dal modello target.
- **Conf AI [16]:** Simile a Conf MI, questo attacco identifica il sibling che ottiene la confidenza di previsione più alta dal modello target.

8 Datasets

In questo capitolo, esploreremo in dettaglio i dataset impiegati per condurre gli attacchi di *Membership Inference Attack (MIA)* e *Attribute Inference Attack (AIA)*, incluse le loro varianti. Analizzeremo il comportamento degli attacchi su ciascun dataset, facendo riferimento ai lavori "On the (In)Feasibility of Attribute Inference Attacks on Machine Learning Models" [15] e "Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting" [13].

8.1 On the (In)Feasibility of Attribute Inference Attacks on Machine Learning Models

Questo lavoro esamina in profondità la fattibilità degli attacchi di inferenza di attributi sui modelli di machine learning, in particolare focalizzandosi su come la loro efficacia sia influenzata dalla capacità di distinguere un membro del training set da un non-membro "vicino" (Strong Membership Inference).

8.1.1 Dataset Location

Descrizione del Dataset: Il dataset Location [12] costituisce una collezione di 467 feature binarie inizialmente non etichettate, derivate da dati di check-in di social network. Per gli esperimenti, sono state generate 30 etichette mediante l'applicazione di un algoritmo di clustering k-means, da cui deriva la designazione del dataset come Loc-30 negli esperimenti condotti. Data la sua intrinseca natura binaria, per quantificare la "vicinanza" tra i vettori di feature, è stata impiegata la distanza di Hamming (d_H). Al fine di esaminare le performance degli attacchi di inferenza di appartenenza (Membership Inference - MI) in funzione di diverse distanze, sono stati metodicamente generati vettori non-membri sintetici. Questo

processo ha coinvolto la selezione casuale di un membro dal set di addestramento e la successiva inversione di un numero variabile di feature, coprendo un intervallo di distanze di Hamming che spazia da 1 a 467. Per ciascun gruppo di distanza, sono stati creati 5 non-membri sintetici. Il modello target è una rete neurale completamente connessa (fully connected NN) composta da un singolo strato nascosto di 128 nodi e dotata di funzione di attivazione "tanh". Le configurazioni adottate per la suddivisione dei dati in relazione alle diverse metodologie di attacco sono state le seguenti:

- Per gli attacchi Confidenza MI (Conf MI) e Perdita MI (Loss MI), il dataset completo è stato partizionato con il 20% destinato all'addestramento del modello target e il restante 80% per le fasi di testing.
- Nel contesto dell'attacco Shadow MI, la ripartizione del dataset ha previsto il 20% per l'addestramento del modello target, il 64% per l'addestramento dei modelli 'shadow' e il residuo 16% per le fasi di testing. Per il dataset Location, sono stati impiegati complessivamente 60 modelli 'shadow'.
- Per gli attacchi Local White Box (WB) e Global White Box (WB) MI, sono stati utilizzati 1.158 record per l'addestramento iniziale del modello target (corrispondente al 20% del dataset) e 5.790 record per il testing (pari all'80%)

Per gli attacchi di inferenza di attributi, un set di 15 feature binarie, identificate come le più informative tramite l'applicazione del criterio mRMR (Minimal Redundancy Maximal Relevance), sono state selezionate e successivamente mascherate. Nel contesto dell'Inferenza Approssimata di Attributi (Approximate Attribute Inference - AAI) applicata alle 15 feature incognite del dataset Location, il parametro di distanza α è stato fissato a 7.5. Tale valore è stato determinato come la distanza media che una supposizione casuale presenterebbe rispetto ai valori originali.

Esiti degli Attacchi:

- **Membership Inference (MI):** Gli attacchi MI, quantificati tramite l'Area Under the Curve (AUC), hanno mostrato un miglioramento progressivo all'aumentare della distanza dei non-membri dal training set. Al contrario, per i non-membri più prossimi al set di addestramento, l'AUC tende ad avvicinarsi a un valore che indica una supposizione casuale (0.5). Le analisi hanno altresì evidenziato che gli attacchi di Strong Membership Inference (SMI) si sono dimostrati meno efficaci nel discriminare tra membri e non-membri che sono molto simili. Ad esempio, per il dataset Location, pur mostrando una Loss MI un AUC elevato a distanze maggiori (es. 0.7 a distanza di Hamming 10), la sua efficacia diminuisce drasticamente per distanze più ravvicinate (AUC prossimo a 0.5 tra distanza 1 e 3). Questa osservazione è cruciale, poiché suggerisce che l'attuale formulazione degli attacchi MI non cattura adeguatamente il comportamento di un avversario MI per i non-membri situati a piccole distanze dai dati di addestramento, una capacità che risulta indispensabile per l'efficacia degli attacchi di inferenza di attributi. La performance degli attacchi MI non è risultata uniforme tra tutte le classi del

dataset, manifestando una correlazione inversa con la dominanza di classe (intesa come l'ampiezza della sua regione di decisione nello spazio delle feature). Le classi che occupano una porzione significativamente maggiore dello spazio delle decisioni si sono dimostrate meno suscettibili sia agli attacchi MI che agli attacchi SMI. Le osservazioni concernenti la difficoltà degli attacchi MI nel distinguere tra membri e non-membri vicini sono state coerentemente riscontrate anche con l'impiego di altri modelli di machine learning, quali la Regressione Logistica (LR), le Macchine a Vettori di Supporto (SVM) e i Random Forests (RF), indicando la generalizzabilità di tale fenomeno oltre le sole reti neurali.

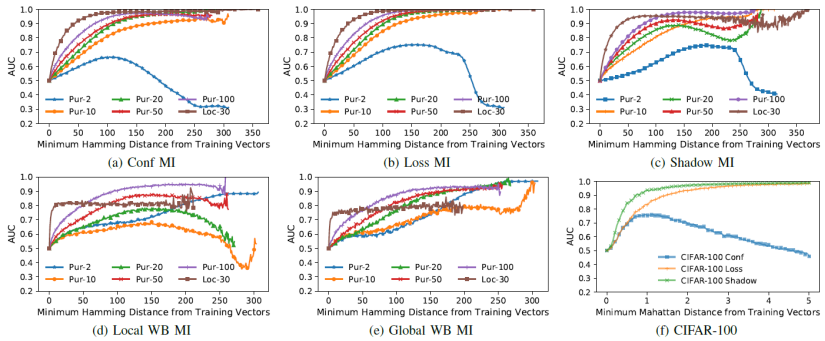
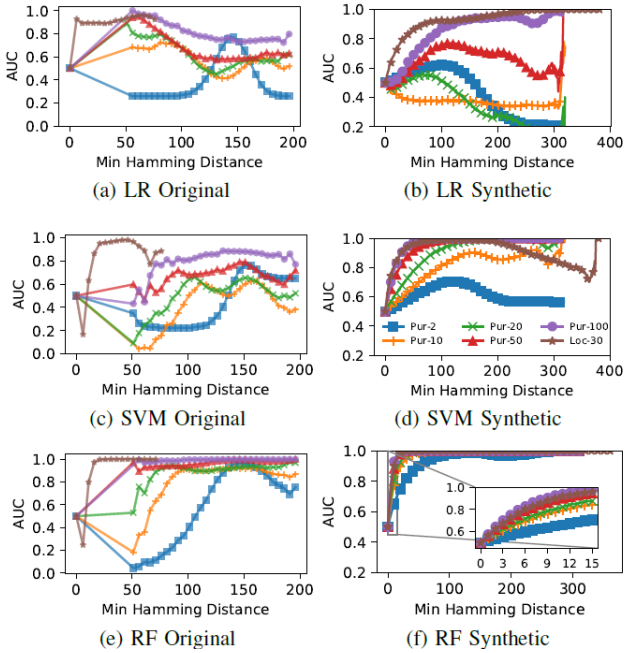


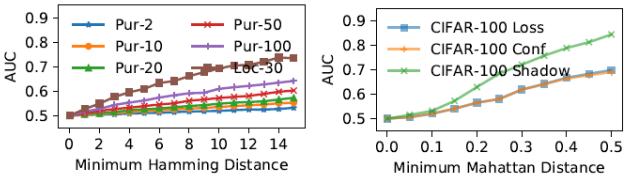
Fig. 3. Curva ROC degli attacchi Conf MI, Loss MI, Shadow MI, Local WB MI, Global WB MI su vari dataset, inclusi Location e Purchase.

- **Attribute Inference (AI):** Nonostante i modelli target fossero vulnerabili agli attacchi MI, gli attacchi di Inferenza Esatta di Attributi (AI) sul dataset Location hanno esibito un vantaggio trascurabile, prossimo allo zero. Questo esito si contrappone a quanto rilevato in studi precedenti su problemi di regressione, dove l'AI si era dimostrata fattibile. Tale discrepanza è attribuibile alla natura discreta dell'etichetta di classe nei problemi di classificazione, la quale fornisce meno informazioni all'attaccante rispetto a un valore continuo. La ragione principale di questa inefficacia risiede nell'incapacità intrinseca degli attacchi MI (utilizzati come subroutine per l'AI) di distinguere efficacemente tra membri e non-membri molto simili (ossia, il fallimento di SMI).
- **Approximate Attribute Inference (AAI):** In contrasto con l'AI esatta, gli attacchi AAI, che mirano all'inferenza di attributi approssimativamente vicini ai valori reali, hanno mostrato un vantaggio considerevolmente superiore rispetto all'AI, sebbene rimanga inferiore al massimo teorico di 1. Per il dataset Location, la Loss AI ha raggiunto un vantaggio AAI di 0.1609. Questo risultato indica la possibilità di inferire attributi approssimativamente corretti con una percentuale di successo significativamente superiore rispetto a una mera supposizione casuale. È stata inoltre osservata una correlazione positiva tra il livello di overfitting del modello target e il vantaggio



(a) LR Original (b) LR Synthetic (c) SVM Original (d) SVM Synthetic (e) RF Original (f) RF Synthetic

Fig. 4. Curva ROC degli attacchi MI su Location e Purchase con diversi modelli (LR, SVM, RF).



(a) Loss MI - Location and Purchase, 15 hamming distance. (b) CIFAR-100, zoomed to 0.5 Manhattan distance.

Fig. 5. Confronto del vantaggio per Loss MI su Location e Purchase (a) per attacchi di inferenza di attributi a 15 bit, e (b) zoom del dataset CIFAR-100.

AAI; in particolare, modelli che presentano un maggiore overfitting (come il Location-30) tendono a manifestare un vantaggio AAI più elevato.

In sintesi, lo studio conclude che l'inferenza esatta degli attributi risulta infeasibile per i modelli di classificazione, anche qualora questi siano vulnerabili agli attacchi di inferenza di appartenenza. Questa infeasibilità è primariamente dovuta alla difficoltà di distinguere tra membri e non-membri molto simili (SMI). Ciononostante, una nozione più flessibile di inferenza di attributi, l'AAI, si dimostra fattibile, e la sua efficacia è correlata positivamente al grado di overfitting del modello target.

8.1.2 Purchese

Descrizione del dataset: Il dataset Purchase[2] è una raccolta di transazioni di acquisto (599 caratteristiche binarie, dove '1' indica l'acquisto di un articolo e '0' il non acquisto), originariamente derivante dalla "Acquire valued shoppers challenge" su Kaggle. Per l'analisi, sono state create cinque varianti del dataset (Pur-2, Pur-10, Pur-20, Pur-50, Pur-100) differenziate dal numero di classi, ottenute tramite clustering k-means. La distanza di Hamming (dH) è la metrica impiegata per questo dataset, data la sua natura binaria. Negli esperimenti, il dataset è stato campionato in diverse quantità (es. 20.000 o 40.000 record) e diviso in set di training e testing con proporzioni variabili (es. 80% training, 20% testing per Conf e Loss MI; 25% target, 67.5% shadow, 7.5% testing per Shadow MI). È stato anche esaminato l'overfitting, utilizzando dataset con dimensioni fino a 200.000 record per Purchase-100. Per gli attacchi di inferenza di attributi (AI e AAI), sono state mascherate 15 caratteristiche binarie considerate "più informative" (identificate con il criterio mRMR). Per l'AAI, il parametro di distanza (alpha) è stato fissato a 7.5 per il dataset Purchase. Il dataset Purchase è considerato idoneo all'assunto di vicini indistinguibili (Indistinguishable Neighbor Assumption), ovvero modifiche minime agli attributi non sono considerate anomalie significative. Questa ipotesi è stata convalidata addestrando una Rete Avversaria Generativa (GAN) sul dataset, che ha mostrato un vantaggio minimo nel distinguere vettori reali da vettori perturbati a piccole distanze. L'architettura delle reti neurali utilizzate prevedeva uno strato nascosto di 128 nodi con funzione di attivazione "tanh". Si è osservato che un maggior numero di classi nel dataset Purchase genera un overfitting maggiore.

Comportamento degli Attacchi di Inferenza di Appartenenza (MI) Gli attacchi MI mirano a determinare se un record faceva parte del dataset di training del modello. Sono stati usati cinque tipi di attacchi MI: Conf MI, Loss MI, Shadow MI (black-box), e Local WB MI, Global WB MI (white-box).

1. Performance in funzione della Distanza dal Dataset di Training:
   - Dati Originali: L'AUC (Area Under the Curve), metrica di performance dell'attacco MI, mostra un miglioramento meno evidente con l'aumentare della distanza dei non-membri dal dataset di training rispetto ad altri dataset. Ciò è dovuto al fatto che i non-membri nel dataset Purchase sono già a una distanza maggiore dal dataset di training originale, limitando la disponibilità di dati a distanze molto vicine o molto lontane.
   - Dati Sintetici: L'utilizzo di vettori sintetici ha rivelato che l'AUC per gli attacchi MI è prossima a 0.5 (ipotesi casuale) per vettori vicini al dataset di training, migliorando solo con l'aumentare della distanza, indicando un fallimento degli attacchi MI nel senso di Strong Membership Inference (SMI) a piccole distanze.
   - Overfitting e Classi: La performance dell'attacco MI migliora all'aumentare del numero di classi nel modello target, indicando una correlazione positiva tra overfitting e successo dell'attacco MI.

– Conf MI vs. Loss MI: Le performance di AUC di Loss MI e Conf MI sono quasi identiche per il dataset Purchase, in quanto la massima confidenza di predizione (Conf MI) spesso coincide con la confidenza usata per calcolare la perdita (Loss MI) nei modelli di classificazione.
– Comportamento Irregolare: Alcune curve AUC mostrano picchi e cali, in particolare per le varianti a 2, 10 e 20 classi, dovuti al fatto che a determinate distanze un non-membro può trovarsi nella regione di decisione di un'altra classe.

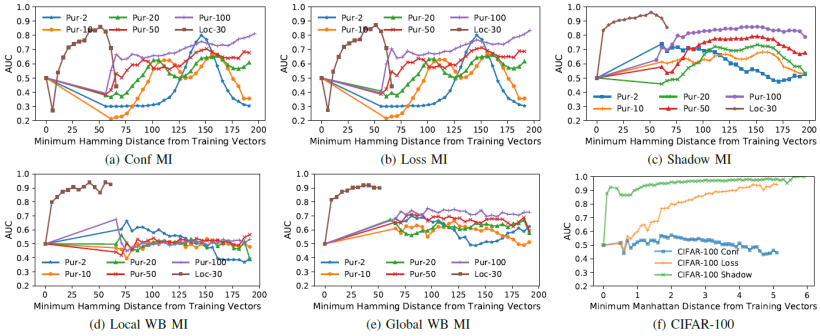


Fig. 6. Curva ROC degli attacchi Conf MI, Loss MI, Shadow MI, Local WB MI, Global WB MI su varianti del dataset Purchase e Loc-30. La parte (f) mostra i risultati su CIFAR-100.

2. Performance per Classe (Decision Region): Le analisi per classe (es. su Purchase-20) mostrano che l'AUC della classe più dominante (con la regione di decisione più ampia) è significativamente inferiore alla media, specialmente a distanze vicine al dataset. Questo implica che gli attacchi MI faticano a distinguere membri e non-membri di classi dominanti, portando a performance complessive inferiori per problemi di classificazione binaria.

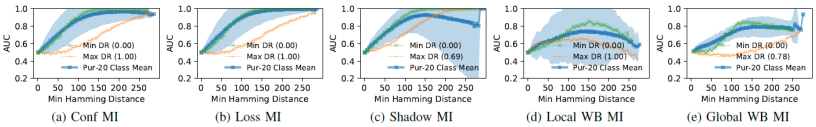


Fig. 7. Performance dell'attacco MI per classe (Min/Max DR) su Purchase-20 per Conf MI, Loss MI, Shadow MI, Local WB MI, Global WB MI.

