Università degli Studi Mediterranea di Reggio Calabria
Dipartimento di Giurisprudenza, Economia e Scienze Umane

Corso di Laurea Magistrale in
ECONOMICS LM-56

IL FENOMENO DEL "DATA POISONING" E L'IMPORTANZA DELL'APPRENDIMENTO NELLA MACCHINA IN AMBITO IA

Project Work in
Business Analytics

Prof. Massimiliano Ferrara

Project Work di
Caterina Itri
Domenico Naso
Giorgio Lo Giudice
Giovanni Dattola
Luigi Cozzupoli
Maria Carla Moio
Marta Pirrottina
Miriam Cusato
Pietro Rosato Forti
Silvia Strano

Anno Accademico 2023/2024

INDICE

INTRODUZIONE

L'intelligenza artificiale sta rivoluzionando il mondo, con i modelli di apprendimento automatico che influenzano sempre più la nostra vita quotidiana.
L'utilizzo sempre più diffuso degli algoritmi di Machine Learning ha portato progressi notevoli in innumerevoli settori, ma ha altresì aperto le porte a nuove minacce, come il fenomeno del *data poisoning*.
Si pensi ad un mondo in cui l'intelligenza artificiale ricopre un ruolo fondamentale in ogni aspetto della nostra vita, dai conti correnti bancari alle diagnosi mediche, gestendo quantità smisurate di dati; un mondo in cui ci si affida quotidianamente alle decisioni dell'algoritmo confidando nella sua veridicità ed imparzialità. Ma cosa accadrebbe se un individuo riuscisse a manipolarla, iniettando informazioni errate e fuorvianti nei suoi sistemi di apprendimento? Se venisse immessa benzina sporca nel motore di un'auto, si comprometterebbe il funzionamento mettendo a rischio la sicurezza dei passeggeri. Questo, analogamente, è il fenomeno del *data poisoning*: un attacco subdolo e dannoso che sfrutta la vulnerabilità dei modelli di machine learning per contaminarli nel loro interno.
Lo scopo di questo elaborato è quello di esplorare a fondo le diverse strategie utilizzate dai malintenzionati aggressori di sistemi di intelligenza artificiale, le conseguenze di suddetti attacchi, per poi esaminare le contromisure disponibili per la protezione degli algoritmi di machine learning.
Nei seguenti capitoli sono stati trattati più aspetti relativi ai modelli di machine learning, e come questi possano essere "contaminati" dagli attacchi di *data poisoning*.
In particolare, nel primo capitolo sono stati illustrati i modelli di apprendimento automatico con le loro logiche di funzionamento, le loro potenzialità e vulnerabilità; nel secondo capitolo è stato approfondito il fenomeno del *data poisoning*, ponendo l'enfasi sulle molteplici modalità di attacco, nonché sulle tecniche difensive mediante le quali rimediare alle malevole azioni degli aggressori.

L'INTELLIGENZA ARTIFICIALE E L'APPRENDIMENTO

L'INTELLIGENZA ARTIFICIALE E IL MACHINE LEARNING

"*Una forma di risparmio di lavoro è quello di usare algoritmi per ridurre al minimo il numero di passaggi necessari per ottenere il risultato desiderato*[1]".

Alan Turing, uno dei pionieri della scienza informatica e dell'intelligenza artificiale (IA), ha lasciato un'impronta indelebile nel mondo della tecnologia e della crittografia. Celebre per il suo concetto di "macchina universale", che ha gettato le basi per i moderni computer. La sua vita è stata segnata da tragici eventi personali, ma il suo genio ha contribuito in modo significativo allo sviluppo della tecnologia moderna; egli è noto per il suo lavoro fondamentale nell'ambito dell'intelligenza artificiale.
La sua concezione dell'intelligenza artificiale, come accennato, si basava sulla nozione di una "macchina universale". Egli fu il primo a porsi alcune domande e successivamente esplicitò, in un suo famoso articolo del 1950, interrogando il mondo con una domanda alquanto insolita all'epoca: "Le macchine possono pensare?"; fu così che propose il Test, denominato "Test di Turing", ossia un metodo per valutare l'intelligenza di una macchina in base alla sua capacità di sostenere una conversazione umana senza essere rilevata come non umana. Ciò ha avuto un impatto significativo non solo sull'evoluzione dell'intelligenza artificiale ma anche sulla comprensione dell'evoluzione stessa. La sua visione dell'intelligenza artificiale come una disciplina che potrebbe emulare il pensiero umano ha contribuito a plasmare il modo in cui consideriamo il rapporto tra macchine e intelligenza. Il suo lavoro pionieristico ha aperto la strada a una vasta gamma di ricerche nel campo dell'intelligenza artificiale, dalle reti neurali artificiali all'apprendimento automatico. Il lavoro da egli svolto ha influenzato profondamente lo sviluppo dell'intelligenza artificiale e gettando le basi di

[1] Alan Turing.

tutto quelle nozioni e informazioni oggi a disposizione, continuando a essere una fonte di ispirazione per gli scienziati informatici correnti.

Alla base dell'intelligenza artificiale vi è il *Machine Learning* (ML), il quale può essere considerato un sottoinsieme dell'intelligenza artificiale.
Il machine learning studia algoritmi e modelli matematici che consentono ai computer di imparare ad eseguire un compito particolare attraverso dati ed esperienza senza essere appositamente programmati. Dopo Alan Turing, altri studiosi, già a partire dagli anni '50, come Arthur Samuel e Tom Mitchell, hanno dato diverse definizioni di Machine Learning, ma la definizione per eccellenza è che essa si occupa di fare previsioni, "imparando" dai dati. Quanti più dati riceve il machine learning, tanto più precisa e completa sarà la sua risposta. Può capitare che il machine learning non riesca ad ottenere i risultati desiderati e in questi casi entra in gioco il c.d. deep learning (o apprendimento approfondito), il quale è un *subset* del machine learning, ovvero agisce dove il machine learning non riesce ad agire. Esso è, dunque, il contraltare dell'uomo, ovvero si concentra sull'utilizzo di reti neurali più complesse per riconoscere le relazioni tra input diversi, osservando strutture nascoste dei dati; ciò accade perché i dati sono meno utili e si hanno meno informazioni. Spesso, dovendo prendere delle decisioni, ci troviamo di fronte a delle situazioni senza avere una perfetta conoscenza delle informazioni. Nonostante ciò, la decisione si deve avere, e si crea una macchina che apprende le informazioni che possediamo. Il compito degli algoritmi di machine learning consiste nell'elaborazione di esempi descritti da caratteristiche o feature (soluzioni) fornite in input sottoforma di vettore di n elementi, dove n è il numero di feature.
I metodi attraverso i quali il machine learning riesce a recepire le informazioni possono essere classificati in diversi tipi di apprendimento:

- apprendimento supervisionato (*supervised learning*), dove le modalità machine learning attraversano tre fasi: formazione, convalida e test. In questo tipo di apprendimento si costruisce, nella fase della formazione, un modello a partire da

dati di addestramento etichettati. Pertanto, si considera y come la variabile di output e x come il vettore di caratteristiche dei campioni di ingresso. Tale modello deve capire come predire durante la fase di addestramento y da x, stimando la probabilità $p(y|x)$. L'obiettivo di questo tipo di apprendimento è riuscire a costruire modelli in grado di eseguire previsioni in condizioni d'incertezza, in modo tale da riuscire a predire la variabile y nel momento in cui viene fornito un nuovo input. Il motivo per il quale viene definito apprendimento supervisionato è dato dal fatto che l'algoritmo esegue iterativamente delle previsioni che, se errate, vengono corrette da un supervisore, finché il livello di precisione del programma non viene ritenuto sufficiente. Tale tipo di apprendimento trova applicazione nel settore medico, in cui sono in grado di prevedere lo scatenarsi di particolari crisi sulla base dell'esperienza di passati dati biometrici, per arrivare all'identificazione vocale o al riconoscimento delle immagini;

- apprendimento non supervisionato (*unsupervised learning*), si costruisce un modello a partire da dati di addestramento non etichettati o senza un corrispondente valore di output. L'obiettivo, in questo tipo di apprendimento, è mettere l'agente nella condizione di trovare pattern nascosti o strutture intrinseche nei dati che sono stati forniti. In questo modo è possibile fare deduzioni in base al gruppo di dati che si possiede, senza avere una possibile risposta esatta e un supervisore che corregga l'eventuale errore. Inoltre, viene utilizzato quando il problema richiede una quantità enorme di dati non etichettati e trova applicazioni nel mondo del marketing, in cui usando la tecnica del raggruppamento (clustering), si individuano gruppi di consumatori con caratteristiche simili a cui rivolgere campagne di marketing specifiche, oppure alla scoperta di caratteristiche che differenziano segmenti di consumatori dagli altri;
- apprendimento semi-supervisionato (*semi-supervised learning*), il quale rappresenta una via di mezzo i primi due tipi di apprendimento. In particolare,

per l'addestramento viene utilizzato un ridotto volume di dati classificati e un più ampio volume di dati non classificati e incompleto. Per tale motivo, può essere considerato un modello di apprendimento "ibrido". L'obiettivo principale è quello di identificare regole e funzioni per la risoluzione dei problemi, nonché modelli e strutture di dati utili al raggiungimento di determinati obiettivi. L'apprendimento semi-supervisionato viene utilizzato con metodi di classificazione, regressione e previsione;

- apprendimento per rinforzo (*reinforcement learning*), il cui obiettivo è costruire un sistema che, attraverso le interazioni con l'ambiente, migliori le proprie performance. Tale tipo di apprendimento è un paradigma automatico basato sull'analisi dei feedback, premi e penalità. Può essere definito "meritocratico", in quanto una ricompensa incoraggia i comportamenti corretti dell'agente. Dunque, l'algoritmo di reinforcement learning modifica il proprio comportamento per ricevere più premi e ridurre al minimo le punizioni. In questo modo, si viene a creare e a perfezionare un modello decisionale o di classificazione. Viene spesso usato nell'ambito della robotica e dei videogiochi, oltre che nello sviluppo delle automobili a guida autonoma che, proprio attraverso il machine learning, imparano a riconoscere l'ambiente circostante e adattare il loro comportamento in base alle specifiche situazioni che devono affrontare lungo il tragitto.

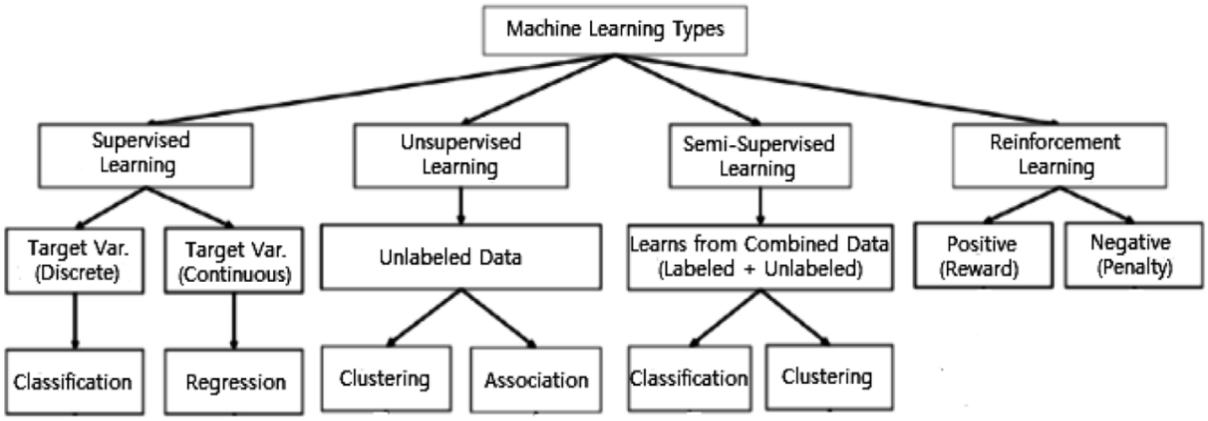

Figura 1. I diversi tipi di apprendimento.

LE SFIDE DEL MACHINE LEARNING

Come detto in precedenza, al machine learning servono grandi quantità di dati anche per risolvere problemi semplici. Un essere umano può apprendere con un solo esempio cosa è una mela, un computer ha bisogno di molti più esempi.
Pertanto, si ritiene che i dati siano più importanti degli algoritmi e proprio per tale motivo è necessario usare un training set costituito da dati che generalizzano bene i nuovi casi. Nel caso in cui si possiedono pochi dati, si può andare incontro a "*sampling noise*", ovvero dati non rappresentativi per caso. Nel caso in cui, invece, si possiedono troppi dati si va incontro a "*sampling bias*", in particolare quando sono selezionati male.
Un esempio può essere il caso delle elezioni presidenziali in USA nel 1936, dove era stato fatto un sondaggio che coinvolgeva dieci milioni di persone e di queste hanno risposto 2,4 milioni. Il dato vincente al 57% era Landon, ma alla fine alle elezioni presidenziali vinse Roosevelt con il 62%. Il problema che venne messo in evidenza e che condusse al risultato indicato era che la lista di persone coinvolte, anche se sufficientemente grande, era presa da rubriche telefoniche, iscrizioni a club, iscrizioni a magazine e a tali liste erano iscritte principalmente persone facoltose. Pertanto, da tale esempio possiamo dedurre che se i dati contengono errori o *outliers*, l'algoritmo di machine learning non funzionerà bene, per cui bisognerà pulire i dati, ad esempio scartando gli outliers.
Per quanto riguarda le *features* irrilevanti si possono attuare le seguenti procedure:

- *Feature selection*, ovvero selezionare quelle più utili tra tutte le possibili;
- *Feature extraction*, ovvero combinare più feature per produrne una più significativa.

