Relazione

Elisa Acciari, Giulio Cappelletti

January 2024

Abstract In un mondo dove l'intelligenza Artificiale è sempre più presente nelle nostre vite e nel settore industriale, nasce la necessità di comprendere e proteggere questi sistemi e i dati sensibili che trattano dalle possibili minacce a cui sono costantemente sottoposti.

L'obiettivo del lavoro svolto è di presentare una revisione dello stato dell'arte relativo alla disciplina dell'Adversarial Machine Learning. Questa revisione offre un punto di ingresso alla disciplina permettendo di comprendere la tassonomia degli attacchi, delle difese principali e dei recenti sviluppi.

La prima parte tratta della fase di modellazione delle minacce, in cui sono state approfondite le vulnerabilità nei sistemi di Machine Learning, le minacce che ne derivano ed è stato modellato l'attaccante in base alle sue proprietà. Questa ha permesso di fare luce nella disciplina dell'AML e di realizzare una tassonomia degli attacchi.

La seconda parte si concentra sulla presentazione degli attacchi principali, trattati nello stato dell'arte, rivolti contro i sistemi di apprendimento automatico. In questa parte, assumono rilevante importanza due classi specifiche, gli Attacchi di Avvelenamento dei dati (Poisoning Attacks), condotti nella fase di training, e gli Attacchi Esplorativi o di Evasione (Exploratory/Evasion Attack), condotti in fase di testing e di inferenza.

La terza parte presenta una panoramica delle soluzioni più conosciute come meccanismi di difesa agli attacchi presentati in precedenza. Nello specifico, abbiamo esplorato questi metodi in base a se gli attacchi avvengono nella fase di training o di testing, ma soprattutto, come meccanismo per la difesa della privacy dei dati sensibili, si è posta attenzione su una recente metodologia di apprendimento progettata, appunto, per diminuire perdite di dati sensibili, il Federated Learning (FL). Ampliamo questa particolare difesa andando a riferirci al paper [1] il quale spiega nel dettaglio il suo funzionamento.

Nello studio svolto, siamo andati ad espandere ogni fase appena descritta; di seguito tratteremo i capitoli presentati nel documento in questione, fornendone una panoramica e i punti chiave trattati.

1 Background

In questa sezione dello studio sono state date le basi degli argomenti trattati, spiegando cosa si intende per Intelligenza Artificiale (AI) e per Machine Learning (ML).

Inizialmente, abbiamo suddiviso le tipologie di Machine Learning in quattro categorie: Supervised, Unsupervised, Semisupervised e Reinforcement Learning.

Poi abbiamo elencato le fasi di sviluppo di un sistema di ML, in quanto ci serviranno per capire dove avvengono gli attacchi e le difese che tratteremo nel capitolo degli attacchi (Capitolo3):

- Raccolta dati
- Preparazione e pulizia dei dati
- Training
- Validazione del modello attraverso un validation set con l'obiettivo di ottimizzarne gli iperparametri
- Testing

Successivamente, si è spiegata la definizione della funzione di perdita e dei gradienti, in quanto vedremo che ci saranno degli attacchi che si basano o meno su quest'ultimo, sempre nel capitolo riferito agli attacchi 3.

2 Sicurezza nei sistemi di ML

Nel **capitolo 3** del progetto, sono articolati gli elementi ricorrenti nella sicurezza dei sistemi di ML.

Approccio alla sicurezza (Capitolo 3.1): L'approccio varia in base ai comportamenti che un attaccante e un difensore possono mantenere:

- *Approccio Reattivo*: Attaccante e Difensore adattano la propria mossa in funzione dell'avversario imparando dalle azioni passate. Tuttavia, presenta un elevato grado di rischio in presenza di attacchi mai visti.
- *Approccio Proattivo*: L'obiettivo è la prevenzione dagli attacchi. Per anticipare gli attacchi si devono identificare le vulnerabilità e le minacce che ne derivano prima del verificarsi di un attacco. Simulare l'attacco e pianificare adeguate procedure di difesa.