3. Generalizzazione ad altri Modelli: Le osservazioni sugli attacchi MI si estendono anche a modelli di Regressione Logistica, SVM e Random Forest,

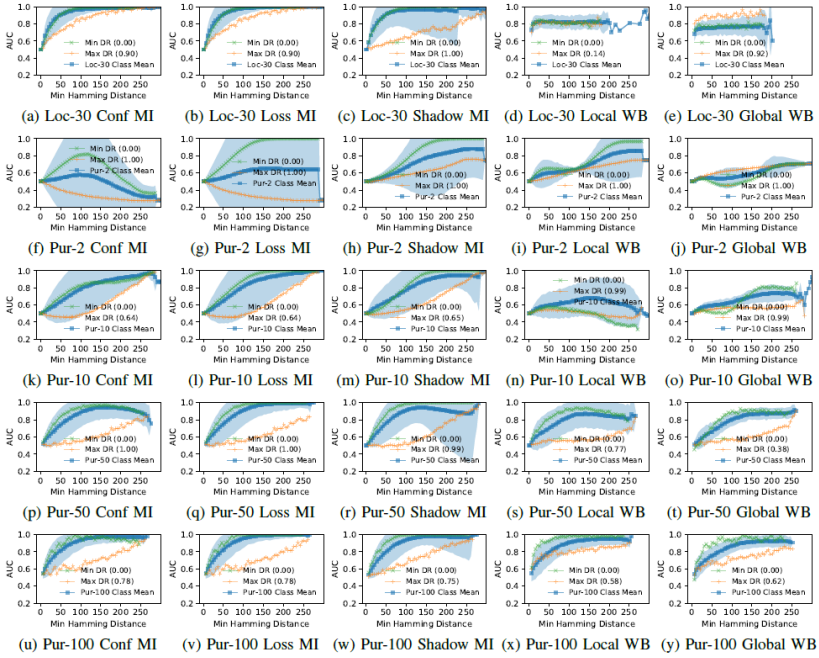


Fig. 8. Performance dell'attacco MI per classe (Min/Max DR) su Location, Purchase-2, Purchase-10, Purchase-50 e Purchase-100 per Conf MI, Loss MI, Shadow MI, Local WB MI, Global WB MI.

confermando che l'AUC per i non-membri sintetici è vicina a 0.5 a piccole distanze e migliora con l'aumentare della distanza.

Comportamento degli Attacchi di Inferenza di Attributi (AI) Gli attacchi AI mirano a inferire attributi mancanti di un record parzialmente noto. I risultati per il dataset Purchase mostrano un vantaggio trascurabile (vicino a zero) per gli attacchi AI, indipendentemente dalla variante del dataset e dall'attacco (Shadow AI, Loss AI, Conf AI). Ciò contrasta con studi precedenti sui modelli di regressione, ma gli autori sottolineano che i loro risultati si applicano ai problemi di classificazione. Anche per i modelli target più overfit, il vantaggio dell'AI rimane molto basso.

Comportamento degli Attacchi di Inferenza di Attributi Approssimata (AAI) L'AAI è una nozione più permissiva di AI, dove l'avversario deve trovare attributi vicini a quelli veri. Per il dataset Purchase, il parametro di distanza (alpha) è stato fissato a 7.5 (corrispondente alla distanza media di un'ipotesi casuale). I risultati mostrano che il vantaggio AAI è considerevolmente più alto rispetto al vantaggio AI, suggerendo che questi attacchi possono indovinare approssimativamente gli attributi mancanti con una probabilità superiore a una

supposizione casuale (es. per Purchase-100, il vantaggio AAI per Shadow AI è salito da 0.0042 a 0.0964).

– Correlazione con l'Overfitting: Esiste una correlazione positiva tra il livello di overfitting e il vantaggio AAI. Ad esempio, aumentando la dimensione del dataset di training per Purchase-100 da 20.000 a 200.000 record, l'overfitting diminuisce, e di conseguenza, il vantaggio AAI per Shadow AI si riduce da 0.118 a 0.026. L'AI esatta, al contrario, mostra un impatto minimo.

Validazione dell'Indistinguishable Neighbor Assumption. Per il dataset Purchase, l'assunto di vicini indistinguibili è stato convalidato sperimentalmente. L'addestramento di una GAN ha mostrato un vantaggio minimo nel distinguere un vettore reale da uno perturbato a piccole distanze, con un aumento significativo del vantaggio all'aumentare della distanza. Questo supporta l'ipotesi che piccole modifiche agli attributi non siano considerate anomalie significative.

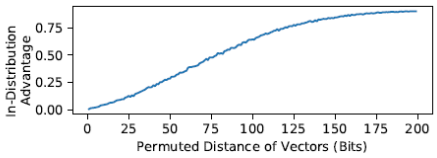


Fig. 9. Vantaggio in-distribuzione in funzione della distanza permutata di vettori per il dataset Purchase.

In sintesi, mentre gli attacchi di inferenza di appartenenza mostrano un successo variabile sul dataset Purchase (fallendo per i vicini prossimi), gli attacchi di inferenza esatta di attributi sono risultati inefficaci. Tuttavia, l'AAI ha dimostrato un potenziale significativo nell'indovinare gli attributi, e la sua efficacia è direttamente correlata al grado di overfitting del modello target.

8.1.3 Dataset CIFAR

Descrizione del Dataset CIFAR-100: Il dataset CIFAR-100 [5] è composto da 60.000 immagini a colori di 32×32 pixel, suddivise in 100 classi. Per gli esperimenti, sono stati selezionati casualmente 15.000 punti dal set completo. Le immagini sono state pre-elaborate tramite Principal Component Analysis (PCA), riducendo la dimensionalità a 50 feature continue, normalizzate tra -1 e 1. La distanza di Manhattan (d_M) è stata la metrica di distanza utilizzata, data la natura continua delle feature.

Modello Target e Preparazione dei Dati: Il modello target principale è una rete neurale (NN), in particolare un multilayer perceptron con due strati nascosti da 256 unità, funzione di attivazione ReLU e strato di output softmax. Questa

architettura replica configurazioni standard di riferimento. La suddivisione dei dati per gli attacchi è stata la seguente:

- Per Conf MI e Loss MI, sono stati campionati 50.000 record: 20% per il training del modello target e 80% per il testing.
- Per Shadow MI, è stato utilizzato l'intero dataset (circa 50.000 record): 20% per il training del modello target, 72% per i modelli shadow e 8% per il testing.

Il dataset CIFAR-100 ha mostrato un significativo livello di overfitting (scarto $\geq$0.8 tra accuratezza di training e test), correlato all'elevato numero di classi.

Attacchi di Inferenza di Attributi (AI e AAI): Per l'AAI su CIFAR-100, con 5 feature continue sconosciute, il parametro α è stato impostato a 3.33, che rappresenta la distanza media di una supposizione casuale dai valori originali.

Esiti degli Attacchi:

- **Membership Inference (MI):**
  - *Performance in funzione della Distanza:* L'AUC non ha mostrato un miglioramento marcato con l'aumento della distanza dei non-membri dal training set. Sui dati sintetici, l'AUC si è mantenuta vicina a 0.5 per i vettori prossimi al training set (distanza di Manhattan inferiore a 0.7-0.8), migliorando solo con l'aumentare della distanza, indicando un fallimento degli attacchi MI nel contesto di Strong Membership Inference (SMI) a piccole distanze.
  - *Confronto Conf MI vs. Loss MI:* Le curve AUC per Conf MI e Loss MI divergono a distanze di Manhattan maggiori ($\geq$0.7-0.8), con Loss MI che performa meglio quando il modello target produce previsioni errate.
  - *Accuratezza e Numero di Classi:* L'accuratezza degli attacchi MI migliora con un numero maggiore di classi nel modello target; CIFAR-100 mostra una maggiore accuratezza rispetto a CIFAR-20.

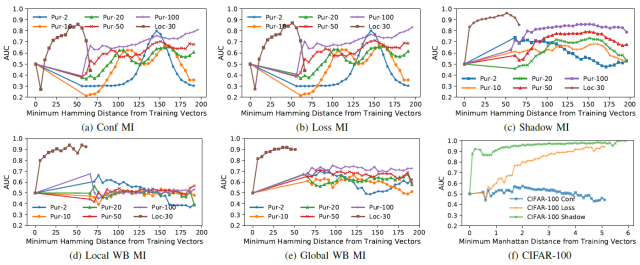


Fig. 10. Curva ROC degli attacchi Conf MI, Loss MI, Shadow MI, Local WB MI, Global WB MI su varianti del dataset Purchase e Loc-30. La parte (f) mostra i risultati su CIFAR-100.

- **Attribute Inference (AI - Esatta):**
  - *Vantaggio Trascurabile:* Sul CIFAR-100, gli attacchi AI hanno mostrato un vantaggio trascurabile (tra 4.14×10^{-7} e 7.99×10^{-4}), anche con modelli target altamente overfittati. Risulta difficile inferire attributi esatti da un modello di machine learning addestrato per compiti di classificazione, anche se il modello è vulnerabile all'inferenza di appartenenza.
- **Approximate Attribute Inference (AAI):**
  - *Vantaggio Significativo:* Il vantaggio AAI su CIFAR-100 (es. per Shadow AAI 0.0445) è stato notevolmente superiore rispetto all'AI esatta. Questo indica che gli attacchi possono indovinare approssimativamente gli attributi mancanti con una probabilità superiore a una supposizione casuale.
  - *Correlazione con Overfitting:* È stata riscontrata una correlazione positiva tra il livello di overfitting e il vantaggio AAI; una maggiore overfitting (ottenuta riducendo la dimensione del training set) porta a un aumento del vantaggio AAI su CIFAR-100.

In conclusione, mentre il dataset CIFAR-100 si è dimostrato vulnerabile agli attacchi di inferenza di appartenenza (specialmente in scenari di overfitting), la sua natura di problema di classificazione lo rende più resistente agli attacchi di inferenza di attributi esatta. Tuttavia, gli attacchi di inferenza di attributi approssimata possono ottenere un vantaggio significativo, la cui efficacia è amplificata dal grado di overfitting del modello target.

Descrizione del Dataset CIFAR-20: Il dataset CIFAR-20 [5] è una variante più ampia del CIFAR-100. Condividendo le caratteristiche di base relative alla preelaborazione (immagini elaborate tramite Principal Component Analysis (PCA) per 50 feature continue, normalizzate tra -1 e 1) e alla metrica di distanza (distanza di Manhattan, dM), si distingue per contenere 20 etichette di classe, che rappresentano un superset delle 100 classi di CIFAR-100 (es. "fiori" è il superset di orchidee, rose, ecc.). Per gli esperimenti, sono stati selezionati casualmente 15.000 punti dal set completo.

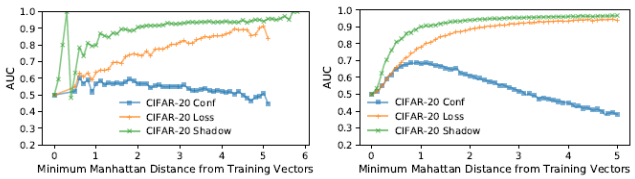
Modello Target e Preparazione dei Dati: La suddivisione dei dati per gli attacchi (Conf MI, Loss MI e Shadow MI) segue lo schema del CIFAR-100. Il divario di accuratezza tra training e test per CIFAR-20, pur non essendo grande come quello di CIFAR-100 ($\geq$0.8), è comunque presente, indicando un certo grado di overfitting.

Attacchi di Inferenza di Attributi (AI e AAI): Per l'AAI su CIFAR-20, il parametro α è stato impostato a 3.33, che rappresenta la distanza media di una supposizione casuale dai valori originali.

Esiti degli Attacchi:

- **Membership Inference (MI):**

- *Performance in funzione della Distanza:* Le curve AUC degli attacchi MI su CIFAR-20 (sia con vettori originali che sintetici) mostrano una tendenza simile a CIFAR-100: l'AUC è vicina a 0.5 (ipotesi casuale) per vettori vicini al training set (distanza di Manhattan inferiore a 0.2-0.5) e migliora solo con l'aumentare della distanza. Ciò implica che gli attacchi MI falliscono nel senso di Strong Membership Inference (SMI) a piccole distanze.



(a) Original vectors (b) Generated vectors

Fig. 11. Curva ROC degli attacchi Conf MI, Loss MI e Shadow MI su CIFAR-20 per vettori originali e generati.

- *Accuratezza e Numero di Classi:* Le curve AUC per CIFAR-20 sono leggermente inferiori rispetto a quelle di CIFAR-100, un risultato atteso data la riduzione del numero di classi. Questo conferma che l'accuratezza degli attacchi MI migliora con un numero maggiore di classi nel modello target.

– **Attribute Inference (AI - Esatta):**
  - *Vantaggio Trascurabile:* Per CIFAR-20, gli attacchi AI hanno mostrato un vantaggio trascurabile (tra -3.32×10^{-7} e 3.33×10^{-4}). Questo indica la difficoltà di inferire attributi esatti dal modello, anche se è suscettibile all'inferenza di appartenenza.
– **Approximate Attribute Inference (AAI):**
  - *Vantaggio Significativo:* Il vantaggio AAI su CIFAR-20 è considerabilmente più alto rispetto al vantaggio AI (es. 0.0339 per Shadow AAI). Ciò suggerisce che gli attacchi possono indovinare approssimativamente gli attributi mancanti con una probabilità migliore di una supposizione casuale. L'efficacia dell'AAI è correlata ai livelli di overfitting del modello target.

In sintesi, il dataset CIFAR-20 dimostra che, sebbene i modelli basati su di esso possano essere vulnerabili agli attacchi di inferenza di appartenenza (specialmente a grandi distanze), sono resilienti agli attacchi di inferenza di attributi esatta. Tuttavia, un avversario può ottenere un vantaggio significativo negli attacchi di inferenza di attributi approssimata, la cui efficacia è amplificata dai livelli di overfitting del modello target.

8.2 Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting

Questo lavoro si focalizza sulla relazione tra l'overfitting e la fuga di informazioni sensibili nei modelli di machine learning, esaminando come questi fattori influenzino gli attacchi di inferenza di appartenenza e di attributi.

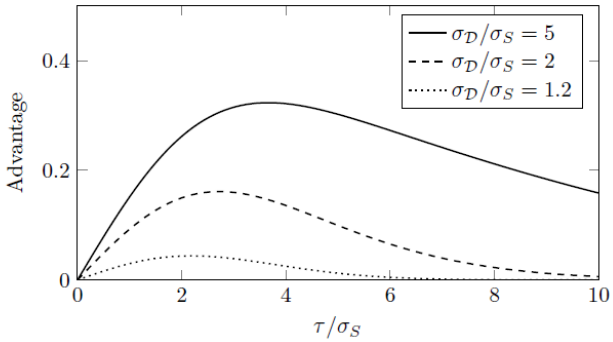


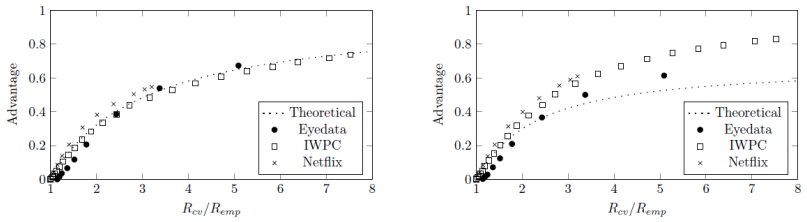
Fig. 12. Vantaggio teorico dell'attacco di inferenza di appartenenza in funzione della soglia τ/σ_S per diversi rapporti σ_D/σ_S.

8.2.1 Dataset Eyedata

Descrizione del Dataset: Il dataset Eyedata [10] consiste in dati di espressione genica provenienti da tessuti oculari di ratto, strutturati come una matrice di 120×200, dove 120 sono le osservazioni e 200 le feature. L'output è un vettore a 120 dimensioni di numeri in virgola mobile, corrispondente a ciascuna osservazione. Ogni attributo è stato scalato a media zero e varianza unitaria utilizzando la libreria scikit-learn di Python.

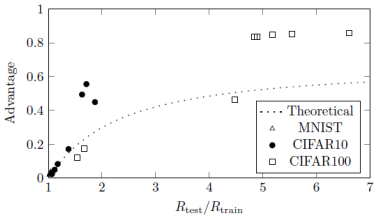
Metodologia Sperimentale: Negli esperimenti con modelli di regressione lineare (Ridge regression) e alberi decisionali, i dati sono stati divisi casualmente in set di training (75%) e test (25%). Questo processo è stato ripetuto 100 volte con diverse divisioni per mediare i risultati. Le metriche R_{emp} (errore empirico) e R_{cv} (errore di convalida incrociata leave-one-out) sono state impiegate per approssimare σ_S e σ_D rispettivamente, che rappresentano le deviazioni standard dell'errore del modello sui dati di training e test.

Esiti degli Attacchi di Inferenza di Appartenenza (MI): Lo studio ha valutato le performance degli attacchi di inferenza di appartenenza sul dataset Eyedata, utilizzando un avversario a soglia che presuppone la conoscenza della distribuzione di errore.



(a) Regression and tree models assuming knowledge of σ_S and $\sigma_\mathcal{D}$.

(b) Regression and tree models assuming knowledge of σ_S only.



(c) Deep CNNs assuming knowledge of average training loss L_S.

Fig. 13. Vantaggio degli attacchi di inferenza di appartenenza in funzione del rapporto $\mathrm{R}_{cv}/\mathrm{R}_{emp}$ o $\mathrm{R}_{test}/\mathrm{R}_{train}$. (a) Modelli di regressione e albero assumendo conoscenza di σ_S e σ_D. (b) Modelli di regressione e albero assumendo conoscenza solo di σ_S. (c) CNN profonde assumendo conoscenza della perdita media di training L_S.

- **Scenario con Conoscenza Completa (σ_S e σ_D):** Quando l'avversario possiede conoscenza sia di σ_S che di σ_D, i risultati sperimentali su Eyedata sono in linea con le previsioni teoriche per l'attacco di inferenza di appartenenza. Ciò supporta l'ipotesi che la relazione tra l'overfitting (misurato dal rapporto σ_D/σ_S o $\mathrm{R}_{cv}/\mathrm{R}_{emp}$) e il vantaggio dell'attacco sia valida in questo scenario ideale.
- **Scenario con Conoscenza Limitata (solo σ_S):** Sorprendentemente, quando l'avversario non conosce σ_D (la deviazione standard dell'errore sul set di test), le performance dell'attacco sono state molto migliori di quanto previsto dalla teoria. Questa discrepanza si spiega con il fatto che le distribuzioni di errore del set di training non sono perfettamente Gaussiane, ma mostrano un picco molto più alto a zero. Di conseguenza, impostare la soglia decisionale a $|\epsilon|{=}\sigma_S$ (come si farebbe senza conoscere σ_D) può rivelarsi più vantaggioso per l'attaccante.