Come specificato nel paragrafo precedente, il machine learning si basa su reti neurali ed il più grande vantaggio che queste apportano nel campo della computazione

artificiale è probabilmente la capacità di approssimare una funzione arbitraria imparando da un insieme di dati osservati.

Tuttavia, il funzionamento desiderato da una certa rete non è semplice da ottenere, e la progettazione costituisce un punto delicato dello sviluppo di un sistema di machine learning basato su reti neurali.

Pertanto, la progettazione può essere divisa in due scelte particolari: la scelta del modello e la scelta dell'algoritmo.

Per quanto concerne la scelta del modello, dipende dalla rappresentazione dei dati disponibili e dal tipo di applicazione per cui la rete viene progettata. Ciò avviene perché modelli troppo semplici potrebbero portare ad una fase di addestramento difficile da portare a termine. Tale situazione viene detta *underfitting*, in cui la rete fatica ad apprendere informazioni. Nel caso, invece, in cui vi sono modelli troppo complessi, questi potrebbero portare la rete a dipendere troppo dai dati di esempio, limitando le capacità di generalizzazione. Tale situazione è nota come *overfitting*, in cui la rete impara a gestire solo le informazioni contenute nei dati di esempio.

Per quanto concerne la scelta dell'algoritmo di apprendimento che evidenzia il fatto che esistono diversi compromessi di cui tener conto tra l'utilizzo di un determinato algoritmo piuttosto che di un altro. Generalmente, una volta scelti in modo corretto i c.d. iperparametri qualunque algoritmo lavora bene su un particolare set di dati. Gli iperparametri sono le variabili che caratterizzano il funzionamento e che sono scelte prima del processo di ottimizzazione.

Tuttavia, l'operazione di scelta di un algoritmo e dei suoi parametri per l'addestramento su nuovi dati richiede un abbondante quantità di sperimentazioni.

GLI ALGORITMI DI MACHINE LEARNING

Dopo aver rappresentato i principali modelli di apprendimento l'importanza scelta dell'algoritmo di apprendimento per la progettazione, e, in particolare, per l'utilizzo dei dati, è opportuno trattare i principali algoritmi di machine learning.

Per "algoritmo" intendiamo una successione di istruzioni o passi che definiscono le operazioni da eseguire sui dati per ottenere i risultati. In particolare, gli algoritmi di machine learning usano parametri basati su dati di training, ovvero un sottoinsieme di dati che rappresenta il set più grande. L'espansione dei dati di training per rappresentare il mondo in modo più realistico consente all'algoritmo di calcolare risultati più accurati.

Tra gli algoritmi più diffusi in ambito machine learning vi sono la classificazione, la regressione e il *clustering*.

Per quanto concerne la classificazione gli input sono divisi in due o più classi e il sistema di apprendimento deve produrre un modello in grado di assegnare ad un input una o più classi tra quelle disponibili. Questi tipi di task sono tipicamente affrontati mediante tecniche di apprendimento supervisionato[2].

La regressione è, invece, concettualmente simile alla classificazione con la differenza che l'output ha un dominio continuo e non discreto. Tipicamente è affrontata con l'apprendimento supervisionato[3].

Infine, il clustering, come nella classificazione, è un insieme di dati che viene diviso in gruppi che però, a differenza di questa, non sono noti a priori.

La natura stessa dei problemi appartenenti a questa categoria li rende tipicamente dei task di apprendimento non supervisionato.

Di seguito si rappresentano gli algoritmi più utilizzati:

---

[2] Un esempio di classificazione è l'assegnazione di una o più etichette ad una immagine in base agli oggetti/soggetti contenuti in essa.

[3] Un esempio di regressione può essere la stima della profondità di una scena a partire dalla sua rappresentazione sotto forma di immagine a colori.

- Il *Naïve Bayes Classifier* è un algoritmo di classificazione probabilistica che utilizza il teorema della probabilità di Bayes o le regole di Bayes. Presuppone che l'occorrenza di una determinata caratteristica sia indipendente all'occorrenza di altre caratteristiche e prevede la probabilità di appartenenza per ciascuna classe. La classe con la probabilità più alta viene considerata la regione più probabile;
- La regressione logistica ha lo scopo di determinare la probabilità del verificarsi di un evento, dati determinati fattori di rischio;
- Il *K-Nearest Neighbours* (KNN) è un algoritmo che si basa sul presupposto che punti simili possono essere trovati l'uno vicino all'altro. Vengono così selezionati i K vettori più vicini e si sceglie la classe più frequente tra quelle associate. Considerando che un valore di K piccolo porta ad *overfitting* e un valore di K elevato ad *underfitting*, è necessario usare la *Cross Validation* per valutare i diversi K e selezionare quello con le performance migliori;
- L'algoritmo *Random Forest* viene comunemente utilizzato per combinare l'output di più strutture ad albero decisionali per raggiungere un unico risultato.
- Il *Support Vector Machine* (SVM) è un classificatore binario non probabilistico che separa i dati attraverso un limite decisionale (piano nello spazio delle caratteristiche multidimensionali), ma la classificazione non può avvenire in più di due classi. Tuttavia, ci sono delle soluzioni alternative. I metodi implicano la creazione di più SVM che confrontano i vettori di funzionalità tra loro usando varie tecniche tra *one-versus-rest* (OVR) o *one-versus-one* (OVO). Il metodo OVR per K classi addestra K classificatori in modo che ogni classe discrimini le restanti K-1 classi. Il metodo OVO invece, crea un problema di classificazione binaria per tutti i possibili accoppiamenti di classi richiedendo K(K-1)/2 classificatori. Il SVM tratta ogni oggetto come un punto nello spazio delle caratteristiche che appartiene a una sola delle due classi, definendo che le etichette delle due classi sono yi=1 o yi= -1. Durante l'addestramento il classificatore SVM trova un limite decisionale nello spazio delle caratteristiche

che meglio separa gli oggetti dati nelle due classi. Inoltre, viene utilizzato un iperpiano lineare altrimenti uno non lineare tenderebbe ad *overfitting*, ovvero, separerebbe perfettamente i dati di training, ma non si adatterebbe ai nuovi dati;

- *Decision Tree* è un algoritmo che si può usare sia per la classificazione sia per la regressione. Parte dall'idea di divisione dello spazio delle *features* in regioni non sovrapposte. Quando arriva una nuova istanza da predire, si associa alla media dei valori che ci sono in quella regione (regressione) o alla classe più numerosa di valori che cadono in quella regione[4].