Definizione del problema (Capitolo 3.2): In base all'approccio proattivo emerge la necessità di considerare, non solo la parte di learning del modello, ma soprattutto la fase di definizione del problema [2]. Mancata attenzione in questa fase si ripercuote in un debito tecnico. Alcuni aspetti da definire sono:

- Fonte dei dati da usare e qualità dei dati.

- Proteggere eventuali dati sensibili.
- Origine del modello in caso sia pre-addestrato.
- Finalità del modello: E' emersa una prima importante considerazione, ovvero, che la maggior parte degli attacchi conosciuti si concentrano contro sistemi di apprendimento nell'ambito della Computer Vision, sfruttando l'alta dimensionalità (i pixel) dei dati di training.

L'approccio proattivo è stato articolato secondo la definizione delle "three golden rules" [3]: 1) "Know your adversary", 2) "Be proactive", 3) "Protect yourself". Abbiamo seguito questa struttura, andandola a integrare con altri lavori per approfondire le singole parti.

Modeling Threats (Capitolo 3.3): Secondo la prima regola, la prima fase prevede la modellazione delle minacce, o threat modelling [3]: un processo strutturato attraverso il quale si identificano le vulnerabilità del sistema, si modellano le minacce, le capacità dell'attaccante e si quantifica la gravità delle minacce modellate.

Nel Capitolo 3.3.1, concentrandosi su [2] [4] sono state identificate le vulnerabilità che possono verificarsi nella progettazione di un sistema di apprendimento automatico. **Vulnerabilità nei dati**: Queste sono prevalentemente in fase di training. Fonti di vulnerabilità possono essere:

- Assunzioni sui dati: linearità dei dati, separabilità tra le classi, stazionarietà dei dati e indipendenza delle features (i.i.d.) . L'avversario potrebbe presentare dati che violano queste assunzioni.
- Iterative Learning: Training in cui un modello viene continuamente aggiornato e addestrato con nuovi dati variabili nel tempo.

Vulnerabilità nel modello: L'avversario pur non avendo accesso ai dettagli interni del modello, come pesi o i bias, potrebbe comunque "copiare" o "rubare" il modello osservandone le previsioni. Alcune vulnerabilità sono:

- Overfitting
- Modelli pre-addestrati

Vulnerabilità nel sistema: Questa vulnerabilità riguarda le vulnerabilità presenti nelle componenti costruttive e funzionali dell'intero sistema [4].

In base alle vulnerabilità gli attacchi possono quindi essere rivolti ai dati o al modello (Tabella 2).

Nel Capitolo 3.3.2 è stata effettuata la **modellazione dell'attaccante** in base a tre dimensioni:
Attacker's Goal: Individuare le intenzioni o i risultati a cui l'attaccante mira. L'attacco può essere categorizzato in base a:

- Specificità dell'attacco:

  - Targeted: dato l'input x classificato nella classe C(i), si vuole perturbare x in x' in modo che sia classificato come appartenente a una specifica classe C(j).
  - Untargeted: l'unico obiettivo è quello di mal classificare una classe C(i) in qualsiasi altra classe C(j).

- Tipo di violazione: Descrive il tipo di attacco sulla base di ciò che viene compromesso nel sistema. In base alla triade CIA (Confidenzialità Integrità e Disponibilità) sono stati categorizzati gli attacchi (Tabella 2).

Attacker's Influence: Prendendo in esame [4],[3] [5], l'attaccante può essere categorizzato in base al livello di influenza e alle sue capacità. Nell'AML, l'influenza è:

- *"Causativa" (Causative Attacks)* quando l'attaccante può manipolare sia i dati di training che di testing. Questi sono riconducibili ai *Poisoning Attacks*.

- *"Esplorativa" (Exploratory Attacks)* l'attaccante può influire solamente sui dati di testing e mira ad estrapolare informazioni interrogando il modello. Questo scenario è noto come *Evasion Attacks* (Tabella 2).

Attacker's Knowledge: L'attaccante è stato modellato in base a ciò che conosce del sistema. In base al livello di accesso ai dati, considerando quindi sia conoscenza che capacità dell'attaccante, ci può essere [6]:

- Offline Attacker: Può accedere a una versione statica del modello.

- Online Attcker: L'attaccante agisce in fase di inferenza inviando input al sistema.