Connessione con l'Overfitting: I risultati complessivi, supportati dai dati di Eyedata, ribadiscono che il comportamento di generalizzazione del modello, e in particolare l'overfitting, è un forte predittore della vulnerabilità agli attacchi di inferenza di appartenenza. Un maggiore overfitting (indicato da un rapporto $\mathrm{R}_{cv}/\mathrm{R}_{emp}$ più elevato) è direttamente correlato a un maggiore vantaggio per l'avversario. Le fonti non forniscono dettagli specifici sull'applicazione diretta

degli attacchi di inferenza di attributi con il dataset Eyedata. In sintesi, i modelli addestrati su Eyedata, in particolare Ridge regression e alberi decisionali, hanno mostrato una chiara connessione tra l'overfitting e la vulnerabilità agli attacchi di inferenza di appartenenza. La performance di questi attacchi ha confermato le previsioni teoriche in condizioni di conoscenza completa dell'errore, ma ha superato le aspettative teoriche quando la conoscenza era limitata alla sola deviazione standard del training set, evidenziando il ruolo delle distribuzioni di errore non perfettamente Gaussiane.

8.2.2 Dataset IWPC

Descrizione del Dataset: Il dataset IWPC [1] (International Warfarin Pharmacogenetics Consortium) emerge come uno strumento cardine per l'esplorazione empirica della connessione intrinseca tra il fenomeno dell'overfitting e il rischio di privacy nei modelli di machine learning. Originariamente comprendente dati di pazienti a cui è stata prescritta la warfarina, il set finale di 4819 record è stato ottenuto dopo un'accurata rimozione delle righe con valori mancanti. Struttura e Pre-elaborazione del Dataset. Gli attributi di input si suddividono in categorie distinte:

- Demografici: età, altezza, peso, trattati come valori reali e scalati a media zero e varianza unitaria.
- Medici: uso di amiodarone e di induttore enzimatico, presentati come attributi binari.
- Genetici: VKORC1 e CYP2C9, sottoposti a codifica one-hot encoding.

L'output del modello è rappresentato dalla dose settimanale di warfarina in milligrammi. Per ovviare all'asimmetria nella distribuzione di questa dose, è stata applicata una trasformazione tramite radice quadrata, che ha reso il modello lineare più predittivo, seguendo una raccomandazione consolidata in letteratura. Anche la radice quadrata della dose è stata successivamente scalata a media zero e varianza unitaria.

Metodologia di Addestramento e Misurazione: Il dataset IWPC è stato impiegato per l'addestramento di modelli di regressione lineare (Ridge regression) e di alberi decisionali. La robustezza dei risultati è stata assicurata attraverso una metodologia rigorosa: i dati sono stati suddivisi in set di training e test con un rapporto 75%-25% rispettivamente, e questo processo è stato replicato 100 volte con diverse partizioni casuali, per poi calcolare una media dei risultati. Per la quantificazione dell'errore e del grado di overfitting, sono state calcolate due metriche fondamentali: l'errore empirico (R_{emp}) e l'errore di convalida incrociata leave-one-out (R_{cv}). Queste metriche sono state utilizzate per approssimare, rispettivamente, le deviazioni standard dell'errore del modello sui dati di training (σ_S) e sui dati di test (σ_D).

Esiti degli Attacchi di Inferenza di Appartenenza (MI): L'analisi delle performance degli attacchi di inferenza di appartenenza sul dataset IWPC ha rivelato dinamiche interessanti:

- **Conoscenza Completa dell'Avversario (σ_S e σ_D):** Quando l'avversario dispone la conoscenza sia della deviazione standard dell'errore sul set di training (σ_S) sia su quello di test (σ_D), i risultati sperimentali sul dataset IWPC si allineano con le previsioni teoriche. Ciò conferma che, in questo scenario ideale, la relazione tra overfitting (quantificato dal rapporto σ_D/σ_S o $\mathrm{R}_{cv}/\mathrm{R}_{emp}$) e il vantaggio dell'attacco è conforme alle aspettative.
- **Conoscenza Limitata dell'Avversario (solo σ_S):** Al contrario, se l'avversario non conosce σ_D, le performance dell'attacco di inferenza di appartenenza sul dataset IWPC hanno superato significativamente le previsioni teoriche. Questa discrepanza è attribuita alla natura non perfettamente Gaussiana delle distribuzioni di errore del training set, che esibiscono un picco più accentuato a zero. Tale conformazione rende vantaggioso per l'avversario fissare la soglia decisionale a $|\epsilon|{=}\sigma_S$, migliorando l'efficacia dell'attacco oltre quanto teoricamente atteso.

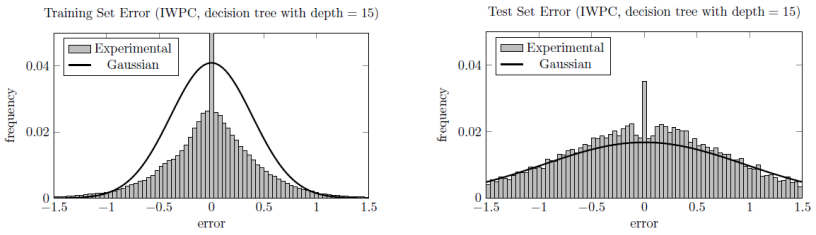
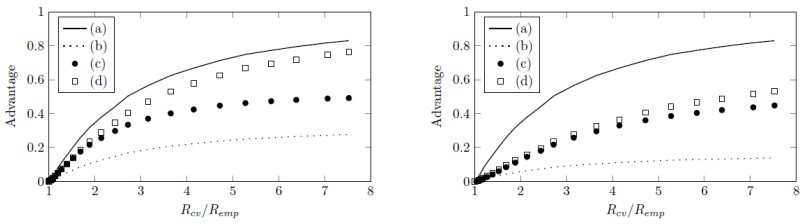


Fig. 14. Distribuzioni dell'errore del training set e del test set per IWPC (albero decisionale con profondità = 15), confrontate con una distribuzione Gaussiana.

Esiti degli Attacchi di Inferenza di Attributi (AI): Lo studio ha esplorato l'attacco di inferenza di attributi generale (Adversary 4) nel contesto dell'inversione di modello, focalizzandosi sugli attributi genetici VKORC1 e CYP2C9 come target. L'attacco su IWPC ha manifestato un vantaggio considerevole, il quale è cresciuto in proporzione all'aumento dell'overfitting del modello. Sebbene questo vantaggio sia risultato inferiore rispetto all'inferenza di appartenenza diretta, la sua entità è rimasta comunque significativa.

Attacchi di Riduzione: L'analisi ha incluso anche la valutazione dell'efficacia dell'avversario di riduzione a query multiple (Adversary 7). Sui dati IWPC, l'applicazione di query multiple all'oracolo di appartenenza ha condotto a un aumento notevole del tasso di successo in confronto all'avversario di riduzione



(a) t = VKORC1 (b) t = CYP2C9

Fig. 15. Vantaggio degli attacchi di inferenza di attributi sui dataset IWPC per gli attributi target VKORC1 (a) e CYP2C9 (b) in funzione di Rcv/R_{emp}.

uniforme (Adversary 6). Sorprendentemente, questa strategia di riduzione si è dimostrata più efficace dell'esecuzione diretta dell'attacco di inferenza di attributi. Ciò suggerisce che, per il dataset IWPC, la trasformazione di un attacco di inferenza di appartenenza in uno di inferenza di attributi tramite riduzione può configurarsi come una tattica più potente e fruttuosa. In conclusione, il dataset IWPC, con la sua ricchezza di dati demografici, medici e genetici, ha fornito una base empirica solida per delineare la relazione tra overfitting e il rischio di privacy, in particolare per gli attacchi di inferenza di appartenenza e attributi. Lo studio ha inoltre messo in luce come le assunzioni sulla distribuzione degli errori possano influenzare criticamente le prestazioni degli attacchi e come determinate tecniche di riduzione possano amplificare l'efficacia degli attacchi di inferenza di attributi.

8.2.3 Dataset Netflix

Descrizione del Dataset: Il dataset Netflix, originariamente proveniente dal concorso Netflix Prize, costituisce un esempio significativo per l'analisi del rischio di privacy nei modelli di machine learning, in particolare per comprendere le dinamiche degli attacchi di inferenza su dati utente. Si tratta di un dataset sparso che registra le valutazioni assegnate dagli utenti ai film. Struttura e Preelaborazione del Dataset. Il dataset si concentra su 2416 utenti con valutazioni specifiche:

- Attributo di Output (Variabile Dipendente): È stata utilizzata la valutazione del film "Dragon Ball Z: Trunks Saga", scelto per la sua distribuzione di valutazione particolarmente polarizzata. Queste valutazioni sono state scalate a media zero e varianza unitaria.
- Attributi di Input (Feature): Le feature sono variabili binarie, indicanti se un utente ha valutato o meno ciascuno degli altri 17.769 film presenti nel dataset.

Metodologia di Addestramento e Misurazione: Il dataset Netflix è stato impiegato per addestrare modelli di regressione lineare (Ridge regression) e di alberi

decisionali. La metodologia sperimentale ha previsto una suddivisione casuale dei dati in set di training (75%) e test (25%). Questo processo è stato replicato 100 volte con diverse partizioni, e i risultati ottenuti sono stati mediati per garantirne la robustezza. Per la quantificazione dell'errore, sono state calcolate l'errore empirico (R_{emp}) e l'errore di convalida incrociata leave-one-out (R_{cv}), utilizzate per approssimare rispettivamente le deviazioni standard dell'errore del modello sui dati di training (σ_S) e sui dati di test (σ_D). Per gli attacchi di inferenza di attributi (Adversary 4), l'attributo target è stato definito come l'atto di un utente di aver valutato un film specifico, campionando casualmente tra i film disponibili.

Esiti degli Attacchi di Inferenza:

- **Membership Inference (MI):** I risultati degli attacchi di inferenza di appartenenza (Adversary 2) su Netflix mostrano un comportamento analogo a quello osservato su altri dataset. Quando l'avversario ha conoscenza sia di σ_S che di σ_D, i risultati sperimentali si allineano con le previsioni teoriche. Tuttavia, in assenza della conoscenza di σ_D, le performance dell'attacco superano significativamente le previsioni teoriche, a causa della natura non perfettamente Gaussiana delle distribuzioni di errore del training set, che presentano un picco più accentuato a zero.

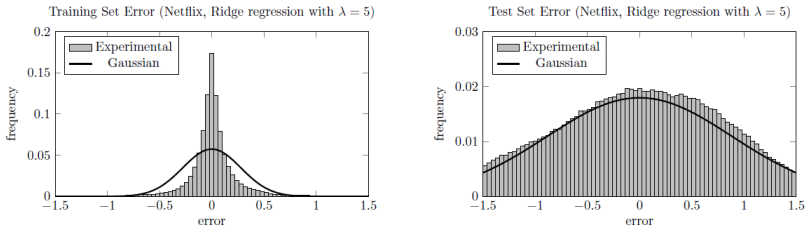


Fig. 16. Distribuzioni dell'errore del training set e del test set per Netflix (Ridge regression con λ=5), confrontate con una distribuzione Gaussiana.

- **Attribute Inference (AI - Adversary 4):** A differenza di quanto osservato con il dataset IWPC, sul dataset Netflix nessuno degli attacchi AI è riuscito a inferire efficacemente se un utente avesse valutato un determinato film. Questo indica che l'attacco di inferenza di attributi diretto non ha mostrato un vantaggio significativo, né un aumento in correlazione con l'overfitting. Non è stato possibile, inoltre, controllare simultaneamente l'effetto del rapporto σ_D/σ_S e dell'influenza (τ) per misurare l'effetto dell'influenza come previsto dal Teorema 5.
- **Attacchi di Riduzione (Adversary 7):** Con i dati Netflix, l'avversario di riduzione a query multiple (Adversary 7) si è rivelato spesso leggermente meno efficace dell'avversario di riduzione uniforme (Adversary 6). Nonostante ciò, ha comunque superato la performance dell'inferenza diretta di

attributi. Questo risultato contrasta con quello ottenuto sul dataset IWPC, dove l'Adversary 7 ha migliorato significativamente il tasso di successo, suggerendo che le dinamiche degli attacchi di riduzione possono variare notevolmente a seconda delle caratteristiche intrinseche del dataset.

In sintesi, il dataset Netflix ha permesso di approfondire la comprensione del rischio di privacy, mostrando vulnerabilità agli attacchi di inferenza di appartenenza ma una resilienza notevole agli attacchi di inferenza di attributi diretti. Ha inoltre evidenziato come l'efficacia delle strategie di riduzione possa essere specifica per il dataset, con risultati divergenti rispetto ad altri contesti.

8.2.4 Dataset MNIST

Descrizione del Dataset: Il dataset MNIST [6] è una collezione di immagini di cifre scritte a mano, in scala di grigi e con una dimensione di 28×28 pixel. Ogni immagine è associata a un'etichetta di classe che indica la cifra rappresentata. Il dataset completo di MNIST contiene 70.000 immagini, da cui è stato selezionato casualmente un sottoinsieme di 17.500 punti per gli esperimenti.

Pre-elaborazione e Utilizzo negli Esperimenti: I valori dei pixel delle immagini sono stati normalizzati nell'intervallo [0,1], e le etichette di classe sono state codificate come vettori one-hot. Il dataset MNIST è stato impiegato per addestrare reti neurali convoluzionali (CNNs), la cui architettura si basava sulla rete VGG, con un parametro di dimensione s variabile per definire il numero di unità in ogni strato. L'addestramento è stato condotto utilizzando l'ottimizzatore Adam e la funzione di perdita categorical cross-entropy. I dati disponibili (il sottoinsieme di 17.500 punti) sono stati divisi casualmente in set di training e test di dimensioni uguali, per facilitare il confronto con lavori precedenti. Per la misurazione dell'errore, lo studio ha utilizzato l'errore di generalizzazione, calcolato come la differenza tra l'accuratezza sul training e sul test set.

Comportamento dei Modelli e Risultati degli Attacchi:

- **Attacchi di Inferenza di Appartenenza (MI):**
  - *Attacco Basato sulla Soglia (Adversary 2):* I risultati dell'attacco di inferenza di appartenenza basato sulla soglia su CNNs addestrate su MNIST hanno mostrato che, nonostante l'errore non sia Gaussiano, i risultati empirici non divergono significativamente dalla curva teorica. L'attacco su MNIST ha raggiunto una precisione di 0.505 e un recall superiore a 0.99. Queste prestazioni sono state comparabili, sebbene leggermente inferiori in precisione, rispetto a un attacco più complesso basato su "shadow models" (Shokri et al.), ma con requisiti computazionali e di conoscenza preliminare notevolmente inferiori.
  - *Attacchi di Collusione per Inferenza di Appartenenza:* MNIST è stato utilizzato anche per valutare l'algoritmo di addestramento collusivo (Algorithm 1) e l'avversario collusivo (Adversary 3). Gli esperimenti hanno

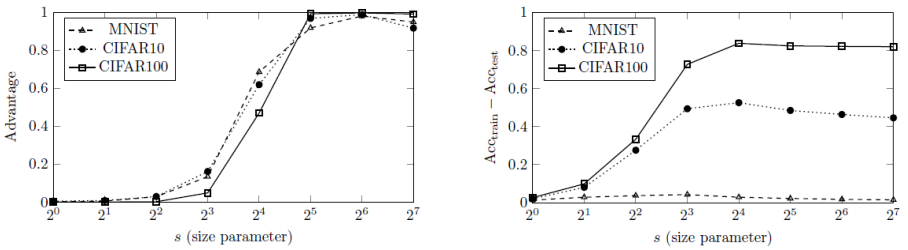


Fig. 17. Vantaggio degli attacchi di inferenza di appartenenza (sinistra) e errore di generalizzazione (destra) in funzione del parametro di dimensione s per MNIST, CIFAR-10 e CIFAR-100.

dimostrato che le parti collusive hanno ottenuto un elevato vantaggio nell'inferenza di appartenenza (almeno 0.9 per s≥16) senza compromettere significativamente le prestazioni del modello (l'accuratezza del modello sovvertito era solo 0.014 inferiore). Sorprendentemente, per i modelli MNIST, non è stata riscontrata una relazione discernibile tra l'errore di generalizzazione e il vantaggio nell'inferenza di appartenenza. I modelli MNIST hanno mostrato un errore di generalizzazione quasi nullo (<0.02 per s≥32), eppure l'avversario di appartenenza ha raggiunto prestazioni quasi identiche. Questo suggerisce che la capacità del modello di "memorizzare" il dataset, dovuta a un numero di parametri che supera significativamente la dimensione del training set, può avere gravi implicazioni per la privacy, anche in assenza di overfitting misurabile.

8.2.5 Dataset CIFAR

Descrizione del Dataset CIFAR-100: Il dataset CIFAR-100 [5] si presenta come un banco di prova cruciale per dissezionare le vulnerabilità dei modelli di machine learning, in particolare le reti neurali convoluzionali profonde (CNNs), in relazione agli attacchi di inferenza di appartenenza (membership inference). La sua struttura e il suo impiego negli esperimenti consentono di illuminare dinamiche complesse legate alla "memorizzazione" dei dati e agli scenari di collusione. Struttura e Pre-elaborazione del Dataset. CIFAR-100 è una raccolta di 60.000 immagini a colori, ciascuna di 32×32 pixel, categorizzate in 100 classi distinte. Per gli scopi sperimentali, è stato selezionato un sottoinsieme casuale di 15.000 immagini, ottimizzando così i tempi di addestramento e facilitando la comparazione con lavori precedenti. La fase di pre-elaborazione ha comportato la normalizzazione dei valori dei pixel e la codifica delle etichette di classe in vettori one-hot, procedure standard per l'addestramento di reti neurali.