LA VULNERABILITA' DEL MACHINE LEARNING

In generale le metodologie legate al machine learning fanno uso di grandi capacità di calcolo dei moderni computer affiancati all'uso di algoritmi ottenendo delle prestazioni all'avanguardia in diverse applicazioni, dai sistemi finanziari alla sicurezza informatica. Questi metodi hanno bisogno di una grande mole di dati che siano principalmente adeguati e di alta qualità per poter apprendere le caratteristiche correttamente. Da qui nascono le prime vulnerabilità dei sistemi; nonostante gli alti livelli di performance spesso raggiunti, vi è un'ampia attenzione sul fatto che la loro affidabilità dovrebbe essere attentamente valutata. C'è il rischio che questi algoritmi prendano inconsapevolmente conclusione errate dovute principalmente a correlazioni anomale nei dati di addestramento soprattutto nell'era dei big data dove pervade l'uso dei modelli di machine learning. In relazione a questo, per aumentare la fiducia degli utenti e identificare eventuali vulnerabilità in questi approcci, si presta notevole attenzione nell'esplorare suddetti sistemi e sviluppare delle tecniche che rendano l'apprendimento robusto difronte a possibili attacchi che mirano ad ingannare l'algoritmo.

[4] Stimare lo stipendio di un giocatore di baseball sulla base di due features: il numero di anni in *Major League* e il numero di *hit* effettuati l'anno precedente all'analisi.

In relazione al tema del progetto, si sottolinea la necessità di un costante lavoro di monitoraggio e supervisione, poiché è importante essere consapevoli del potenziale rischio di *data poisoning* e includere valutazioni di sicurezza approfondite durante lo sviluppo e l'implementazione di sistemi di intelligenza artificiale, al fine di proteggere il modello e garantire la sua affidabilità e sicurezza nell'uso pratico. Non essendo immune, l'intelligenza artificiale può essere vulnerabile al *data poisoning*, ovvero una tecnica di attacco che consiste nel manipolare i dati utilizzati per addestrare un modello di machine learning al fine di influenzarne le prestazioni o persino compromettere la sicurezza del sistema. Il *data poisoning* può avere conseguenze significative, soprattutto se il modello viene utilizzato in contesti critici come la sicurezza informatica, la diagnosi medica o la guida autonoma. Per mitigare il rischio di *data poisoning*, sono necessarie varie misure preventive, come l'adozione di tecniche di rilevamento e filtraggio dei dati dannosi durante il processo di addestramento del modello, l'implementazione di controlli per garantire l'integrità e l'autenticità dei dati utilizzati e l'adozione di approcci robusti per la sicurezza informatica, come l'implementazione di meccanismi di autenticazione e autorizzazione.

IL DATA POISONING

COSA È IL DATA POISONING

Il primo capitolo ha investito il machine learning, strumento capace di migliorare la vita dell'uomo facilitando la raccolta e l'analisi dei dati, si pensi ad esempio all'attività di screening ambito in sanitario oppure all'attività di analisi dei dati aziendali per orientare al meglio il posizionamento strategico di un'impresa in rampa di lancio.
Il fatto che tali tecnologie siano ancora relativamente giovani espone questo ambito alle vulnerabilità intrinseche dei sistemi: a differenza dei tradizionali attacchi informatici causati da bug o errori umani nel codice, gli attacchi all'intelligenza artificiale sono abilitati da limitazioni degli stessi algoritmi. Si possono stabilire le caratteristiche degli algoritmi di machine learning alla base dell'intelligenza artificiale che rendono i sistemi vulnerabili agli attacchi:

- il machine learning funziona "apprendendo" pattern relativamente fragili che funzionano bene ma sono facili da distruggere. I modelli di machine learning non sono "intelligenti" o in grado di imitare veramente l'abilità umana nei compiti, ma lavorano imparando associazioni statistiche che sono relativamente facili da distruggere. Gli attaccanti possono sfruttare questa debolezza per creare attacchi che distruggono le performance di un modello altrimenti ben fatto;
- la dipendenza totale dai dati fornisce un canale principale per corrompere un modello di machine learning, esso "impara" esclusivamente estraendo modelli da una serie di esempi noti come dataset. A differenza degli esseri umani, i modelli di machine learning non hanno conoscenze di base da poter sfruttare; la loro intera conoscenza dipende interamente dai dati che vedono. Manomettere i dati a disposizione equivale a manomettere l'intero sistema;

Con l'aiuto dell'intelligenza artificiale, le aziende stanno scoprendo opportunità nascoste ed entrano in mercati inesplorati ma, insediandosi nei nuovi campi di applicazione della tecnologia sorgono, di conseguenza, dei nuovi rischi. Uno di questi

è il *data poisoning* che consiste in una minaccia alla sicurezza dei sistemi di intelligenza artificiale riferito alle componenti di machine learning: una modifica "malevola" dei dati di addestramento (*training data set*) di tali componenti può generare una distorsione dei risultati prodotti dal sistema in favore degli obiettivi perseguiti dall'aggressore informatico. Difendersi dagli attacchi di *poisoning* dei dati è un'attività complessa.

Il *data poisoning* è letteralmente l'avvelenamento dei dati, ossia vengono modificati i dati oggetto di analisi o vengono inseriti all'interno dell'insieme selezionato dei dati manomessi che modificheranno poi i risultati ottenuti. Ciò si verifica quando un utente malintenzionato introduce informazioni false, fuorvianti o dannose nei dati di addestramento con l'intenzione di alterare il processo di addestramento dell'algoritmo e compromettere la validità delle risposte, il processo decisionale e dunque della sua affidabilità. Gli attacchi di *data poisoning* possono avere molti obiettivi, tra cui la generazione di errori di previsione del modello, l'elusione dei sistemi di sicurezza e le decisioni algoritmiche. Le conseguenze di un errato addestramento dei modelli di machine learning possono essere molto dannose; uno studio del 2018 ha rivelato che un errore di costruzione di un sistema di machine learning ha indotto a diagnosticare melanomi laddove non ve n'era alcuna traccia. Altro esempio è ciò che è avvenuto a Google, in particolare al filtro antispam di gmail, che tra la fine del 2017 e l'inizio 2018 è stato oggetto di attacco ben quattro volte con l'obiettivo di rendere il classificatore incapace di riconoscere le mail spam da quelle non-spam.

Cosa può succedere in caso di *data poisoning*? Se si considera l'ipotesi di uno studente che utilizza un algoritmo per cercare degli articoli su un determinato tema riguardo il quale fare una tesi, il peggio che può succedere è fare un lavoro fuorviante o privo di ogni fondamento, ma se consideriamo il caso di modelli di intelligenza artificiale che governano il funzionamento di automobili autonome non più capaci di interpretare il codice della strada, allora la situazione diventa molto più grave.