- Semi-Online Attacker: L'attaccante agisce sempre in tempo reale, ma elabora le informazioni raccolte offline per creare una copia del modello

Successivamente, abbiamo trattato il livello di conoscenza in base agli scenari: **white-box**, **black-box** e **grey-box**. Per esaminare questi scenari abbiamo riportato un formalismo in cui gli scenari sono caratterizzati in base al grado di conoscenza dei seguenti elementi [3]:

- D: Dati di training

- X: Insieme delle features X

- f: Algoritmo di apprendimento f

- w: parametri/iperparametri del modello

	Fase di Training	Fase di Testing
Integrità	Causative Attacks (Poisoning Attacks rivolti a consentire intrusioni, come i Backdoor Attacks)	Exploratory Attacks (Evasion Attacks)
Confidenzialità		Model Extraction e Model Inversion
Disponibilità	Poisoning Attacks rivolti a massimizzare errori di classificazione (Scenario UNTARGETED)	
Attacchi ai dati	Poisoning Attacks	Evasion Attacks (Adversarial Examples)
Attacchi al modello		Model Extraction e Model Inversion

Tassonomia degli attacchi in base al Threat Modeling.

Come futuri sviluppi si incoraggia la comunità di ricerca a una maggior comprensione del Machine Learning, e soprattutto del Deep Learning, per andare a delineare le vulnerabilità che potrebbero essere sfruttate dagli attaccanti, soprattutto nei casi con conoscenza limitata da parte dell'avversario.

3 Adversarial Machine Learning

L'Adversarial Machine Learning è un campo di studio nato dall'intersezione tra i campi dell'Intelligenza Artficiale (AI) e della Sicurezza Informatica, che studia come i modelli possano essere manipolati o ingannati da attori malevoli attraverso input malevoli o perturbarti, e di come difendere i modelli da questi attacchi.

In questo capitolo, seguendo la seconda delle Three golden rules presentate, "Be Proactive", siamo andati ad espandere gli attacchi presentati nella tassonomia del capitolo precedente 2. Inizialmente li suddividiamo in due categorie, in base a se avvengono nella:

- Fase di **Training**
- Fase di **Testing**

Questa differenziazione è stata svolta riportando quanto scritto nel paper [3]

3.1 Attacchi durante la fase di Training

Nella fase di Training rientrano i **Poisoning Attacks**, questi hanno l'obbiettivo di minare le prestazioni del modello, aumentando il numero di campioni mal

classificati, effettuando un "avvelenamento" dei dati. Per questi tipi di attacchi ci siamo riferiti ai paper [2], [6], ma soprattutto ai documenti [3] e [7].

Secondo quanto riportato in [3], si distinguono in:

- **targeted**, ovvero mirano ad avvelenare una porzione specifica di dati
- **untargeted**, mirano a indurre quante più possibili classificazioni errate per rendere il modello inutilizzabile

Rispetto alla triade CIA, questi tipi di attacchi rientrano in quelli che compromettono l'**integrità** e la **disponibilità** rispettivamente.

Gli attacchi di avvelenamento a loro volta, per la definizione riportata in [7], includono:

- **Inadeguate data injection**, vi è l'inserimento nel dataset pulito dei dati non rappresentativi per l'output del modello; questo può portare a diminuire l'accuratezza del modello e di conseguenza le sue prestazioni.portare
- **Logic Corruption**, attacchi che vanno a compromettere la logica del modello, ovvero l'algoritmo su cui si basa e il modo con cui apprende.
- **Backdoor Attack** , permettono di alterare il comportamento del training attraverso l'introduzione di un "*trigger*" all'interno del processo di addestramento. Quest'ultimo, una volta verificatasi la sua condizione di attivazione, apporta modifiche alla label di un input in modo che sia diversa da quella originale. In questo modo il modello viene addestrato sui dati avvelenati.
- **Data Manipulation**, va a manipolare input, feature e label, ma può essere collegato anche alla validità e alla qualità dei dati raccolti per il training, ovvero nel momento in cui questi dati provengono da fonti non attendibili o se non sono coerenti con l'obbiettivo del modello. Poichè porterebbe ad una diminuzione della qualità dell'output del modello.