Configurazione Sperimentale e Modelli Target: I modelli target impiegati sono CNNs profonde, la cui architettura si ispira alla rinomata rete VGG. Un parametro

di dimensione, s (con valori 2^i per $0 \leq i \leq 7$), ha modulato il numero di unità in ciascun livello della rete. La struttura prevedeva: due strati convoluzionali ha modulato il numero di unità in ciascun livello della rete. La struttura prevedeva: due strati convoluzionali 3×3 (con s filtri), seguiti da un livello di max pooling 2×2; ulteriori due strati convoluzionali 3×3 (con 2s filtri) e un secondo livello di max pooling 2×2; un livello completamente connesso con 2s unità; e, in conclusione, un livello di output softmax. Tutte le funzioni di attivazione erano di tipo rettificato lineare. L'addestramento è stato condotto con l'ottimizzatore Adam (utilizzando i parametri predefiniti) e la funzione di perdita categorical cross-entropy, scelta convenzionale per modelli con attivazione softmax. I dati disponibili sono stati equamente divisi tra set di training e test per uniformità comparativa con la ricerca preesistente. La generalizzazione del modello, un indicatore cruciale del suo apprendimento, è stata misurata come la differenza tra l'accuratezza sul training set (Acc train) e l'accuratezza sul test set (Acc test). Questo divario riflette quanto il modello sia in grado di estendere le conoscenze acquisite a dati non visti.

Comportamento dei Modelli e Risultati degli Attacchi:

- **Attacchi di Inferenza di Appartenenza (Membership Inference - MI):**
  - *Attacco Basato sulla Soglia (Adversary 2):* L'attacco basato sulla soglia (Adversary 2), applicato alle CNNs addestrate su CIFAR-100, ha rivelato performance notevoli. Nonostante il modello operi in un contesto di classificazione e utilizzi una funzione di perdita (categorical cross-entropy) che non aderisce all'assunto Gaussiano del Teorema 3, i risultati empirici su CIFAR-100 non hanno manifestato una divergenza dalla curva teorica così ampia come ci si sarebbe potuti attendere. L'attacco ha conseguito una precisione di 0.874 e un recall superiore a 0.99. Queste metriche, sebbene la precisione fosse leggermente inferiore rispetto a un attacco più complesso basato su "shadow models" (con precisione >0.99 su CIFAR-100) di Shokri et al., si sono dimostrate comparabili in termini di recall (>0.99). È degno di nota che l'approccio adottato in questo studio ha richiesto risorse computazionali e conoscenze preliminari significativamente inferiori.
  - *Attacchi di Collusione per Inferenza di Appartenenza:* Il dataset CIFAR-100 è stato altresì impiegato per valutare gli effetti di un algoritmo di addestramento collusivo (Algorithm 1) e di un avversario collusivo (Adversary 3), con un numero di chiavi (k) fissato a 3. Gli esperimenti hanno evidenziato che le parti collusive hanno ottenuto un elevatissimo vantaggio nell'inferenza di appartenenza (almeno 0.98) per valori di s sufficientemente grandi (es. $s \geq 16$). L'impatto sulle prestazioni del modello è risultato minimo, con un'accuratezza del modello sovvertito di soli 0.031 inferiore rispetto a quello non sovvertito.
  - *Differenze con l'Overfitting:* A differenza del comportamento osservato con il dataset MNIST, per CIFAR-100 si è riscontrato un significativo

divario tra l'accuratezza di training e test (ovvero, un errore di generalizzazione pronunciato), con un massimo di 0.82 per s=16. Ciononostante, e qui risiede un'osservazione cruciale, nonostante questo ampio "gap di performance" o errore di generalizzazione, l'avversario di inferenza di appartenenza ha conseguito prestazioni quasi identiche. Questo suggerisce inequivocabilmente che la capacità intrinseca di un modello di "memorizzare" il dataset, spesso correlata a un numero di parametri che eccede significativamente la dimensione del training set, può avere gravi implicazioni per la privacy, indipendentemente dalla misurabilità o meno dell'overfitting.

Descrizione del Dataset CIFAR-10: Il dataset CIFAR-10 [5], analogo al CIFAR-100 per la sua composizione di immagini a colori di 32×32 pixel e la sua origine (60.000 immagini totali), si distingue principalmente per la presenza di 10 classi differenti. Per gli esperimenti, è stato selezionato un sottoinsieme casuale di 15.000 immagini, come avvenuto anche per il CIFAR-100, per ottimizzare i tempi di addestramento e facilitare le comparazioni.

Pre-elaborazione e Configurazione Sperimentale: La pre-elaborazione dei dati ha seguito le procedure standard per le reti neurali, ovvero la normalizzazione dei valori dei pixel e la codifica delle etichette di classe in vettori one-hot, in linea con quanto applicato al CIFAR-100. I modelli target impiegati sono CNNs profonde, con un'architettura basata sulla rete VGG, modulata da un parametro di dimensione s (con valori 2^i per $0 \leq i \leq 7$). La configurazione degli strati convoluzionali, di max pooling, completamente connessi e di output softmax, così come l'uso di funzioni di attivazione rettificate lineari, dell'ottimizzatore Adam e della funzione di perdita categorical cross-entropy, è identica a quella impiegata per il CIFAR-100. Anche la divisione dei 15.000 punti disponibili in set di training e test di dimensioni uguali è stata mantenuta per coerenza comparativa. La generalizzazione del modello è stata valutata attraverso la differenza tra l'accuratezza sul training set (Acc train) e sul test set (Acc test), una misura comune per quantificare l'apprendimento del modello su dati non visti.

Comportamento dei Modelli e Risultati degli Attacchi:

- **Attacchi di Inferenza di Appartenenza (Membership Inference - MI):**
  - *Attacco Basato sulla Soglia (Adversary 2):* L'attacco basato sulla soglia (Adversary 2), applicato alle CNNs addestrate su CIFAR-10, ha mostrato un comportamento coerente con quello osservato sul CIFAR-100: i risultati empirici, nonostante l'errore non sia strettamente Gaussiano, non hanno manifestato una divergenza significativa dalla curva teorica.
  - *Differenze nei risultati:* Sebbene l'approccio sia lo stesso e richieda meno risorse rispetto agli "shadow models" di Shokri et al., l'attacco su CIFAR-10 ha conseguito una precisione di 0.694 (inferiore rispetto al 0.874 del

CIFAR-100), mantenendo un recall superiore a 0.99 (identico al CIFAR-100). La precisione di Shokri et al. su CIFAR-10 si attestava tra 0.72 e 0.74.

- *Attacchi di Collusione per Inferenza di Appartenenza:* Anche per CIFAR-10, sono stati valutati gli effetti dell'algoritmo di addestramento collusivo (Algorithm 1) e dell'avversario collusivo (Adversary 3), con k=3 chiavi. Gli esperimenti hanno dimostrato che le parti collusive hanno ottenuto un elevato vantaggio nell'inferenza di appartenenza (almeno 0.9 per s$\geq$16) senza compromettere significativamente le prestazioni del modello (l'accuratezza del modello sovvertito di soli 0.047 inferiore rispetto a quello non sovvertito (un impatto leggermente superiore rispetto al 0.031 del CIFAR-100).
- *Differenze con l'Overfitting:* A differenza del comportamento del CIFAR-100, che mostrava un errore di generalizzazione pronunciato (massimo 0.82), per CIFAR-10 è stata osservata una significativa differenza tra l'accuratezza di training e test (un errore di generalizzazione pronunciato), con un massimo di 0.52 per s=16. Nonostante questo errore di generalizzazione sia inferiore a quello del CIFAR-100, l'attacco di inferenza di appartenenza ha comunque raggiunto un vantaggio molto alto. Questo rafforza l'idea che la capacità intrinseca di un modello di "memorizzare" il dataset, spesso correlata a un numero di parametri che eccede significativamente la dimensione del training set, possa portare a gravi implicazioni per la privacy, indipendentemente dalla misurabilità o meno dell'overfitting come grande errore di generalizzazione. Questo fenomeno, in cui un modello può "memorizzare" i dati pur mostrando una minore discrepanza tra training e test rispetto a un modello più overfit, sottolinea come la vulnerabilità alla privacy non sia esclusivamente legata all'overfitting classico, ma anche alla capacità di memorizzazione latente del modello.

9 Main Idea

Uno degli obiettivi principali del nostro lavoro è stato quello di indagare un'apparente contraddizione emersa da due studi di riferimento che abbiamo analizzato. Entrambi i paper affrontano attacchi alla privacy nei confronti di modelli di machine learning, in particolare **Membership Inference Attacks** (**MIA**) e **Attribute Inference Attacks** (**AIA**). Tuttavia, mentre il primo studio, focalizzato su modelli di regressione, sostiene che l'overfitting del modello aumenti significativamente la vulnerabilità sia a MIA che ad AIA, il secondo, concentrato su modelli di classificazione, afferma che, pur in presenza di overfitting e di una marcata esposizione al MIA, l'efficacia degli AIA rimane negligibile.

Questa divergenza nei risultati ha motivato la nostra indagine sperimentale: ci siamo proposti di verificare quale delle due ipotesi fosse più fondata. La nostra assunzione iniziale era che l'overfitting costituisse una condizione favorevole alla perdita di privacy, indipendentemente dalla natura del compito (regressione o classificazione), rendendo quindi plausibili entrambi gli attacchi.

Per testare questa ipotesi, abbiamo replicato, nei limiti delle risorse disponibili, alcune delle sperimentazioni proposte nei due lavori. Poiché l'utilizzo delle piattaforme cloud descritte negli articoli comportava costi non sostenibili, abbiamo optato per implementare modelli personalizzati sulla piattaforma **Colab**, sia per il task di regressione che per quello di classificazione (**clicca qui per andare al codice completo**).

Nel caso della regressione, abbiamo costruito due modelli distinti:

- uno deliberatamente progettato per overfittare (basato su una rete neurale, in quanto tale architettura tende facilmente all'overfitting se non vengono sfruttate tecniche di regolarizzazione),

- e uno progettato per generalizzare correttamente (utilizzando una regressione Ridge, che grazie alla regolarizzazione mitiga il rischio di overfitting).

Anche per la classificazione, abbiamo adottato un approccio simile, creando due reti neurali con diverse configurazioni, in linea con quanto riportato nel secondo paper, per analizzare le implicazioni dell'overfitting sia sul MIA che sull'AIA.
Queste scelte ci hanno permesso di effettuare un confronto coerente e sistematico con i risultati riportati in letteratura.

10 Approch

Come prima cosa, abbiamo scelto quali dataset utilizzare e la scelta è ricaduta sul dataset **IWPC 2009** per quanto riguarda la creazione dei due modelli di **Regressione**, mentre su **CIFAR100** (disponibile all'interno della librari Python **torchvision**) per i modelli di **Classificazione**.

10.1 Implementazione e Risultati Sperimentali: Modelli di Regressione

In questa sezione andremo a descrive l'implementazione dei modelli di regressione e l'esecuzione Membership Inference Attack e Attribute Inference Attack su di essi. Verranno presentati il codice utilizzato e i risultati ottenuti, analizzando la relazione tra overfitting e la suscettibilità a tali attacchi.

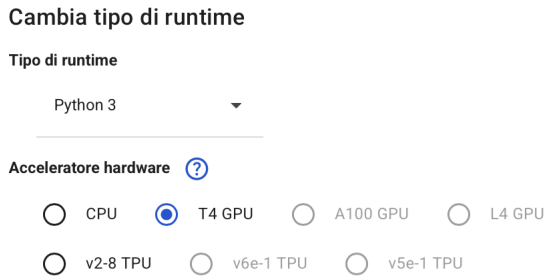
Preparazione dei Dati e Costruzione dei Modelli di Regressione
Per i modelli di regressione, è stato utilizzato il dataset IWPC 2009 con l'obiettivo di predire la "**Therapeutic Dose of Warfarin**". Sono stati sviluppati due modelli: una Rete Neurale Profonda progettata per overfittare e un modello di Ridge Regression, noto per la sua capacità di generalizzare grazie alla regolarizzazione L2.

Codice per i Modelli di Regressione

Il seguente codice Python illustra le fasi di caricamento e pre-elaborazione del dataset, la definizione e l'addestramento dei due modelli, e una prima valutazione delle loro performance.

Nota bene: prima di eseguire il seguente codice è necessario:

– connettersi al runtime selezionando "Python 3" come tipo di runtime e "T4 GPU" come acceleratore harware;



– caricare il dataset in formato "*xlsx*" direttamente all'interno del notebook di Colab all'interno della cartella "**sample_data**";

Creazione dei modelli di Regressione

```
1
2  import pandas as pd
3  import numpy as np
4  from sklearn.model_selection import train_test_split
5  from sklearn.preprocessing import StandardScaler
6  from sklearn.linear_model import Ridge
7  from sklearn.metrics import mean_squared_error
8  import matplotlib.pyplot as plt
9  from tensorflow import keras
10 from tensorflow.keras import layers
11 from tensorflow.keras import regularizers
12 from tensorflow.keras.callbacks import EarlyStopping
13
14 df = pd.read_excel('/content/sample_data/IWPC_2009.xlsx',
   ↪  sheet_name='IWPC 2009')
15
16 print("Colonne disponibili nel dataset:")
```

```
print(df.columns.tolist())

target_col = 'Therapeutic Dose of Warfarin'

X = df.drop(columns=[target_col])
y = df[target_col]

X = pd.get_dummies(X, drop_first=True)

X = X.fillna(X.mean())
y = y.fillna(y.mean())

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42
)

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Modello 1: Rete Neurale Profonda (overfitting)

# Architettura: 4 layer densi con attivazione ReLU.
model_overfit = keras.Sequential([
    layers.Dense(512, activation='relu',
    ↪  input_shape=(X_train_scaled.shape[1],)),
    layers.Dense(256, activation='relu'),
    layers.Dense(128, activation='relu'),
    layers.Dense(1)
])

model_overfit.compile(optimizer='adam', loss='mse')
history_overfit = model_overfit.fit(
    X_train_scaled, y_train,
    epochs=200,
    batch_size=32,
    validation_data=(X_test_scaled, y_test),
    verbose=0
)


```

```
# Modello 2: Ridge Regression (No Overfitting)

ridge = Ridge(alpha=100000.0)
ridge.fit(X_train_scaled, y_train)

# Valutazione dei Modelli

y_train_pred_overfit = model_overfit.predict(X_train_scaled).flatten()
y_test_pred_overfit = model_overfit.predict(X_test_scaled).flatten()
mse_train_overfit = mean_squared_error(y_train, y_train_pred_overfit)
mse_test_overfit = mean_squared_error(y_test, y_test_pred_overfit)

y_train_pred_ridge = ridge.predict(X_train_scaled)
y_test_pred_ridge = ridge.predict(X_test_scaled)
mse_train_ridge = mean_squared_error(y_train, y_train_pred_ridge)
mse_test_ridge = mean_squared_error(y_test, y_test_pred_ridge)

print("\n=== Modello OVERFITTING (Rete Neurale Profonda) ===")
print(f"MSE Train: {mse_train_overfit:.2f}")
print(f"MSE Test:  {mse_test_overfit:.2f}")

print("\n=== Modello NO OVERFITTING (Ridge) ===")
print(f"MSE Train: {mse_train_ridge:.2f}")
print(f"MSE Test:  {mse_test_ridge:.2f}")

plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
plt.plot(history_overfit.history['loss'], label='Train Loss')
plt.plot(history_overfit.history['val_loss'], label='Test Loss')
plt.title('Learning Curve - Modello Overfitting')
plt.xlabel('Epoch')
plt.ylabel('MSE')
plt.legend()

plt.subplot(1, 2, 2)
plt.bar(['Train', 'Test'], [mse_train_ridge, mse_test_ridge])
plt.title('MSE - Modello Ridge')
plt.ylabel('MSE')
plt.show()
```

Risultati della Valutazione dei Modelli

La valutazione dei modelli ha confermato il comportamento atteso in termini di overfitting e generalizzazione.
Come si può osservare dai risultati, il modello con Rete Neurale presenta un MSE di training molto basso (0.72) e un MSE di test significativamente più alto (1114.93), indicando un chiaro caso di overfitting. La curva di apprendimento per la Rete Neurale mostra come la "Train Loss" diminuisca costantemente mentre la "Test Loss" aumenta o si stabilizza su un valore elevato, segno che il modello ha memorizzato i dati di training senza generalizzare bene a nuovi dati.
Al contrario, il modello di Ridge Regression mostra un MSE di training (191.71) e un MSE di test (255.94) che sono relativamente vicini tra loro, suggerendo una buona capacità di generalizzazione e l'assenza di overfitting significativo.