Gli attacchi di *data poisoning* rientrano in due categorie principali: quelli che mirano a compromettere i modelli di apprendimento e quelli che mirano a compromettere

l'integrità dei dati. Il secondo è il più difficile da individuare e dunque il più pericoloso, poiché questo tipo di attacco manterrà intatta la maggior parte del database, il modello funzionerà come al solito e creerà una *backdoor*[5] che consentirà all'aggressore di controllare il modello. Inoltre, poiché le tecnologie di intelligenza artificiale si connettono sempre più direttamente a internet (si pensi alle *chatbots*[6]), questi sistemi raccoglieranno una quantità crescente di dati non strutturati che potrebbero non essere adatti all'utilizzo come base dati. Secondo alcuni esperti, infatti, il problema più grande dell'intelligenza artificiale è che la sua efficacia è quasi direttamente correlata alla qualità dei dati. Informazioni di bassa qualità portano a risultati negativi e, per le informazioni trovate in grandi quantità su un sito web pubblico, è impossibile determinare se il problema è dovuto alla corruzione intenzionale dei dati o alla qualità dei dati stessi.

Nella maggior parte dei casi l'approvvigionamento dei dati avviene direttamente sul web, ciò espone il programmatore a rischi significativi in quanto è molto semplice inquinare i dati limitandosi ad alterare le pagine web. In questa ipotesi ci possono essere due tipi di attacchi:

- *split-view poisoning*, quando l'aggressore agisce prendendo il controllo di una risorsa web avvelenando i dati raccolti e sabotando dunque i risultati dell'algoritmo;
- *front-running poisoning*, quando l'aggressore non ha il pieno controllo dello specifico set di dati, ma ha la possibilità di prevedere con precisione quando una risorsa web sarà accessibile per essere inclusa in un'istantanea del set di dati, avvelenandolo appena prima che le informazioni vengano raccolte.

[5] Una *backdoor* è un metodo, spesso segreto, per passare oltre (aggirare, bypassare) la normale autenticazione in un prodotto, un sistema informatico, un crittosistema o un algoritmo.

[6] Un *chatbot* è un software che simula ed elabora le conversazioni umane (scritte o parlate), consentendo agli utenti di interagire con i dispositivi digitali come se stessero comunicando con una persona reale (Oracle). https://www.oracle.com/it/chatbots/what-is-a-chatbot/

Il data poisoning può manifestarsi in diverse forme, a seconda dell'obiettivo dell'attaccante e del tipo di sistema di intelligenza artificiale che viene preso di mira. Alcuni esempi includono:

- l'inserimento di dati falsi: introduzione di esempi di addestramento ingannevoli per confondere l'algoritmo;
- la manipolazione di etichette: cambiare le etichette dei dati di addestramento per indurre errori di classificazione;
- gli attacchi mirati: modificare specifici frammenti di dati per alterare le decisioni del modello su input particolari.

Il *data poisoning* rappresenta una minaccia significativa per l'integrità e l'affidabilità dei sistemi basati sull'intelligenza artificiale e il machine learning. Questi rischi possono manifestarsi in vari modi, con conseguenze che vanno dal degrado delle prestazioni del modello fino a implicazioni etiche e legali molto serie. Introducendo deliberatamente nei dati di addestramento elementi che riflettono pregiudizi specifici, gli attaccanti possono indurre i modelli di intelligenza artificiale ad adottare questi stessi *bias* nelle loro decisioni. Questo può portare a discriminazioni sistematiche contro determinati gruppi o individui, sollevando questioni etiche significative e potenziali conseguenze legali per le organizzazioni che utilizzano tali sistemi.
In ipotesi in cui i modelli di intelligenza artificiale sono impiegati per la sicurezza o la privacy, come nei sistemi di riconoscimento facciale o nella protezione dei dati personali, il *data poisoning* può essere utilizzato per eludere le misure di sicurezza. Ad esempio, un utente malintenzionato può manipolare i dati per impedire ai sistemi di riconoscere tentativi di accesso non autorizzati o violazioni della privacy. Aziende e organizzazioni possono fare affidamento su modelli di intelligenza artificiale per attività critiche. Errori nelle previsioni, interruzioni del servizio e la necessità di riprogrammare i sistemi possono causare aumenti dei costi notevoli. Uno dei tipi di attacchi più diffusi di attacchi ai sistemi di intelligenza artificiale è la manipolazione delle immagini per falsificarne i modelli di classificazione. Un primo esempio di ciò è

Tay, la chat Twitter di Microsoft lanciata nel 2016. Twitter voleva che *Tay* fosse un bot in grado di interagire con gli utenti di Twitter. *Tay* lo ha fatto fino a quando un utente malintenzionato ha fatto apprendere al bot dei tweet contenti un linguaggio offensivo che ha mutato completamente il modo di interagire del bot sul social network. Di conseguenza Microsoft non ha potuto fare altro che rimuovere *Tay* dall'app.
Man mano che vengono rilasciati nuovi sistemi e funzionalità di intelligenza artificiale, ricercatori e professionisti stanno anche sviluppando difese contro ciascuno degli attacchi sopra elencati.

GLI ATTACCHI E L'AVVELENAMENTO DEI DATI - GLI EFFETTI SULLE PRESTAZIONI DELL'INTELLIGENZA ARTIFICIALE

Le prestazioni di un modello di apprendimento automatico dipendono fortemente dalla qualità e dalla quantità di dati con cui viene addestrato. Per addestrare un modello di apprendimento automatico accurato, è spesso necessaria una grande quantità di dati di addestramento e, al fine di ottenere dati di formazione sufficienti, i professionisti possono rivolgersi a fonti potenzialmente inattendibili. Questa diminuzione della qualità dei dati, specialmente se quei dati non sono stati sistematicamente controllati per verificare la correttezza delle sue etichette, apre la porta ad attacchi di avvelenamento dei dati in cui i dati etichettati deliberatamente erroneamente possono essere inseriti nel set di formazione di un modello con l'obiettivo di compromettere l'accuratezza di quel modello. Un attacco di avvelenamento dei dati mira a modificare un set di addestramento in modo tale che il modello addestrato utilizzando questo set di dati faccia previsioni errate. Gli attacchi di avvelenamento dei dati mirano a degradare il modello bersaglio al momento dell'allenamento, il che accade frequentemente. Gli attacchi di avvelenamento hanno un effetto duraturo perché compromettono l'integrità del modello e gli fanno commettere errori coerenti in fase di

esecuzione quando si fanno previsioni. Una volta che un modello è stato avvelenato, il recupero dall'attacco in un secondo momento è una procedura non banale.

Ci sono due diversi sotto-obiettivi per un attacco di avvelenamento dei dati:

- attacco *denial-of-service*, in cui l'obiettivo è ridurre le prestazioni di un modello bersaglio nel suo complesso. L'accuratezza predittiva del modello diminuirà per qualsiasi input (o la maggior parte degli input) ad esso inviato;
- attacco *backdoor/trojan*, in cui l'obiettivo è ridurre le prestazioni o forzare previsioni specifiche ed errate per un input o un insieme di input selezionati. L'accuratezza predittiva del modello diminuirà solo per gli input selezionati dall'attaccante. L'accuratezza è preservata per tutti gli altri input.