3.2 Attacchi durante la fase di Testing

Quando il modello è già addestrato e viene usato per effettuare previsioni su nuovi dati, ovvero per fare inferenza, è esposto a una serie di attacchi avversari. In questa fase rientrano gli attacchi nello scenario di influenza "Esplorativa".

Gli attacchi più noti, secondo quanto scritto in [7], sono:

- **Evasion Attacks**, mirano a introdurre perturbazioni nei dati in fase di test con l'intento eludere un classificatore già addestrato. Per la definizione di questi abbiamo fatto riferimento oltre che sul paper [7], anche sui documenti [3] e [4]. Questi input sono noti come **Adversarial examples**. Si suddividono in:
  - *Gradient based*, in cui l'attaccante calcola il gradiente per ottenere informazioni aggiuntive con le quali modificano degli input in input

perturbati, ovvero adversarial exemple, tra i dati di test originari, in modo da inficiare sulle classificazioni del modello.

  - *Gradient Free*, ovvero non basati sul gratiente, ma su metriche diverse. In questo caso non è necessario per l'attaccante accedere al dataset completo del modello, ma solamente a parametri limitati.

- **Oracle attacks**, in questo caso l'attaccante non dispone delle informazioni delle metriche interne del modello, ma ha conoscenze solamente sulle coppie input-output e sulle probabilità di classe.

  - *Model Extraction*, attraverso le informazioni che l'attaccante acquisisce facendo inferenza, hanno l'obbiettivo di estrarre un modello sostitutivo che emula il comportamento di quello originario. Quest'ultimo sarà usato per generare adversarial examples, che riusciranno ad attaccare il modello grazie al principio della trasferibilità di questi.
  - *Model Inversion*, l'obiettivo principale dell'attaccante è determinare gli input originali. L'attaccante, avendo accesso solamente ad alcuni parametri e alle etichette di input, cerca di invertire il processo di addestramento del modello per ottenere informazioni sugli input.
  - *Membership Inference*, ha l'obbiettivo di risalire agli input utilizzati nell'addestramento del modello attraverso un approccio brute force di inferenza.

Successivamente siamo andati a spiegare più nel dettaglio, in una sezione apposita , gli Adversarial Examples.

3.3 Adversarial Examples

Sono le singole istanze di input (immagini, suoni, testi o altro) creati applicando piccole ma intenzionali perturbazioni, in modo che il dato perturbato induca il modello a restituire una risposta errata con alta probabilità e quindi a mal classificare. Per spiegare questo argomento ci siamo basati su quanto riportato nel documento [8] e [7].

Secondo quanto scritto in [8], uno dei metodi più utilizzati per generare adversarial examples, è il Fast gradient sign method (FGSM). Questa è stata approfondita mediante un esempio che mostra come degli input perturbati inseriti in fase di test portano il classificatore a mal classificare. In particolare, dagli studi su questo argomento, è emersa come preoccupazione la loro proprietà di trasferibilità[8]. La quale indica che questi possano essere adottati per attaccare anche differenti modelli.

Inoltre, spieghiamo una delle possibili cause ipotizzate in [8] dell'efficacia degli adversarial examples, ovvero l'alta linearità delle reti neurali.

Considerazioni su direzioni future. Nella modellazione degli attacchi emerge la necessità di incrementare la rilevazione delle manipolazioni in fase di input per contrastare gli attacchi di avvelenamento. Inoltre, approfondire la spiegabilità degli adversarial examples, ancora oggi oggetto di controversie,

permetterebbe di indagare più a fondo in questi attacchi, sulla loro effettiva efficacia e sulla loro proprietà di trasferibilità

3.4 Difese

Come abbiamo visto nel capitolo "Sicurezza nei problemi di Machine learning" 2, si hanno due approcci per affrontare i problemi di sicurezza: reattivo o proattivo. Anche qui siamo andati a definire le difese, secondo quanto riportato in [3], in questi termini:

- Difesa Reattiva, strategia che mira a contrastare attacchi già avvenuti, e che si basa sul rilevamento rapido di nuovi attacchi non appena si verificano.
- Difesa Proattiva, si basa proprio sull'andare ad identificare le minacce prima che si verifichino. Siamo andati ad espandere quest'ultimo approccio, in quanto è quello più complesso, ma più efficacie.