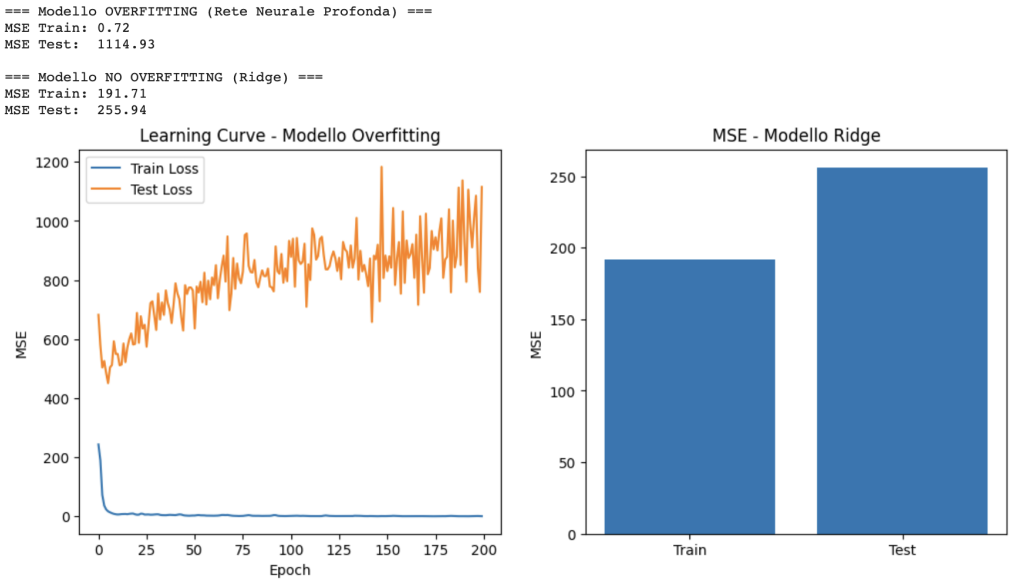


Fig. 18. MSE nei due modelli

10.2 Membership Inference Attack (MIA) sulla Regressione

Il **Membership Inferece Attack** mira a determinare se un particolare record di dati è stato utilizzato nel training set del modello. Questo attacco si basa sull'osservazione che i modelli overfittati tendono a produrre loss inferiori per i campioni visti durante l'addestramento rispetto a quelli non visti.

Codice per il Membership Inference Attack

Il codice seguente implementa il **loss–based Membership Inference Attack** proposto da Yeom *et al.* ([13], §3.2, *Adversary 2 – Threshold*). In questo scenario *black-box* l'avversario osserva solo la predizione $\hat{y} = f_\theta(x)$, calcola per ciascun campione la loss individuale $\ell(x, y) = (y - \hat{y})^2$ e addestra un regressore logistico a distinguere *membri* da *non-membri*. L'efficacia dell'attacco è valutata tramite curva ROC e AUC: valori vicini a 1 denotano forte overfitting (quindi alto rischio di privacy), mentre AUC ≈ 0.5 indica assenza di segnale sfruttabile.

Membership Inference Attack (Regression)

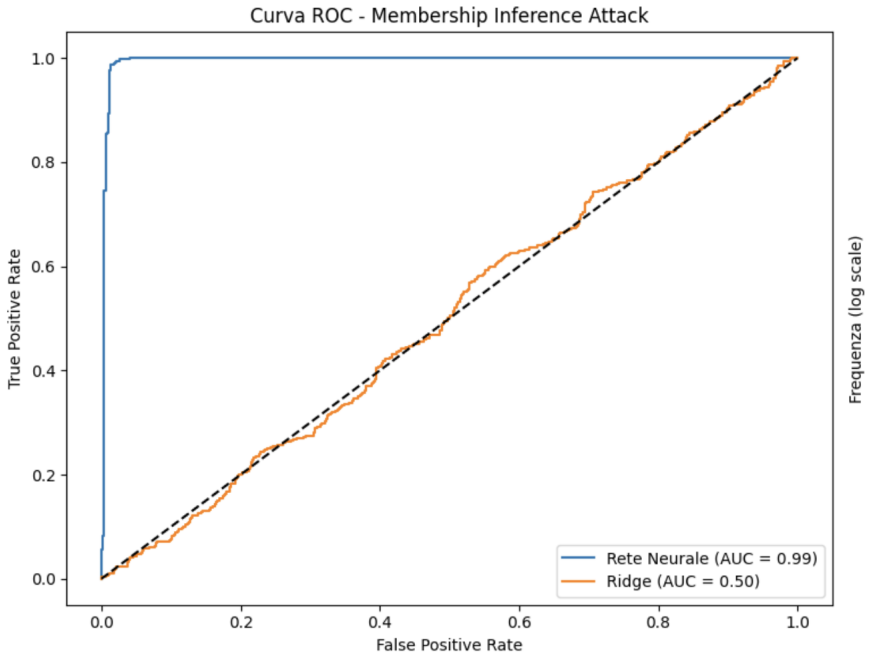
```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import roc_auc_score, roc_curve


def compute_mia(model, X_train, X_test, y_train, y_test):
    """
    Esegue un Membership Inference Attack basato sulla loss individuale.
    Restituisce AUC e dati per le curve ROC.
    """
    # Calcolo delle predizioni
    y_pred_train = model.predict(X_train).flatten()
    y_pred_test = model.predict(X_test).flatten()

    # Calcolo della loss per ogni campione (errore quadratico)
    loss_train = (y_train - y_pred_train) ** 2
    loss_test = (y_test - y_pred_test) ** 2

    # Creazione del dataset per l'attacco
    df_mia = pd.DataFrame({
        'loss': np.concatenate([loss_train, loss_test]),
        'member': np.concatenate([np.ones_like(loss_train), np.zeros_like(loss_test)])
    })

    # Split in train/test per il modello di attacco
    X_attack = df_mia[['loss']]
    y_attack = df_mia['member']
    X_train_attack, X_test_attack, y_train_attack, y_test_attack = train_test_split(
```

```
        X_attack, y_attack, test_size=0.3, random_state=42
    )

    # Addestramento del classificatore (Regressore Logistico)
    clf = LogisticRegression()
    clf.fit(X_train_attack, y_train_attack)

    # Predizione delle probabilità
    y_proba = clf.predict_proba(X_test_attack)[:, 1]

    auc = roc_auc_score(y_test_attack, y_proba)
    fpr, tpr, thresholds = roc_curve(y_test_attack, y_proba)

    return auc, fpr, tpr, loss_train, loss_test

# Attacco sul modello in overfitting (rete neurale)
auc_nn, fpr_nn, tpr_nn, loss_train_nn, loss_test_nn = compute_mia(
    model_overfit, X_train_scaled, X_test_scaled, y_train.values,
    ↪ y_test.values
)

# Attacco sul modello Ridge (no overfitting)
auc_ridge, fpr_ridge, tpr_ridge, loss_train_ridge, loss_test_ridge =
↪ compute_mia(
    ridge, X_train_scaled, X_test_scaled, y_train.values, y_test.values
)

plt.figure(figsize=(15, 6))

plt.subplot(1, 2, 1)
plt.plot(fpr_nn, tpr_nn, label=f'Rete Neurale (AUC = {auc_nn:.2f})')
plt.plot(fpr_ridge, tpr_ridge, label=f'Ridge (AUC = {auc_ridge:.2f})')
plt.plot([0, 1], [0, 1], 'k--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Curva ROC - Membership Inference Attack')
plt.legend()

plt.subplot(1, 2, 2)
plt.hist(loss_train_nn, bins=50, alpha=0.5, label='Train (NN)')
plt.hist(loss_test_nn, bins=50, alpha=0.5, label='Test (NN)')
```

```
plt.hist(loss_train_ridge, bins=50, alpha=0.5, label='Train (Ridge)')
plt.hist(loss_test_ridge, bins=50, alpha=0.5, label='Test (Ridge)')
plt.yscale('log')
plt.xlabel('Loss (MSE individuale)')
plt.ylabel('Frequenza (log scale)')
plt.title('Distribuzione delle Loss')
plt.legend()

plt.tight_layout()
plt.show()

print("\n=== Risultati Membership Inference Attack ===")
print(f"Modello Overfitting (Rete Neurale):")
print(f"- AUC: {auc_nn:.2f}")
print(f"- Gap medio loss (Train-Test): {np.mean(loss_train_nn) -
↪  np.mean(loss_test_nn):.2f}")

print(f"\nModello Ridge (No Overfitting):")
print(f"- AUC: {auc_ridge:.2f}")
print(f"- Gap medio loss (Train-Test): {np.mean(loss_train_ridge) -
↪  np.mean(loss_test_ridge):.2f}")
```

Risultati del Membership Inference Attack

I risultati dell'attacco MIA sono stati analizzati attraverso le curve ROC e la distribuzione delle loss.




```
=== Risultati Membership Inference Attack ===
Modello Overfitting (Rete Neurale):
- AUC: 0.99
- Gap medio loss (Train-Test): -1114.21

Modello Ridge (No Overfitting):
- AUC: 0.50
- Gap medio loss (Train-Test): -64.23
```

Fig. 19. Efficacia del MIA sui modelli di regressione

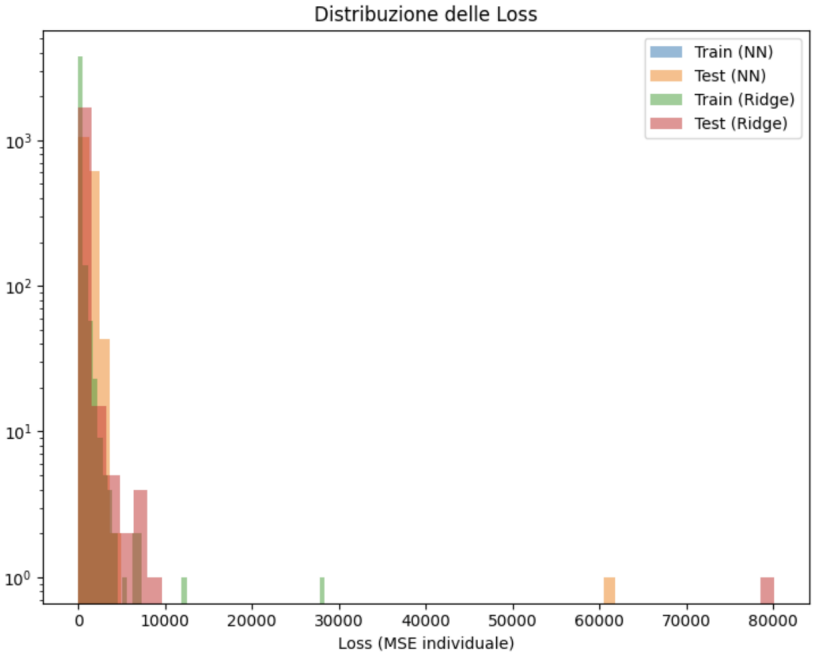


Fig. 20. Distribuzione delle loss sui modelli di regressione

Come si evince dai grafici e dai valori AUC:

Rete Neurale (Overfitting): L'AUC (Area Under the Curve) per il modello overfittato è di 0.99. Un valore di AUC così elevato indica che l'attaccante può distinguere con altissima precisione se un campione è stato o meno parte del training set, basandosi sulla sua loss. Questo è ulteriormente supportato dal significativo "gap medio loss" tra i set di training e test (-1116.21), che mostra come i campioni di training abbiano una loss drasticamente inferiore rispetto ai campioni di test. La distribuzione delle loss (grafico a destra) mostra due distribuzioni ben separate per i dati di training e test del modello overfittato.

Modello Ridge (No Overfitting): L'AUC per il modello Ridge è di 0.50, che è equivalente a una classificazione casuale. Questo suggerisce che l'attaccante non è in grado di distinguere i membri dai non membri. Il "gap medio loss" (-64.23) è molto più piccolo rispetto al modello overfittato, indicando una minore differenza tra le loss dei campioni di training e test. La distribuzione delle loss per il modello Ridge mostra una maggiore sovrapposizione tra le loss dei campioni di training e test.

Questi risultati rafforzano l'ipotesi che l'overfitting aumenti significativamente la vulnerabilità di un modello agli attacchi di tipo Membership Inference.

10.2.1 Confronto qualitativo. Il comportamento osservato è coerente con quello descritto da Yeom *et al.* [13]: i modelli che presentano overfitting con-

sentono all'attaccante di distinguere con grande facilità i campioni di training, mentre la presenza di una forte regolarizzazione riporta l'attacco a prestazioni puramente casuali.

10.3 Attribute Inference Attack (AIA) sulla Regressione

L'attacco di inferenza di attributi mira a inferire il valore di un attributo sensibile (ad esempi 'Gender' nel nostro caso) di un record, anche se tale attributo non è esplicitamente l'obiettivo del modello principale, sfruttando le sue predizioni.

Codice per l'Attribute Inference Attack

Il codice seguente implementa un **Attribute Inference Attack** diretto, come descritto da Yeom *et al.* [13]. L'attacco assume che l'avversario abbia accesso a un insieme di campioni completi, inclusi gli attributi sensibili, e sfrutti l'output del modello target $\hat{y} = f_\theta(x)$ per inferire un attributo mancante (nel nostro caso, `Gender`). Per farlo, si costruisce un modello supervisionato che riceve come input le feature non sensibili e la predizione del modello, e viene addestrato a ricostruire l'attributo nascosto. Questo approccio è noto come *direct attribute inference* e dimostra sperimentalmente una forte correlazione con l'overfitting del modello target.

Attribute Inference Attack (Regression)

```

from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score,
↪  recall_score, confusion_matrix

# 2. Preparazione dati per l'attacco
sensitive_attr = 'Gender'

# Ricostruzione X completo (con Gender incluso) e y
X_full = df.drop(columns=[target_col])
y_full = df[target_col]

# Rimuove i valori mancanti
X_full = pd.get_dummies(X_full, drop_first=True)
X_full = X_full.fillna(X_full.mean(numeric_only=True))
y_full = y_full.fillna(y_full.mean())

# Recupera la colonna Gender originale
```

```
sensitive = df[sensitive_attr].fillna(df[sensitive_attr].mode().iloc[0])

# Divisione coerente in train/test
X_train, X_test, y_train, y_test, s_train, s_test = train_test_split(
    X_full, y_full, sensitive,
    test_size=0.3,
    random_state=42,
    stratify=sensitive
)

# Standardizzazione coerente con i modelli già addestrati
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled  = scaler.transform(X_test)

# 3. Funzione di Attribute Inference Attack
def attribute_inference_attack_gender(target_model, X_tr, X_te,
↪  y_sens_tr, y_sens_te):
    # 3.1. Predizioni del modello target
    pred_tr = target_model.predict(X_tr).flatten()
    pred_te = target_model.predict(X_te).flatten()

    # 3.2. Costruzione delle feature di attacco
    attack_tr = np.column_stack([X_tr, pred_tr])
    attack_te = np.column_stack([X_te, pred_te])

    # 3.3. Codifica dell'attributo sensibile (es. Gender)
    le = LabelEncoder()
    y_tr_enc = le.fit_transform(y_sens_tr)
    y_te_enc = le.transform(y_sens_te)

    # 3.4. Addestramento attaccante
    attacker = LogisticRegression(max_iter=1000)
    attacker.fit(attack_tr, y_tr_enc)

    # 3.5. Valutazione
    y_tr_pred = attacker.predict(attack_tr)
    y_te_pred = attacker.predict(attack_te)

    acc_tr = accuracy_score(y_tr_enc, y_tr_pred)
    acc_te = accuracy_score(y_te_enc, y_te_pred)
```

```
    prec   = precision_score(y_te_enc, y_te_pred)
    rec    = recall_score(y_te_enc, y_te_pred)

    return {
        'acc_train': acc_tr,
        'acc_test':  acc_te,
        'precision': prec,
        'recall':    rec,
        'y_true':    y_sens_te.values,
        'y_pred':    le.inverse_transform(y_te_pred)
    }

# 4. Esecuzione dell'attacco sui due modelli
res_nn    = attribute_inference_attack_gender(model_overfit,
↪  X_train_scaled, X_test_scaled, s_train, s_test)
res_ridge = attribute_inference_attack_gender(ridge, X_train_scaled,
↪  X_test_scaled, s_train, s_test)

print("=== Attribute Inference Attack su Gender ===\n")

print(">> Rete Neurale (Overfitting)")
print(f"- Accuracy Train: {res_nn['acc_train']:.2f}")
print(f"- Accuracy Test:  {res_nn['acc_test']:.2f}")
print(f"- Precision (Test): {res_nn['precision']:.2f}")
print(f"- Recall    (Test): {res_nn['recall']:.2f}\n")

print(">> Ridge Regression (No Overfitting)")
print(f"- Accuracy Train: {res_ridge['acc_train']:.2f}")
print(f"- Accuracy Test:  {res_ridge['acc_test']:.2f}")
print(f"- Precision (Test): {res_ridge['precision']:.2f}")
print(f"- Recall    (Test): {res_ridge['recall']:.2f}\n")

cm = confusion_matrix(res_nn['y_true'], res_nn['y_pred'],
↪  labels=['female','male'])
print("Confusion Matrix (NN - Test):")
print(pd.DataFrame(cm, index=['Vero Female','Vero Male'], columns=['Pred
↪  Female','Pred Male']))
```

Risultati dell'Attribute Inference Attack

Di seguito i risultati dell'attacco di inferenza di attributi per entrambi i modelli.

```
=== Attribute Inference Attack su Gender ===

>> Rete Neurale (Overfitting)
- Accuracy Train: 1.00
- Accuracy Test:  0.94
- Precision (Test): 0.91
- Recall    (Test): 1.00

>> Ridge Regression (No Overfitting)
- Accuracy Train: 1.00
- Accuracy Test:  0.98
- Precision (Test): 0.97
- Recall    (Test): 1.00

Confusion Matrix (NN - Test):
             Pred Female  Pred Male
Vero Female          619         93
Vero Male              2        996
```

Fig. 21. Efficacia del AIA sui modelli di regressione

Analizzando i risultati dell'attacco di inferenza sull'attributo '**Gender**':

Rete Neurale (Overfitting): L'attaccante ha raggiunto un'Accuracy di test di 0.94, con una Precision di 0.91 e un Recall di 1.00. Questi valori elevati indicano che il modello overfittato, pur essendo di regressione, ha indirettamente "memorizzato" informazioni sull'attributo sensibile, rendendolo vulnerabile a questo tipo di attacco. La Confusion Matrix mostra che l'attaccante ha avuto successo nel predire correttamente la maggior parte dei campioni, con pochi falsi negativi per 'male' e alcuni falsi positivi per 'female'.