Si considera un semplice esempio di un modello progettato per rilevare gli ordini fraudolenti effettuati su un sito web di e-commerce.
Il modello dovrebbe essere in grado di prevedere se un ordine effettuato sarà pagato (“legittimo”) o meno (“fraudolento”) in base alle informazioni sull'ordine dato. Il set di formazione per questo modello è costituito da dettagli contenuti negli ordini storici inseriti sul sito web. Per avvelenare questo set di formazione, un utente malintenzionato si fingerebbe come utente o utenti del sito procedendo con gli ordini. L'attaccante paga per alcuni ordini e non paga per altri in modo tale che l'accuratezza predittiva del modello venga degradata quando viene addestrato. Nel caso dell'attacco *denial-of-service*, l'obiettivo è quello di far sì che il rilevatore di frodi prenda decisioni errate per qualsiasi ordine effettuato da qualsiasi utente. Nel caso dell'attacco backdoor/trojan l'obiettivo è quello di far sì che l'algoritmo del rilevatore di frodi prenda decisioni errate solo per gli ordini fraudolenti effettivi effettuati dall'attaccante.
Per eseguire un attacco *backdoor/trojan*, un aggressore effettua e paga una serie di ordini utilizzando il proprio account che contiene lo stesso nome, indirizzo e altri dettagli degli ordini fraudolenti che intendono effettuare in futuro. Questa cronologia degli ordini verrà, quindi, utilizzata per addestrare l'algoritmo di rilevamento delle

frodi. Successivamente, l'attaccante può utilizzare lo stesso account per effettuare nuovi ordini e non pagarli. Questi ordini fraudolenti saranno ritenuti legittimi dal modello e faranno perdere denaro al commerciante.

Gli attacchi di avvelenamento sono di solito realizzati utilizzando uno dei due metodi:

- l'iniezione di dati, ovvero iniettando nuovi dati nel set di addestramento, come sopra illustrato con l'attaccante che crea nuovi ordini. Questi nuovi dati possono essere di natura sintetica;
- il *label flipping* delle etichette, ovvero modificando le etichette dei dati reali esistenti nel set di formazione. Ciò può essere ottenuto pagando gli ordini con ritardo per cambiare il loro stato da fraudolento a legittimo nell'esempio considerato. In altri casi, le etichette possono essere sovrascritte attraverso i meccanismi di assistenza clienti.

I requisiti per eseguire questi due attacchi sono diversi. Per eseguire l'iniezione di dati, un utente malintenzionato deve solo interagire con il sito di e-commerce stesso per generare punti dati sintetici ed etichette corrispondenti. Per eseguire il ribaltamento delle etichette, un utente malintenzionato deve essere in grado di modificare le etichette dei punti dati reali già esistenti nei dati di addestramento. Sebbene la maggior parte degli aggressori non abbia i mezzi per modificare direttamente questi dati, nel nostro scenario di rilevamento delle frodi, un utente malintenzionato potrebbe utilizzare i meccanismi di assistenza clienti per inviare feedback sulle transazioni classificate "errate" al fine di manipolare le etichette storiche.
L'avvelenamento dei dati può essere eseguito in modo ad-hoc utilizzando l'iniezione di dati o il capovolgimento dell'etichetta. Quando si esegue l'iniezione di dati, l'attaccante può fare ipotesi su come il modello fornisce previsioni e quindi creare e iniettare nuovi dati che dovrebbero cambiare queste previsioni. Nell'esempio considerato di rilevamento delle frodi, l'attaccante paga per ordini simili a quelli che intende effettuare (ma non pagare) in seguito. Ciò che rende gli ordini simili può essere assunto solo in

questo caso e l'impatto previsto dell'iniezione sulla decisione del modello non può essere previsto. Nel ribaltamento dell'etichetta, l'attaccante può modificare le etichette dei punti dati scelti casualmente nel set di addestramento per raggiungere lo stesso obiettivo. Ancora una volta, mentre sappiamo che capovolgendo l'etichetta di abbastanza punti dati, degraderemo l'accuratezza del modello, l'impatto esatto di questo attacco casuale non può essere previsto.

Poiché i modelli di apprendimento automatico sono costruiti utilizzando algoritmi standard che risolvono problemi di ottimizzazione ben definiti, è possibile montare attacchi di avvelenamento dei dati più efficaci. Data una certa conoscenza del modello e dei suoi dati di addestramento, un utente malintenzionato può generare punti dati di addestramento sintetici che degraderanno in modo ottimale l'accuratezza del modello. L'attaccante può quindi ottimizzare un attacco di avvelenamento iniettando il numero minimo di punti dati necessari per raggiungere il loro obiettivo di avvelenamento, che si tratti di *denial-of-service* o *backdooring*.

Per generare dati efficaci di avvelenamento, è necessario iniziare considerando come viene utilizzato un set di dati per addestrare un modello di apprendimento automatico. L'addestramento di un modello comporta l'ottimizzazione di una funzione oggettiva che in genere mira a ridurre al minimo la distanza tra le previsioni fatte dal modello e le vere etichette nei dati di allenamento. Attraverso questo processo, il modello impara a fare previsioni corrette sui punti dati di allenamento. Queste previsioni corrette dovrebbero generalizzarsi in seguito a nuovi dati di test (che non fanno parte del set di formazione). L'addestramento efficace del modello si basa sul presupposto che i dati di addestramento siano simili o provengano dalla stessa distribuzione dei dati di test che verranno forniti al modello in fase di esecuzione. L'obiettivo di un attacco di avvelenamento è quello di sfidare questa ipotesi modificando il set di addestramento in modo che non corrisponda alla vera distribuzione dei dati di prova. Ciò si ottiene più

spesso aggiungendo dati con etichette errate al set di addestramento (utilizzando l'iniezione di dati o il capovolgimento dell'etichetta).

Sapendo che la funzione oggettiva utilizzata durante l'addestramento del modello mira a ridurre al minimo l'errore di allenamento (errore di previsione sui dati di addestramento), un utente malintenzionato può generare nuovi punti dati di addestramento con l'obiettivo di massimizzare l'errore di test (errore di previsione sui dati di test). Un punto dati di avvelenamento ottimale x_p può essere generato algoritmicamente per un determinato modello risolvendo un problema di ottimizzazione *bi-level* in cui:
(a) minimizza l'errore di addestramento
(b) massimizza l'errore di test su un set di test scelto dall'attaccante.
Ciò significa che il punto dati di avvelenamento x_p viene generato in modo tale che, quando viene utilizzato insieme a un set di addestramento predefinito per addestrare un particolare modello, il modello risultante massimizzerà l'errore di test sul set di test scelto dall'attaccante. Questo problema può essere risolto in modo iterativo utilizzando l'ottimizzazione della salita del gradiente per modelli addestrati, ovvero impiegando algoritmi basati su gradienti come SVM, regressione logistica o reti neurali (profonde). Per ottimizzare la generazione di input avvelenati, è necessaria una certa conoscenza del modello che si cerca di avvelenare. Idealmente, si deve conoscere:

- il set di formazione originale in cui inietteremo punti dati sull'avvelenamento;
- il tipo di modello da avvelenare (SVM, regressione logistica, ecc.);
- gli iperparametri del modello;
- la funzione di perdita (funzione oggettiva) utilizzata per calcolare l'errore di previsione durante l'allenamento.