Da ciò che è riportato in [3], abbiamo illustrato i passi del **ciclo di sicurezza proattivo**, che si identificano in:

- identificazione delle minacce
- simulazione degli attacchi
- elaborazione di contromisure agli attacchi identificati nella prima fase.

Abbiamo individuato, sepre nel documento [3], due suddivisioni all'interno di queste difese proattive:

- **Security-by-design**, si basa sul concetto di "irrobustimento" del modello. Queste sono progettate in un ambiente white-box, ovvero in cui si presuppone che l'attaccante abbia completa conoscenza del sistema. Per capire bene l'argomento, siamo andati a trattare degli esempi di difese di questa categoria, spiegando se vanno a contrastare gli evasion attaks o i poisoning attacks.
- **Security-by-obsurity** e, a differenza della prima, questa tecnica di difesa ha l'obbiettivo di offuscare le informazioni di interesse agli aggressori in modo da rendere più difficile effettuare gli attacchi. Quindi lavorano contro gli attacchi black box e grey box. Nello specifico, siamo andati a spiegare le *strategie* che comprendono questa categoria di difese, ovvero:
  - randomizzazione della raccolta dei dati di addestramento
  - utilizzo di classificatori difficili da decodificare
  - rifiuto dell'accesso al classificatore effettivo o ai dati di addestramento
  - randomizzazione dell'output del classificatore.

Le tecniche trattate in queste ultime due categorie saranno spiegate nel dettaglio nella sezione successiva.

3.5 Difese nell'AML

Nella sezione relativa alle **Difese nell'AML**, come per quella degli attacchi, siamo andati a fare una suddivisione in base a se le difese avvengono in fase di Training o di Testing, basandoci su ciò che viene riportato nel documento [7].

3.5.1 Fase di Training

Vengono proposte 3 tipologie di difese tra cui:

- Data encryption, tecnica che converte i dati originali in dati cifrati, rendendoli accessibili solo all'utente e al fornitore di servizi, garantendone la privacy.
- Data sanitization, comporta la rilevazione di campioni di input dannosi per la classificazione per poterli poi rimuovere. Può essere aggirata dagli aggressori, andando a produrre campioni avvelenati che assomigliano molto alla distribuzione dei dati originali.
- Robust learning, ha l'obbiettivo di sopprimere le potenziali distorsioni causate da campioni avvelenati andando ad utilizzare robuste tecniche statistiche e migliorando la generalizzazione, invece di rilevare direttamente i dati avvelenati.

3.5.2 Fase di Testing

Siamo andati a suddividere le tecniche di difesa in questa fase in base all'approccio che adottano:

- Proattive, si concentrano sulla classificazione corretta delle immagini perturbate
- Reattive, rilevano e gestiscono immagini legittime o contraddittorie prima che raggiungano il classificatore.

Il metodo che è considerato dal documento [7] la prima misura di difesa contro gli attacchi in questa fase, soprattutti gli adversarial attacks, è l'**Adversarial Training**. E' una tecnica proattiva che prevede l'addestramento del modello su un set di dati ibrido che include sia immagini legittime che contraddittorie. Quindi impara a riconoscere sia i dati corretti che quelli malevoli.

3.6 Difese della Privacy

Oltre alla divisione delle difese in base a se avvengono per proteggere il training o il testing, il paper [7] introduce un'altra categorizzazione, ovvero le **Difese della Privacy**.

L'obbiettivo di queste è quello di non permettere ad un utente malintenzionato di accedere e rivelare informazioni sensibili sui dati del processo. Per quanto riportato in [7], questo tipo di difesa comprende tre metodi:

- Privacy Differenziale, tecnica per la quale viene inserito un elemento disturbante, chiamato rumore, nei dati sensibili, in modo da permettere l'anonimizzazione di questi già nella fase di acquisizione. Grazie a questo un possibili attaccante non sarà in grado di identificare o dedurre informazioni private sugli individui specifici.
- Crittografia Omomorfa, rappresenta il fatto di andare a crittografare i dati per poterli elaborare senza doverli decifrare, in quanto altrimenti significherebbe che in una parte del processo i dati sensibili saranno mostrati in chiaro e vogliamo evitarlo.
- Federated Learning.