Ridge Regression (No Overfitting): L'attaccante ha ottenuto un'Accuracy di test leggermente più alta, pari a 0.98, con Precision di 0.97 e Recall di 1.00. Sebbene l'accuracy sia elevata per entrambi, la Rete Neurale mostra una precisione inferiore, che potrebbe indicare una maggiore tendenza a classificare erroneamente un sesso rispetto all'altro, suggerendo una potenziale correlazione tra l'overfitting e la "perdita" di specifici attributi, anche se non direttamente evidente solo dall'accuracy in questo caso. È importante notare che, in contesti di regressione, l'inferenza di attributi può essere più complessa, e l'efficacia dell'attacco può dipendere da quanto l'attributo sensibile è correlato alla variabile target o ad altre feature.

Questi risultati suggeriscono che, anche per i modelli di regressione, un modello che overfitta potrebbe avere una "memoria" più forte di attributi specifici nel dataset di training, anche se l'attacco di inferenza di attributi non mostra

sempre una differenza così marcata come il MIA, dipendendo dalla correlazione tra l'attributo sensibile e la funzione del modello.

10.3.1 Confronto qualitativo. Dal confronto tra i due modelli emerge che l'Attribute Inference Attack ha avuto successo sia sul modello overfittato (rete neurale) sia su quello regolarizzato (Ridge). Tuttavia, come osservato da Yeom *et al.* [13], l'overfitting non è una condizione necessaria per la riuscita dell'attacco. Nel nostro caso, nonostante la rete neurale abbia effettivamente "memorizzato" più informazione, il modello Ridge ha comunque mostrato performance migliori nell'inferenza dell'attributo.

Questo suggerisce che l'overfitting può amplificare la vulnerabilità, ma non è il fattore determinante: l'attributo `Gender` risulta inferibile anche da un modello ben generalizzato, se è sufficientemente correlato alla variabile target o ad altre feature. L'effetto dell'overfitting è quindi visibile, ma secondario rispetto alla presenza di informazione strutturale nei dati.

10.4 Implementazione e Risultati Sperimentali: Modelli di Classificazione

Questa sezione estende l'analisi ai modelli di classificazione, seguendo la stessa metodologia dei modelli di regressione. Verranno presentati due modelli (uno in overfitting e uno generalizzato) e l'applicazione degli attacchi di Membership Inference e Attribute Inference su di essi.

Preparazione dei Dati e Costruzione dei Modelli di Classificazione
Per i modelli di classificazione, è stato utilizzato il dataset CIFAR100. Sono stati sviluppati due modelli di rete neurale convoluzionale: il primo (OverfitNet) progettato per overfittare intenzionalmente, e il secondo (NoOverfitNet) con tecniche di regolarizzazione per promuovere una migliore generalizzazione.

Codice per i Modelli di Classificazione
Il seguente codice Python, basato su PyTorch, illustra la preparazione del dataset CIFAR100, la definizione delle due architetture di rete neurale e il loro addestramento.

Creazione dei modelli di Classificazione

```
1
2  import torch
3  import torch.nn as nn
4  import torch.optim as optim
5  import torchvision
6  import torchvision.transforms as transforms
7  import matplotlib.pyplot as plt
8
```

```
# Configurazione iperparametri
batch_size = 32  # Dimensione batch ridotta per limitare uso memoria GPU
num_epochs = 100
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")  # Usa GPU se disponibile

# Spiegazione:
# batch_size=32 riduce il carico di memoria rispetto a un valore pi alto

transform_train_overfit = transforms.Compose([
    transforms.ToTensor(),  # Converte immagini in tensori e normalizza in [0,1]
])

transform_test = transforms.Compose([
    transforms.ToTensor(),
])

# Caricamento dataset CIFAR100
trainset_overfit = torchvision.datasets.CIFAR100(
    root='./data',
    train=True,
    download=True,
    transform=transform_train_overfit
)

testset = torchvision.datasets.CIFAR100(
    root='./data',
    train=False,
    download=True,
    transform=transform_test
)

# Creazione DataLoader per il training
trainloader_overfit = torch.utils.data.DataLoader(
    trainset_overfit,
    batch_size=batch_size,
    shuffle=True,   # Mescola i dati ad ogni epoca
    num_workers=2   # Thread paralleli per il caricamento
```

```
)

testloader = torch.utils.data.DataLoader(
    testset,
    batch_size=batch_size,
    shuffle=False,  # Non necessario per il test
    num_workers=2
)

# shuffle=True previene l'apprendimento di pattern legati all'ordinamento dei dati

class OverfitNet(nn.Module):
    def __init__(self):
        super(OverfitNet, self).__init__()
        # Blocco feature extraction
        self.features = nn.Sequential(
            # Layer 1: 3 canali input (RGB), 64 filtri 3x3, padding per mantenere dimensione
            nn.Conv2d(3, 64, 3, padding=1),
            nn.ReLU(),  # Funzione di attivazione

            # Max pooling riduce dimensione spaziale a 16x16
            nn.MaxPool2d(2),

            # Layer 2: 64 -> 128 canali
            nn.Conv2d(64, 128, 3, padding=1),
            nn.ReLU(),
            nn.MaxPool2d(2),  # 8x8

            # Layer 3: 128 -> 256 canali
            nn.Conv2d(128, 256, 3, padding=1),
            nn.ReLU(),
            nn.MaxPool2d(2),  # 4x4

            nn.Flatten()  # Appiattisce l'output per i layer fully-connected
        )

        self.classifier = nn.Sequential(
            # 256 canali * 4x4 = 4096 features
```

```
            nn.Linear(256*4*4, 512),
            nn.ReLU(),
            # Layer finale: 512 features -> 100 classi (CIFAR100)
            nn.Linear(512, 100)
        )

    def forward(self, x):
        x = self.features(x)  # Passaggio attraverso i layer convoluzionali
        x = self.classifier(x)  # Passaggio attraverso i layer fully-connected
        return x

model = OverfitNet().to(device)  # Sposta il modello sulla GPU
criterion = nn.CrossEntropyLoss()  # Loss function per classificazione
optimizer = optim.Adam(model.parameters(), lr=0.001)  # Ottimizzatore con learning rate

train_acc_list = []
test_acc_list = []

for epoch in range(num_epochs):
    # Fase di training

    # Punti cruciali del training loop:
    # optimizer.zero_grad(): Resetta i gradienti per evitare accumuli
    # loss.backward(): Calcola gradienti tramite autograd
    # optimizer.step(): Aggiorna i pesi usando gli ottimizzatori

    model.train() # Imposta il modello in modalità training
    correct_train, total_train = 0, 0
    for inputs, labels in trainloader_overfit:
        # Sposta dati su GPU
        inputs, labels = inputs.to(device), labels.to(device)
        # Azzera i gradienti accumulati
        optimizer.zero_grad()
        # Forward pass
        outputs = model(inputs)
        # Calcola loss
        loss = criterion(outputs, labels)
        # Backpropagation
```

```
        loss.backward() # Calcola gradienti
        # Aggiorna pesi
        optimizer.step()
        # Calcolo accuratezza
        _, predicted = outputs.max(1) # Indici delle classi predette
        total_train += labels.size(0)
        correct_train += predicted.eq(labels).sum().item()
    train_acc = correct_train / total_train
    train_acc_list.append(train_acc)

    # Fase di valutazione

    # Modalità evaluation:

    model.eval() # Disabilita dropout/batchnorm
    correct_test, total_test = 0, 0
    with torch.no_grad(): #  Disabilita calcolo gradienti per
    ↪  risparmiare memoria
        for inputs, labels in testloader:
            inputs, labels = inputs.to(device), labels.to(device)
            outputs = model(inputs)
            _, predicted = outputs.max(1)
            total_test += labels.size(0)
            correct_test += predicted.eq(labels).sum().item()
    test_acc = correct_test / total_test
    test_acc_list.append(test_acc)

    # Stampa risultati epoca
    print(f"Epoch {epoch+1}/{num_epochs} - Train acc: {train_acc:.4f} -
    ↪  Test acc: {test_acc:.4f}")

    # Pulizia memoria CUDA
    torch.cuda.empty_cache()

# Plot accuracy
plt.figure(figsize=(8,5))
plt.plot(range(1, num_epochs+1), train_acc_list, label='Train Accuracy')
plt.plot(range(1, num_epochs+1), test_acc_list, label='Test Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.title('OverfitNet Accuracy on CIFAR100')
```

```
plt.legend()
plt.grid(True)
plt.show()

# Trasformazioni per il training che aiutano la generalizzazione (data augmentation)
transform_train_no_overfit = transforms.Compose([
    transforms.RandomHorizontalFlip(),
    transforms.RandomCrop(32, padding=4),
    transforms.ToTensor(),
])

# Dataset di training con data augmentation per il modello no overfit
trainset_no_overfit = torchvision.datasets.CIFAR100(
    root='./data',
    train=True,
    download=True,
    transform=transform_train_no_overfit
)

trainloader_no_overfit = torch.utils.data.DataLoader(
    trainset_no_overfit,
    batch_size=batch_size,
    shuffle=True,
    num_workers=2
)

class NoOverfitNet(nn.Module):
    def __init__(self):
        super(NoOverfitNet, self).__init__()
        self.features = nn.Sequential(
            nn.Conv2d(3, 32, 3, padding=1), nn.BatchNorm2d(32), nn.ReLU(),
            nn.MaxPool2d(2),  # 32x32 -> 16x16
            nn.Conv2d(32, 64, 3, padding=1), nn.BatchNorm2d(64), nn.ReLU(),
            nn.MaxPool2d(2),  # 16x16 -> 8x8
            nn.Conv2d(64, 128, 3, padding=1), nn.BatchNorm2d(128), nn.ReLU(),
            nn.MaxPool2d(2),  # 8x8 -> 4x4
            nn.Flatten()
```

```
        )
        self.classifier = nn.Sequential(
            nn.Dropout(0.5),  # Regolarizzazione
            nn.Linear(128*4*4, 128), nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear(128, 100)
        )

    def forward(self, x):
        x = self.features(x)
        x = self.classifier(x)
        return x

no_overfit_model = NoOverfitNet().to(device)
criterion_no = nn.CrossEntropyLoss()
optimizer_no = optim.Adam(no_overfit_model.parameters(), lr=0.001,
↪  weight_decay=1e-4)

train_acc_list_no = []
test_acc_list_no = []

for epoch in range(num_epochs):
    # Training
    no_overfit_model.train()
    correct_train, total_train = 0, 0
    for inputs, labels in trainloader_no_overfit:
        inputs, labels = inputs.to(device), labels.to(device)
        optimizer_no.zero_grad()
        outputs = no_overfit_model(inputs)
        loss = criterion_no(outputs, labels)
        loss.backward()
        optimizer_no.step()
        _, predicted = outputs.max(1)
        total_train += labels.size(0)
        correct_train += predicted.eq(labels).sum().item()
    train_acc = correct_train / total_train
    train_acc_list_no.append(train_acc)

    # Valutazione
    no_overfit_model.eval()
    correct_test, total_test = 0, 0
```

```
    with torch.no_grad():
        for inputs, labels in testloader:
            inputs, labels = inputs.to(device), labels.to(device)
            outputs = no_overfit_model(inputs)
            _, predicted = outputs.max(1)
            total_test += labels.size(0)
            correct_test += predicted.eq(labels).sum().item()
    test_acc = correct_test / total_test
    test_acc_list_no.append(test_acc)

    print(f"[NO OVERFIT] Epoch {epoch+1}/{num_epochs} - Train acc:
    ↪  {train_acc:.4f} - Test acc: {test_acc:.4f}")

    torch.cuda.empty_cache()

plt.figure(figsize=(10,5))
plt.plot(range(1, num_epochs+1), train_acc_list, label='OverfitNet -
↪  Train')
plt.plot(range(1, num_epochs+1), test_acc_list, label='OverfitNet -
↪  Test')
plt.plot(range(1, num_epochs+1), train_acc_list_no, label='NoOverfitNet
↪  - Train')
plt.plot(range(1, num_epochs+1), test_acc_list_no, label='NoOverfitNet -
↪  Test')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.title('Confronto Overfitting vs No Overfitting su CIFAR100')
plt.legend()
plt.grid(True)
plt.show()
```

Risultati della Valutazione dei Modelli di Classificazione:

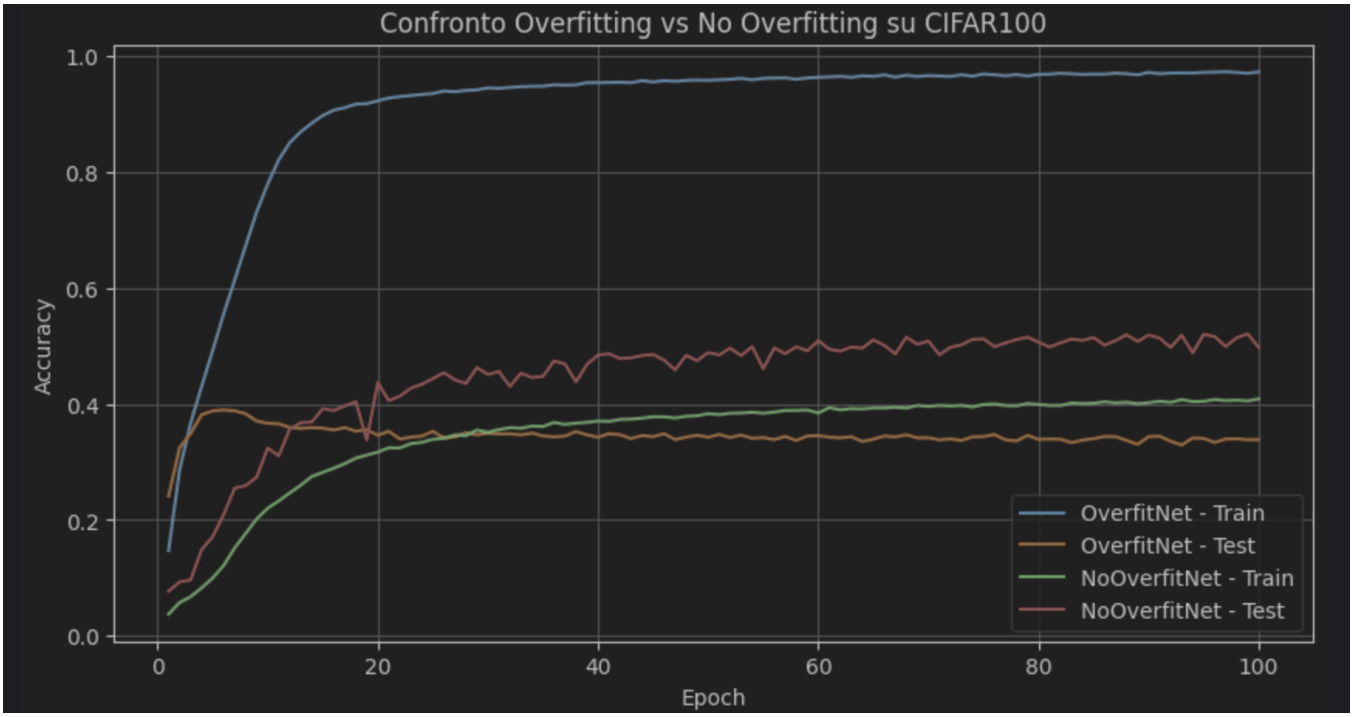


Fig. 22. Accuratezza dei modelli di classificazione

La valutazione dei modelli ha mostrato chiare differenze nel comportamento di overfitting e generalizzazione.

OverfitNet: Come previsto, questo modello mostra un grave caso di overfitting. L'accuratezza sul training set raggiunge quasi il 100% (circa 0.97), mentre l'accuratezza sul test set si attesta intorno al 33%-34%. Questo ampio divario tra le performance di training e test è un classico indicatore di overfitting, dove il modello ha memorizzato i dati di training ma non è in grado di generalizzare a nuovi dati non visti.

NoOverfitNet: Questo modello, grazie all'uso di tecniche di regolarizzazione come BatchNorm2d, Dropout e weight_decay (regolarizzazione L2), mostra un comportamento molto più bilanciato. Sebbene l'accuratezza sul training set sia inferiore rispetto a OverfitNet (circa 0.40), l'accuratezza sul test set è molto più vicina, aggirandosi intorno al 0.50-0.51. Questo indica una migliore generalizzazione e una minore tendenza all'overfitting.

Il grafico di confronto illustra chiaramente il divario tra le curve di train e test accuracy per **OverfitNet**, mentre per **NoOverfitNet** le due curve sono molto più ravvicinate, a dimostrazione di una migliore capacità di generalizzazione.

10.5 Membership Inference Attack (MIA) sulla Classificazione

Analogamente ai modelli di regressione, l'attacco di inferenza di appartenenza sui modelli di classificazione valuta la capacità di un attaccante di determinare se un campione è stato incluso nel training set, basandosi sulla confidenza o sulla probabilità che il modello assegna alla classe vera del campione.