Successivamente, si spiega come generare dati avvelenati algoritmicamente risolvendo il problema di ottimizzazione *bi-level* descritto sopra. In questo esempio, si utilizza un

set di formazione composto da 200 punti dati, appartenenti a 2 diverse classi (blu e rosso), ciascuna rappresentata da 2 caratteristiche (per scopi di visualizzazione).

Si utilizza questo set di dati per addestrare un modello SVM (con *kernel* RBF). La figura illustra i punti dati e i confini decisionali del modello. Questo modello SVM è stato valutato utilizzando i dati di prova su cui ha riportato un'accuratezza di previsione del 95%.

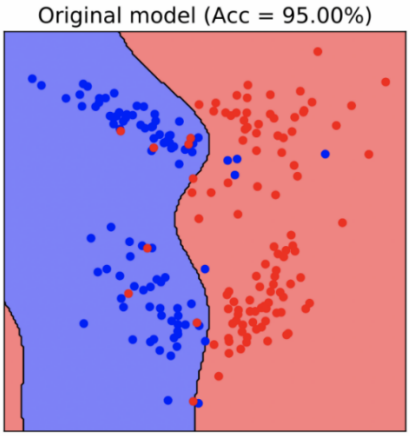

Figure 2: 200 training datapoints and decision boundaries of an SVM model trained using them

Nel primo esempio, si illustra come creare un attacco di avvelenamento *denial-of-service* contro questo modello. Si comincia utilizzando il *test-set*, il modello SVM addestrato e il set di formazione per definire la funzione oggettiva. Quest'ultima calcola l'errore di prova in funzione dei punti dati sull'avvelenamento da iniettare nel set di allenamento.

La Figura 3 mostra il valore della funzione oggettiva in base alle coordinate del punto dati che si iniettano nel set di allenamento. Per generare un punto dati avvelenato, si sceglie un punto casuale nello spazio – triangolo nero – e gli si assegna un'etichetta/classe casuale – in questo caso blu. Dopodiché, si cambiano iterativamente le sue coordinate – valori delle caratteristiche – utilizzando l'ottimizzazione dell'ascesa del gradiente per massimizzare il valore della funzione obiettivo, cioè per massimizzare

l'errore di prova. Si osserva che, attraverso diverse iterazioni, il punto dati di avvelenamento iniziale – stella blu – si sposta da un'area blu scuro – corrispondente a un basso valore della funzione obiettivo – ad un'area rossa – corrispondente a un valore elevato della funzione obiettivo – cioè, questo punto dati massimizzerà l'errore di previsione sui dati di prova considerati.

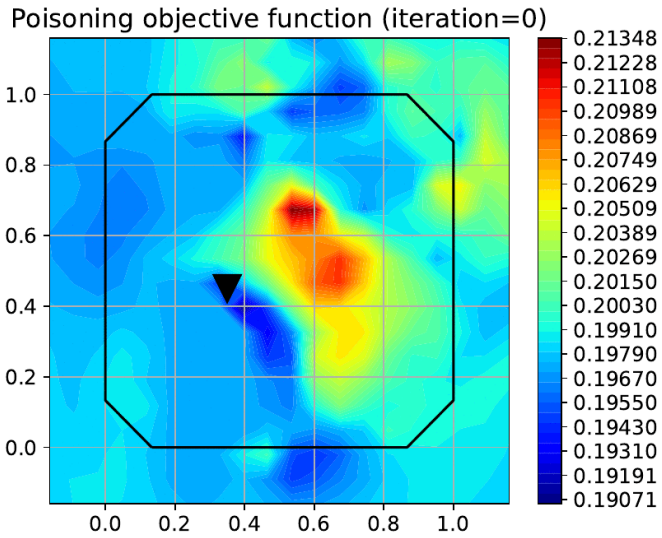

Figura 3: Funzione oggettiva per un attacco di avvelenamento denial-of-service e generazione di un punto dati avvelenato.

Dopo una serie di iterazioni, l'algoritmo definisce i valori ottimali per le due caratteristiche del punto dati avvelenato al fine di degradare al massimo la precisione del modello SVM. Le Figure 4 e 5 mostrano la posizione del nuovo punto dati rispetto ai dati di addestramento originali e come i confini decisionali del modello SVM sono stati influenzati da esso. Il nuovo punto dati blu – indicato da una stella – appare nella classe rossa – etichettata errata. Questo punto avvelenato si trova vicino ad altri punti blu etichettati erroneamente nell'area rossa, una posizione ottimale per modificare il confine decisionale del modello SVM. Si può notare che questo singolo punto dati ha modificato il confine decisionale del modello SVM nelle sue vicinanze e ha avuto scarso effetto sull'accuratezza del modello. Generalmente, al fine di ridurre l'accuratezza del modello, è necessario aggiungere più punti dati sull'avvelenamento.

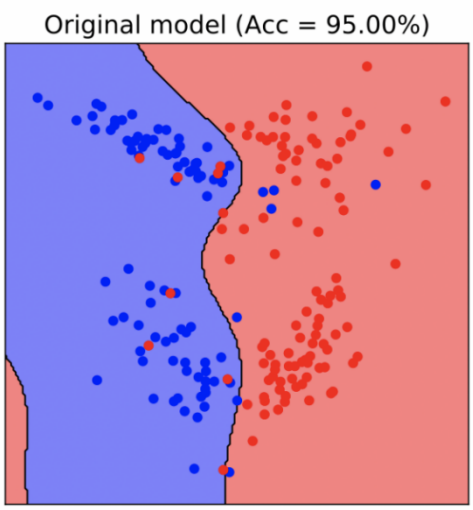

Figure 4: Original SVM model trained with poisoned data point

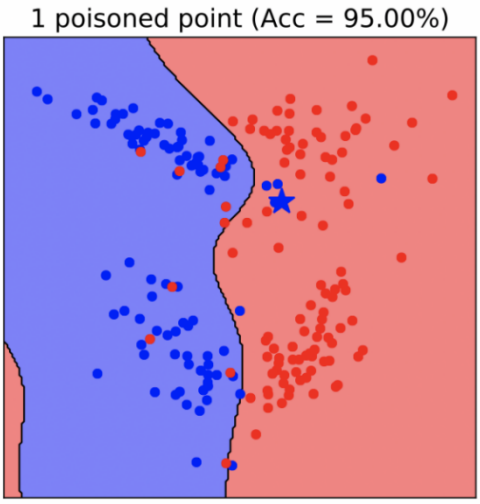

Figure 5: Poisoned SVM model trained with poisoned data point

Questo metodo automatizzato di generazione di dati sull'avvelenamento può essere applicato al precedente esempio di sistema di rilevamento delle frodi. L'ottimizzazione definirebbe automaticamente il valore ottimale per ogni caratteristica che rappresenta l'ordine di avvelenamento da inserire (nome, indirizzo, articoli, quantità, ecc.). Quindi, l'utente malintenzionato, potrebbe effettuare un ordine con questi valori esatti e pagare o meno secondo l'etichetta del punto generato. Questo metodo per generare punti dati di avvelenamento ottimizza l'attacco e rende il suo successo più prevedibile.