3.6.1 Federated Learning

Per spiegare questa tecnica ci siamo riferiti si, al paper [7], ma soprattutto abbiamo fatto riferimento anche a [1], il quale introduce questa difesa, spiega nel dettaglio il funzionamento e la sua importanza. Si compone di due attori fondamentali:

- client
- server

In pratica, è un meccanismo di apprendimento collaborativo in cui non si ha più la fase di training del modello in un server centrale, ma questo processo viene distribuito su vari dispositivi, ovvero i client che detengono i dati sensibili. In questo modo gli input di addestramento saranno in chiaro solamente localmente, nei dispositivi che li posseggono. Dopo avere terminato il training dei modelli su ogni client, vengono inviati al server non i dati, ma i parametri elaborati e crittografati, che ha il compito di aggregarli.

In base al contesto sono state sviluppate più tipologie di apprendimento federato, tra cui

- Federated Horizontal Learning, i dati provengono da uno stesso dominio, ma da diversi individui, e presentano caratteristiche simili.
- Federated Vertical Learning, dati presentanti differenti caratteristiche ma raccolti da uno stesso insieme di individui.Come in ambito bancario per gli stessi utenti sono generati dati di differenti categorie.
- Transfer Learning, i dati non sono né con features simili, né raccolti sullo stesso insieme di utenti.

Purtroppo, quest'ultima difesa non è invincibile, ma deve essere affiancata dagli altri metodi di difesa della privacy, come spieghiamo approfonditamente nella sottosezione "Federated learning" del nostro documento originale. Per concludere l'argomento, enunciamo le sfide dell'apprendimento federato, discusse nel paper [1].

Considerazioni su direzioni future. Nonostante esistano difese contro gli adversarial examples, questi funzionano per una vastità di input, ci saranno quindi sempre degli input al "limite" che possono ingannare il modello. Inoltre, lo sviluppo di metodi di difesa dagli adversarial examples è teoricamente complesso. Nel tempo sono state proposte alcune difese, tuttavia, queste non sono adattive. Particolare attenzione dovrebbe essere posta nel campo del Federated Learning, sviluppando tecniche per l'aumento della privacy in fase di trasferimento del modello, trovando un compromesso tra prestazioni e velocità di comunicazione per l'abbattimento dei costi legati a quest'ultima. Inoltre, dovrebbero essere approfonditi nuovi metodi di convalida dei dispositivi distribuiti da cui si accetterà il modello, per evitare che si coinvolgano modelli con dati inaffidabili in fase di aggregazione.

References

[1] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705121000381

[2] H. Chen and M. A. Babar, "Security for machine learning-based software systems: a survey of threats, practices and challenges," 2023.

[3] B. Battista and R. Fabio, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, p. 317–331, Dec. 2018. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2018.07.023

[4] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar, "Adversarial machine learning," in *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*, ser. AISec '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 43–58. [Online]. Available: https://doi.org/10.1145/2046684.2046692

[5] J. Surma, "Hacking machine learning: Towards the comprehensive taxonomy of attacks against machine learning systems," in *Proceedings of the 2020 the 4th International Conference on Innovation in Artificial Intelligence*, ser. ICIAI '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1–4. [Online]. Available: https://doi.org/10.1145/3390557.3394126

[6] S. Seetharaman, S. Malaviya, R. Vasu, M. Shukla, and S. Lodha, "Influence based defense against data poisoning attacks in online learning," in *2022 14th International Conference on COMmunication Systems NETworkS (COMSNETS)*, 2022, pp. 1–6.

[7] A. Bajaj and D. Vishwakarma, "A state-of-the-art review on adversarial machine learning in image classification," *Multimedia Tools and Applications*, vol. 83, pp. 9351–9416, 2024. [Online]. Available: https://doi.org/10.1007/s11042-023-15883-z

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2015.