Codice per il Membership Inference Attack (Classificazione)

Il codice cerca di implementare il **Confidence-based Membership Inference Attack**, come descritto da Zhao *et al.* [15]. L'attaccante osserva, per ogni campione, la probabilità che il modello assegna alla classe corretta. Poiché i modelli tendono ad essere più confidenti sui dati di training rispetto a quelli mai visti, questa confidenza viene utilizzata come segnale per distinguere i membri dai non membri.

Membership Inference Attack (Classificazione)

```

import torch
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve, auc, confusion_matrix,
↪  accuracy_score

def get_probs_pytorch(model, inputs, device):
    model.eval()
    with torch.no_grad():
        inputs = inputs.to(device)
        outputs = model(inputs)
        probs = torch.softmax(outputs, dim=1) # Calcola le probabilit
        ↪  con softmax
    return probs.cpu().numpy()

def membership_inference_attack_pytorch(model, trainloader, testloader,
↪  device, n_samples=2000):
    model.eval()
    # Estrai n_samples dal train e dal test
    X_train, y_train = [], []
```

```
    for inputs, labels in trainloader:
        X_train.append(inputs)
        y_train.append(labels)
        if len(torch.cat(y_train)) >= n_samples:
            break
    X_train = torch.cat(X_train)[:n_samples]
    y_train = torch.cat(y_train)[:n_samples]

    X_test, y_test = [], []
    for inputs, labels in testloader:
        X_test.append(inputs)
        y_test.append(labels)
        if len(torch.cat(y_test)) >= n_samples:
            break
    X_test = torch.cat(X_test)[:n_samples]
    y_test = torch.cat(y_test)[:n_samples]

    # Calcola la probabilità della classe vera
    probs_train = get_probs_pytorch(model, X_train, device)
    probs_test = get_probs_pytorch(model, X_test, device)
    # Le "scores" sono le probabilit assegnate dal modello alla classe vera di ciascun campione.
    # I modelli overfittati tendono ad assegnare probabilit pi alte ai campioni di training.
    scores_train = probs_train[np.arange(len(y_train)), y_train.numpy()]
    scores_test = probs_test[np.arange(len(y_test)), y_test.numpy()]

    # Combina scores e etichette di appartenenza (1 per membro, 0 per non membro)
    scores = np.concatenate([scores_train, scores_test])
    labels = np.concatenate([np.ones_like(scores_train), np.zeros_like(scores_test)])

    # Calcola la curva ROC e l'AUC
    fpr, tpr, thresholds = roc_curve(labels, scores)
    roc_auc = auc(fpr, tpr)
    best_thresh = thresholds[np.argmax(tpr - fpr)] # Trova la soglia ottimale
    preds = (scores >= best_thresh).astype(int) # Classificazioni basate sulla soglia

```

```
    acc = accuracy_score(labels, preds)
    cm = confusion_matrix(labels, preds)

    print(f"Membership Inference - Accuracy: {acc:.3f}, AUC:
    ↪ {roc_auc:.3f}")
    print("Confusion Matrix:\n", cm)
    plt.figure()
    plt.plot(fpr, tpr, label=f'ROC curve (area = {roc_auc:.2f})')
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('Membership Inference ROC')
    plt.legend()
    plt.show()
    return acc, roc_auc, cm

# Esecuzione dell'attacco sui due modelli di classificazione
print("MIA - Overfit")
membership_inference_attack_pytorch(model, trainloader_overfit,
↪ testloader, device)

print("MIA - No Overfit")
membership_inference_attack_pytorch(no_overfit_model,
↪ trainloader_no_overfit, testloader, device)
```

Risultati del Membership Inference Attack (Classificazione)

Di seguito sono presentati i risultati dell'attacco MIA per i modelli di **classificazione**:

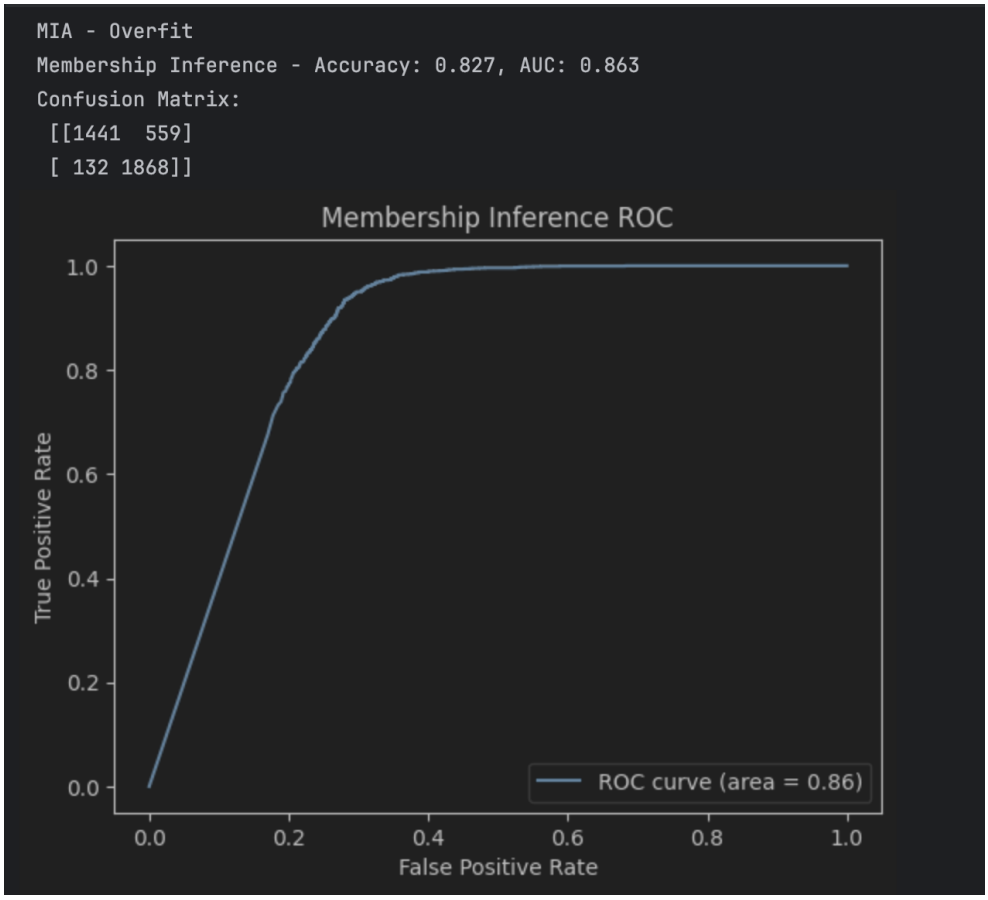


Fig. 23. Accuratezza del Membership Inference Attack sul modello di classificazione overfittato

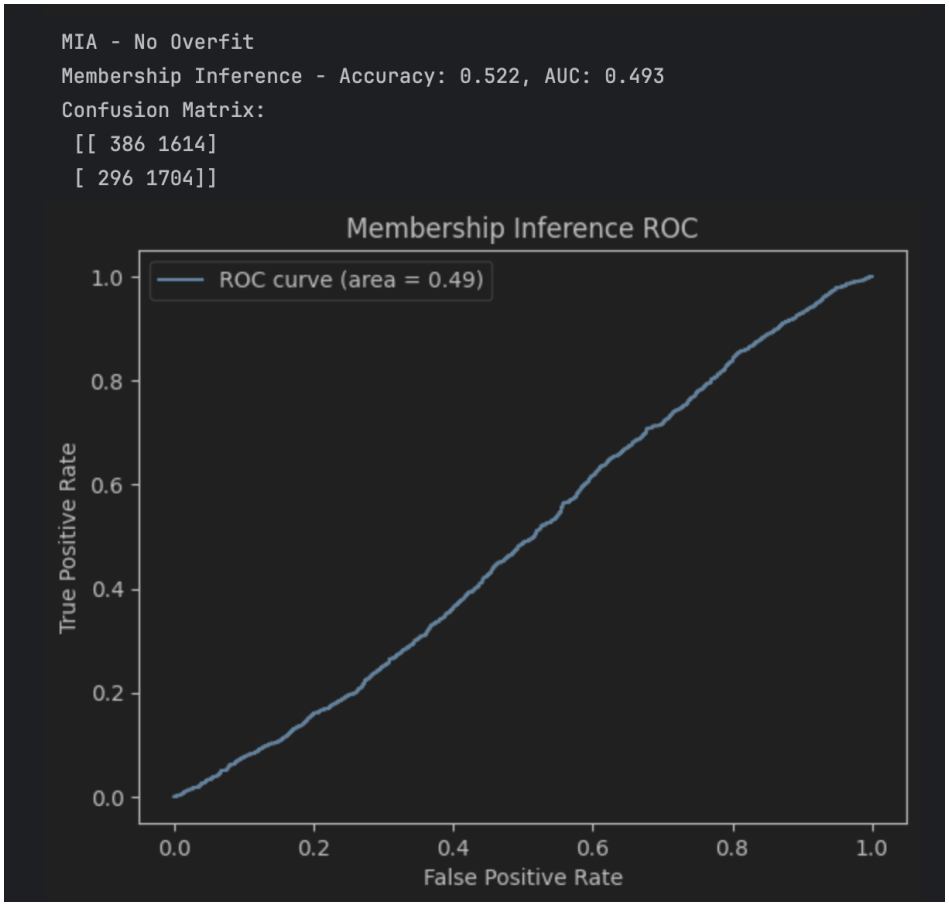


Fig. 24. Accuratezza del Membership Inference Attack sul modello di classificazione non overfittato

OverfitNet (Overfitting): L'attacco di Membership Inference sul modello overfittato ha raggiunto un'accuratezza di 0.827 e un AUC di 0.863. Un valore AUC superiore a 0.5 indica che l'attaccante ha una capacità significativa di distinguere i membri dai non membri del training set. Questo risultato è in linea con la letteratura, che suggerisce che i modelli overfittati sono più vulnerabili ai MIA a causa della loro maggiore confidenza sui campioni di training. La matrice di confusione mostra che il modello attaccante è stato in grado di identificare correttamente una grande parte dei membri e non membri.

NoOverfitNet (No Overfitting): Per il modello non overfittato, l'accuratezza dell'attacco è di 0.522 e l'AUC è di 0.493. Un AUC prossimo a 0.5 (e leggermente inferiore in questo caso) indica che l'attaccante non è in grado di distinguere i membri dai non membri con una probabilità significativamente migliore del caso.

Questo conferma che la regolarizzazione e una buona generalizzazione riducono la vulnerabilità ai Membership Inference Attacks.

Questi risultati per i modelli di classificazione sono coerenti con quelli ottenuti per la regressione, rafforzando l'evidenza che l'overfitting è un fattore chiave nella vulnerabilità ai Membership Inference Attacks.

10.5.1 Confronto qualitativo. I risultati ottenuti sono in linea con quanto riportato nel paper di Zhao et al. [15]: su modelli non overfittati l'attacco non ottiene performance superiori al caso random (AUC ≈ 0.49), mentre su modelli overfittati l'attacco è molto efficace (AUC ≈ 0.86). Il paper sottolinea anche che il successo di MIA non implica automaticamente il successo di attacchi di attribute inference, soprattutto nei modelli di classificazione, perché questi ultimi richiedono una forma più forte di membership inference (Strong Membership Inference).

10.6 Attribute Inference Attack (AIA) sulla Classificazione

L'attacco di inferenza di attributi sui modelli di classificazione mira a inferire un attributo specifico (es. un valore di pixel) di un campione, anche se non è l'obiettivo diretto del modello, sfruttando le sue predizioni e la conoscenza parziale del campione.

Codice per l'Attribute Inference Attack (Classificazione)

Il codice seguente implementa un attacco di **Attribute Inference** basato sulla confidenza predetta dal modello (*Conf AI*), come proposto nel paper [15]. Per ogni campione si generano diverse versioni modificate dell'input, variando il valore dell'attributo da indovinare. Si osserva quale variante produce la massima probabilità assegnata alla classe corretta e si assume che quel valore corrisponda a quello reale.

In particolare, in questo attacco si cerca di indovinare il valore di un pixel specifico (qui `feature_idx = 0`) tra un numero finito di candidati, scegliendo quello che massimizza la probabilità della classe vera predetta dal modello.

Attribute Inference Attack (Classificazione)

```
1
2  import torch
3  import numpy as np
4  import matplotlib.pyplot as plt
5  from sklearn.metrics import roc_curve, auc, confusion_matrix,
   ↪  accuracy_score
6
7
```

```
def get_probs_pytorch(model, inputs, device):
    model.eval()
    with torch.no_grad():
        inputs = inputs.to(device)
        outputs = model(inputs)
        probs = torch.softmax(outputs, dim=1)
    return probs.cpu().numpy()

def attribute_inference_attack_pytorch(model, dataloader, device,
↪  feature_idx=0, n_candidates=5, n_samples=300):
    model.eval()
    # Estrai n_samples dal dataloader
    X, y = [], []
    for inputs, labels in dataloader:
        X.append(inputs)
        y.append(labels)
        if len(torch.cat(y)) >= n_samples:
            break
    X = torch.cat(X)[:n_samples].cpu().numpy()
    y = torch.cat(y)[:n_samples].cpu().numpy()

    correct = 0
    for i in range(n_samples):
        x = X[i].copy()
        true_val = x.flat[feature_idx] # Valore vero dell'attributo da
        ↪  inferire
        candidates = np.linspace(0, 1, n_candidates) # Valori candidati
        ↪  per l'attributo (es. pixel da 0 a 1)
        best_score = -np.inf
        best_val = None
        for val in candidates:
            x_cand = x.copy()
            x_cand.flat[feature_idx] = val # Modifica l'attributo con il
            ↪  valore candidato
            x_cand_tensor =
            ↪  torch.from_numpy(x_cand).unsqueeze(0).to(device)
            probs = get_probs_pytorch(model, x_cand_tensor, device)[0]
            prob = probs[y[i]] # Probabilit della classe vera per il
            ↪  campione modificato
            if prob > best_score:
                best_score = prob
```

```
                best_val = val
        # Se il valore inferito  "vicino" al valore vero, conta come
        ↪  corretto
        if np.isclose(best_val, true_val, atol=1.0/(n_candidates-1)):
            correct += 1
    acc = correct / n_samples
    print(f"Attribute Inference - Accuracy: {acc:.3f} (feature idx
    ↪  {feature_idx})")
    plt.bar(['Correct', 'Incorrect'], [correct, n_samples-correct])
    plt.title(f'Attribute Inference (feature idx {feature_idx})')
    plt.show()
    return acc

# Esecuzione dell'attacco di inferenza di attributi
feature_idx = 0  # Si attacca il primo pixel dell'immagine
print("AIA - Overfit")
attribute_inference_attack_pytorch(model, testloader, device,
↪  feature_idx=feature_idx, n_candidates=5)

print("AIA - No Overfit")
attribute_inference_attack_pytorch(no_overfit_model, testloader, device,
↪  feature_idx=feature_idx, n_candidates=5)
```

Risultati dell'Attribute Inference Attack (Classificazione)

Di seguito sono presentati i risultati dell'attacco AIA per i modelli di classificazione:

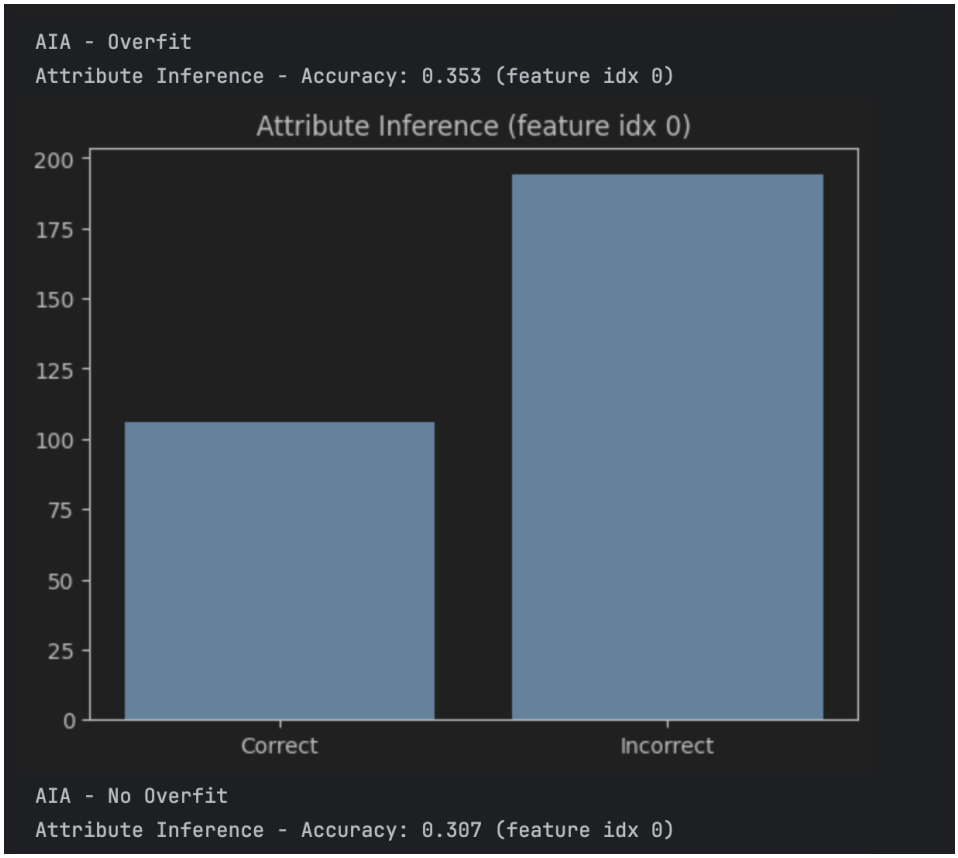


Fig. 25. Accuratezza dell'Attribute Inference Attack sul modello di classificazione overfittato

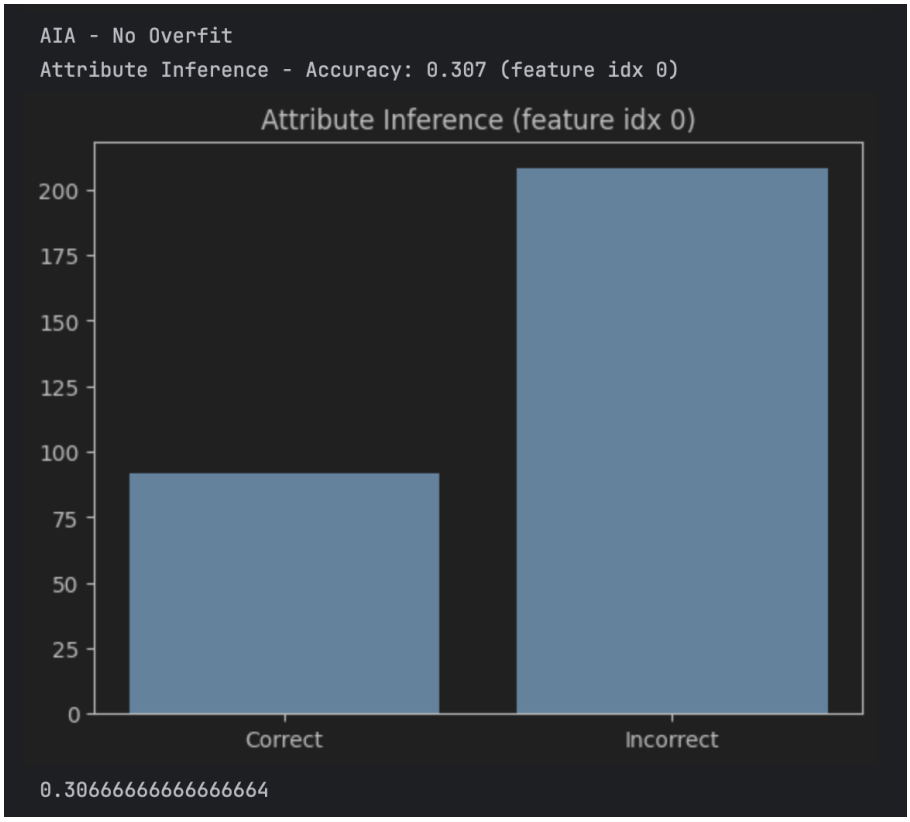


Fig. 26. Accuratezza dell'Attribute Inference Attack sul modello di classificazione non overfittato

OverfitNet (Overfitting): L'attacco di inferenza di attributi sul modello overfittato ha ottenuto un'accuratezza di 0.353. Questo significa che l'attaccante è riuscito a indovinare correttamente il valore del pixel target nel 35.3% dei casi.

NoOverfitNet (No Overfitting): Per il modello non overfittato, l'accuratezza dell'attacco è di 0.307. Questo valore è leggermente inferiore rispetto al modello overfittato.

In questo caso, la differenza nelle accuratezze degli attacchi di Attribute Inference tra i due modelli di classificazione non è così marcata come quella osservata per i Membership Inference Attacks. Questo potrebbe suggerire che, per questo tipo specifico di attacco e per l'attributo scelto (un singolo pixel), la relazione con l'overfitting non è altrettanto diretta o che l'attributo stesso non è fortemente "memorizzato" in modo tale da facilitare l'inferenza anche in un modello overfittato. È possibile che attributi più complessi o più direttamente correlati alla funzione di classificazione del modello possano rivelare una maggiore vulnerabilità in presenza di overfitting.

10.6.1 Confronto qualitativo Zhao et al. [15] sottolineano che, anche su modelli di classificazione altamente overfittati, gli attacchi di attribute inference (AI) non sono efficaci: nei risultati sperimentali, l'accuracy degli attacchi AI sui modelli di classificazione è vicina al caso random (cioè, per un attributo binario, circa 0.5; per attributi con più classi, anche meno), e non migliora significativamente con l'overfitting. Gli autori specificano che il successo di AI richiede una forma più forte di membership inference (SMI), che non viene raggiunta nemmeno da attacchi MI efficaci.
I nostri risultati sono allineati con quelli del paper:

- L'accuracy ottenuta è molto bassa sia in assenza che in presenza di overfitting;
- Nel paper viene spiegato che questo avviene perché, nei modelli di classificazione, anche se il modello è vulnerabile a MI, non lo è necessariamente ad AI, a meno che l'attacco MI non sia in grado di distinguere membri da "vicini" (Strong Membership Inference), cosa che non avviene negli esperimenti da loro eseguiti;

11 Conclusion

L'esperimento condotto ha permesso di esplorare la relazione tra overfitting e la vulnerabilità dei modelli di Machine Learning agli attacchi di inferenza, distinguendo tra modelli di regressione e di classificazione. Sono stati analizzati in particolare gli attacchi di Membership Inference (MIA) e Attribute Inference (AIA).

11.1 Confronto con la Letteratura: *Modelli di Regressione*

Analizziamo innanzitutto gli attacchi eseguiti sui modelli di **regressione** (Rete Neurale overfittata e Ridge Regression generalizzato) addestrati sul dataset IWPC 2009.

Membership Inference Attack (MIA):
I risultati ottenuti per il MIA sono pienamente concordi con le osservazioni del paper "Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting" di Yeom et al. (2018). Il modello overfittato (Rete Neurale) ha mostrato una vulnerabilità significativamente maggiore al MIA (AUC elevato), indicando che i campioni di training sono stati "memorizzati" con alta confidenza. Al contrario, il modello generalizzato (Ridge Regression) ha dimostrato una robustezza notevole contro questo tipo di attacco (AUC prossimo a 0.5), suggerendo una minore perdita di informazioni sull'appartenenza al training set.

Attribute Inference Attack (AIA):
Per l'attacco di Attribute Inference sull'attributo 'Gender', i risultati hanno evidenziato una leggera discrepanza rispetto alle attese basate sul paper di Yeom et al. Sebbene il paper suggerisca che un maggiore overfitting dovrebbe correlare

con una maggiore vulnerabilità all'AIA (quando l'attributo è influente), il nostro esperimento ha mostrato che l'attacco ha avuto successo su entrambi i modelli, con un'accuratezza persino leggermente superiore sul modello generalizzato. Diverse ragioni possono contribuire a spiegare questa discrepanza:

- **Natura dell'Attributo Sensibile ('Gender') e sua "Influenza"**:
  Il paper sottolinea che il successo dell'attacco di Attribute Inference in relazione all'overfitting dipende in parte dall'"influenza" dell'attributo target. Un attributo è considerato influente se le sue variazioni producono un impatto significativo sulla funzione di perdita del modello. Se l'attributo 'Gender' nel dataset IWPC 2009 non è sufficientemente "influente" nel senso descritto dal paper, o se la sua correlazione con l'output del modello target è relativamente semplice e lineare, un modello generalizzato potrebbe già apprendere sufficienti relazioni per esporre l'attributo, rendendo meno rilevante il vantaggio derivante dall'overfitting.

- **Informazioni Latenti nelle Feature del Dataset**:
  Anche in un modello generalizzato come la Ridge Regression, se l'attributo 'Gender' è intrinsecamente correlato ad altre feature presenti nel dataset di input (X), queste correlazioni possono essere apprese dal modello target. Di conseguenza, l'attaccante, avendo accesso alle feature originali oltre alle predizioni del modello target, potrebbe inferire l'attributo sensibile principalmente da queste correlazioni preesistenti, indipendentemente dal grado di overfitting del modello target.

- **Composizione delle Feature per l'Attaccante**:
  L'attacco costruisce le feature per il modello attaccante concatenando le feature originali del dataset $(X_{\text{tr}}, X_{\text{te}})$ con le predizioni del modello target $\langle \text{pred}_{\text{tr}}, \text{pred}_{\text{te}} \rangle$. Se le feature originali da sole contengono già un'informazione sufficiente per predire l'attributo 'Gender', il successo dell'attacco potrebbe essere attribuito primariamente a queste feature, piuttosto che alle specifiche "fughe di informazione" indotte dall'overfitting nelle predizioni del modello target. Un test che potrebbe chiarire questo aspetto sarebbe addestrare il modello attaccante utilizzando esclusivamente le predizioni del modello target come input.

- **Specificità del Dataset e Configurazione del Modello**:
  Nonostante l'utilizzo di un dataset come IWPC 2009, eventuali differenze nella pre-elaborazione dei dati, nella selezione delle feature, o nella specifica configurazione degli iperparametri (es. per la rete neurale e la Ridge Regression, o la metodologia esatta per indurre l'overfitting) rispetto all'implementazione degli autori del paper, potrebbero portare a risultati divergenti.

- **Robustezza dell'Attacco di Inferenza**:
  Gli attacchi di Attribute Inference, specialmente quando si utilizzano modelli attaccanti robusti come la Regressione Logistica, possono avere successo se l'attributo sensibile è intrinsecamente prevedibile sulla base del resto del

dataset, anche in assenza di un overfitting marcato nel modello target. Il paper enfatizza come l'overfitting amplifichi il rischio di privacy, ma ciò non implica che sia l'unica condizione per il successo di un attacco.

11.2 Confronto con la Letteratura: *Modelli di Classificazione*

Ora invece analizziamo gli attacchi eseguiti sui modelli di **classificazione** (OverfitNet e NoOverfitNet) addestrati sul dataset CIFAR100:

Membership Inference Attack (MIA):
I risultati del MIA sui modelli di classificazione sono pienamente concordi con gli studi di Zhao et al. (2021) che evidenziano la relazione tra overfitting e vulnerabilità ai MIA nei modelli di classificazione. OverfitNet ha dimostrato una chiara vulnerabilità (alto AUC), confermando che i modelli che overfittano tendono a esporre l'appartenenza dei dati al training set. Al contrario, NoOverfitNet, grazie alle tecniche di regolarizzazione, ha mostrato una resistenza significativa al MIA (AUC prossimo a 0.5).

Attribute Inference Attack (AIA):
Anche per l'AIA sui modelli di classificazione (attaccando un pixel specifico delle immagini), i risultati sono concordi con quanto generalmente osservato nella letteratura. Sebbene l'accuratezza dell'attacco non sia stata estremamente elevata su entrambi i modelli, è stata leggermente superiore per il modello overfittato (OverfitNet). Questo suggerisce una tendenza, seppur meno marcata rispetto al MIA, a una maggiore esposizione di attributi specifici in presenza di overfitting, coerentemente con le teorie che collegano la memorizzazione di dettagli specifici (inclusi i valori di attributi) all'overfitting. La differenza meno marcata rispetto al MIA indica che l'inferenza di un singolo attributo (come un pixel) potrebbe dipendere meno criticamente dall'overfitting rispetto alla Membership Inference.

In sintesi, i nostri esperimenti rafforzano l'evidenza che l'overfitting è un fattore critico nella vulnerabilità dei modelli di Machine Learning agli attacchi di Membership Inference, sia per la regressione che per la classificazione. Per gli attacchi di Attribute Inference, la relazione con l'overfitting si è rivelata più sfumata e dipendente dalla specifica natura dell'attributo sensibile e del contesto del dataset, come evidenziato dalla potenziale discrepanza osservata per l'attributo 'Gender' nei modelli di regressione. Tuttavia, per i modelli di classificazione, anche l'AIA ha mostrato una correlazione, seppur debole, con l'overfitting. Queste osservazioni sottolineano l'importanza di considerare attentamente le tecniche di regolarizzazione e la natura degli attributi sensibili durante lo sviluppo di modelli di ML, al fine di mitigare i rischi per la privacy.

11.3 Conclusione al Quesito: Are the results by Yeom et al and Zhao et al in contradiction?

La domanda fondamentale che emerge confrontando la letteratura e i nostri risultati è se le scoperte di Yeom et al. e Zhao et al. siano in contraddizione. La nostra analisi suggerisce che entrambi gli studi presentano conclusioni valide, e le

apparenti discrepanze risiedono principalmente nella diversa natura dei modelli di Machine Learning sui quali sono stati condotti gli attacchi:

Yeom et al. [13] si concentra su modelli di regressione. In questi contesti, l'overfitting tende a portare a una memorizzazione più diretta e dettagliata dei campioni di training, inclusi gli attributi sensibili. Se un attributo è sufficientemente rilevante o correlato all'output del modello, l'overfitting può amplificare significativamente la vulnerabilità agli AIA. La lieve discordanza riscontrata nei nostri esperimenti sull'AIA per la regressione non invalida la loro tesi generale sull'aumento del rischio dovuto all'overfitting, ma evidenzia come altri fattori, quali le correlazioni intrinseche degli attributi nel dataset o la costruzione delle feature per l'attaccante, possano influenzare in modo preponderante la possibilità di inferire attributi specifici.

Zhao et al. [15], focalizzandosi sui modelli di classificazione, argomenta che, sebbene questi possano essere soggetti a MIA (soprattutto se overfittati), raramente lo sono a precisi AIA. Questo perché i MIA sui classificatori spesso non raggiungono una "Strong Membership Inference" (SMI), ovvero la capacità di distinguere un membro da un "non-membro molto simile" nello spazio dei dati. I nostri risultati per l'AIA sui modelli di classificazione, con accuratezze modeste anche per il modello overfittato, supportano pienamente questa prospettiva: la memorizzazione indotta dall'overfitting in un classificatore non è sufficientemente fine o specifica da consentire l'inferenza accurata di un attributo.

In definitiva, la presunta "contraddizione" si risolve riconoscendo che:

L'overfitting costituisce un fattore di rischio universale per i Membership Inference Attacks su entrambe le tipologie di modelli, come dimostrato sia da Yeom et al. che da Zhao et al. (e dai nostri esperimenti).

La riuscita degli Attribute Inference Attacks dipende in modo critico dalla natura del modello (regressione vs. classificazione) e dall'attributo in questione. I modelli di regressione, data la loro natura a output continuo, possono esporre gli attributi tramite overfitting in modi diversi rispetto ai classificatori, i quali, pur overfittando, potrebbero non rivelare segnali sufficientemente chiari per un'inferenza precisa di singoli attributi.

Queste osservazioni evidenziano l'importanza di valutare attentamente le tecniche di regolarizzazione e la natura degli attributi sensibili durante lo sviluppo di modelli di ML, al fine di mitigare i rischi per la privacy, e sottolineano la complessità delle interazioni tra overfitting, tipo di modello e vulnerabilità alla privacy.

References

1. Consortium, I.W.P.: Estimation of the warfarin dose with clinical and pharmacogenetic data. New England Journal of Medicine **360**(8), 753–764 (2009)
2. DMDave, B, T., Cukierski, W.: Acquire valued shoppers challenge. https://kaggle.com/competitions/acquire-valued-shoppers-challenge (2014), kaggle
3. Dwork, C., Roth, A., et al.: The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science **9**(3–4), 211–407 (2014)

4. Fredrikson, M., Lantz, E., Jha, S., Lin, S., Page, D., Ristenpart, T.: Privacy in pharmacogenetics: An {End-to-End} case study of personalized warfarin dosing. In: 23rd USENIX security symposium (USENIX Security 14). pp. 17–32 (2014)
5. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
6. LeCun, Y.: The mnist database of handwritten digits. http://yann. lecun. com/exdb/mnist/ (1998)
7. Li, N., Qardaji, W., Su, D., Wu, Y., Yang, W.: Membership privacy: A unifying framework for privacy definitions. In: Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. pp. 889–900 (2013)
8. Nasr, M., Shokri, R., Houmansadr, A.: Comprehensive privacy analysis of deep learning. In: Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP). vol. 2018, pp. 1–15 (2018)
9. Salem, A., Zhang, Y., Humbert, M., Berrang, P., Fritz, M., Backes, M., et al.: Model and data independent membership inference attacks and defenses on machine learning models (2018)
10. Scheetz, T.E., Kim, K.Y.A., Swiderski, R.E., Philp, A.R., Braun, T.A., Knudtson, K.L., et al.: Regulation of gene expression in the mammalian eye and its relevance to eye disease. Proceedings of the National Academy of Sciences **103**(39), 14429–14434 (2006)
11. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: 2017 IEEE symposium on security and privacy (SP). pp. 3–18. IEEE (2017)
12. Yang, D., Zhang, D., Qu, B.: Participatory cultural mapping based on collective behavior data in location-based social networks. ACM Transactions on Intelligent Systems and Technology (TIST) **7**(3), 1–23 (2016)
13. Yeom, S., Giacomelli, I., Fredrikson, M., Jha, S.: Privacy risk in machine learning: Analyzing the connection to overfitting. In: 2018 IEEE 31st computer security foundations symposium (CSF). pp. 268–282. IEEE (2018)
14. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning requires rethinking generalization. arXiv preprint arXiv:1611.03530 (2016)
15. Zhao, B.Z.H., Agrawal, A., Coburn, C., Asghar, H.J., Bhaskar, R., Kaafar, M.A., et al.: On the (in) feasibility of attribute inference attacks on machine learning models. In: 2021 IEEE European Symposium on Security and Privacy (EuroS&P). pp. 232–251. IEEE (2021)
16. Zhao, B.Z.H., Asghar, H.J., Bhaskar, R., Kaafar, M.A.: On inferring training data attributes in machine learning models. arXiv preprint arXiv:1908.10558 (2019)