Si è visto come generare algoritmicamente un singolo punto dati di avvelenamento risolvendo un problema di ottimizzazione a due livelli. Per eseguire un attacco che raggiunga gli obiettivi di avvelenamento, si dovrebbero generare diversi punti dati di questo tipo utilizzando lo stesso problema di ottimizzazione. Tutti questi punti dati avvelenati devono quindi essere aggiunti al set di allenamento originale. Man mano che più punti di avvelenamento vengono aggiunti al set di allenamento, il confine decisionale del modello avvelenato cambia in modo tale che alla fine fornirà gli errori di previsione presi di mira dall'attaccante.

Nelle Figure 6 e 7, si illustra questo processo su un modello di regressione logistica e un modello SVM utilizzando un attacco di avvelenamento *denial-of-service*.

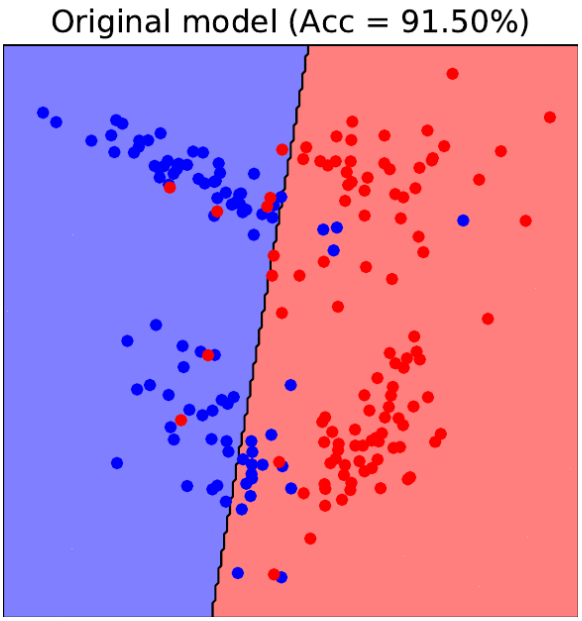

Figure 6: DoS poisoning attack against LR

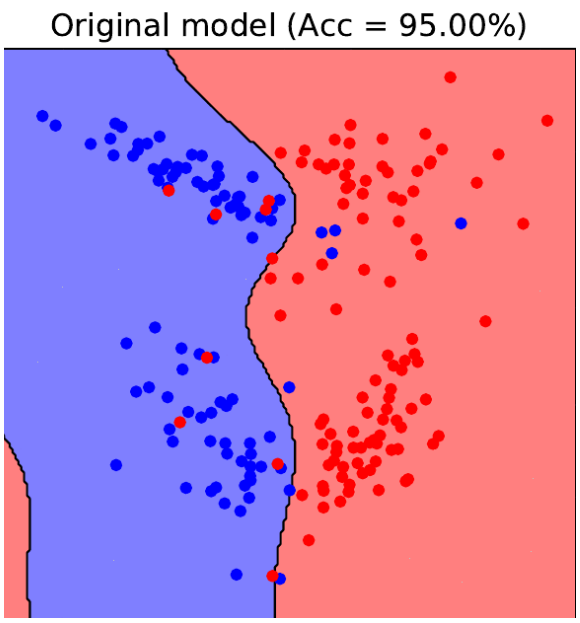

Figure 7: DoS poisoning attack against SVM

Come illustrato nelle figure 6 e 7, i punti di avvelenamento ottimali sono diversi per svariati modelli addestrati utilizzando lo stesso set di dati:

- la regressione logistica costruisce un confine decisionale lineare che è sensibile ai punti di avvelenamento situati il più lontano possibile dalla loro classe reale: la parte sinistra dello spazio per la classe rossa e la parte destra dello spazio per la classe blu;
- SVM (con *kernel* RBF) costruisce un confine decisionale non lineare che, per essere modificato, richiede più punti di avvelenamento sparsi.

L'attacco *denial-of-service* è molto efficace contro il modello di regressione logistica. Aumentando il numero di punti di avvelenamento, il confine di decisione viene progressivamente inclinato causando un calo della precisione del test dal 91,5% al 53%. Questo attacco di avvelenamento può essere considerato efficace contro il modello di regressione logistica. D'altra parte, mentre il confine decisionale del

modello SVM cambia in modo significativo man mano che vengono aggiunti più punti di avvelenamento, la sua precisione diminuisce dal 95% al solo 81,5%. I modelli SVM con *kernel* RBF costruiscono confini decisionali più complessi rispetto ai modelli di regressione logistica e quindi possono adattarsi meglio a punti dati avvelenati con etichette contraddittorie, pur mantenendo previsioni corrette sui punti dati puliti e autentici.

In generale, ciò significa che i modelli complessi (modelli con un'elevata capacità) sono più resistenti agli attacchi di avvelenamento *denial-of-service* rispetto ai modelli semplici, in quanto richiedono più punti dati di avvelenamento da compromettere ed è difficile ridurre la loro accuratezza nel complesso.

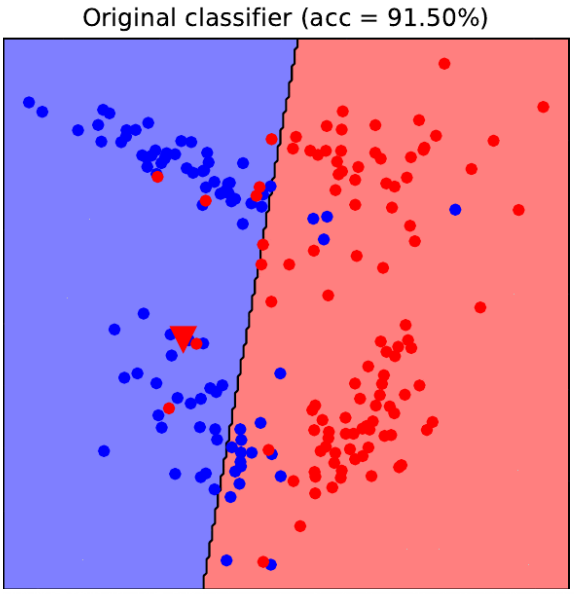

Figure 8: Backdoor poisoning attack against LR model

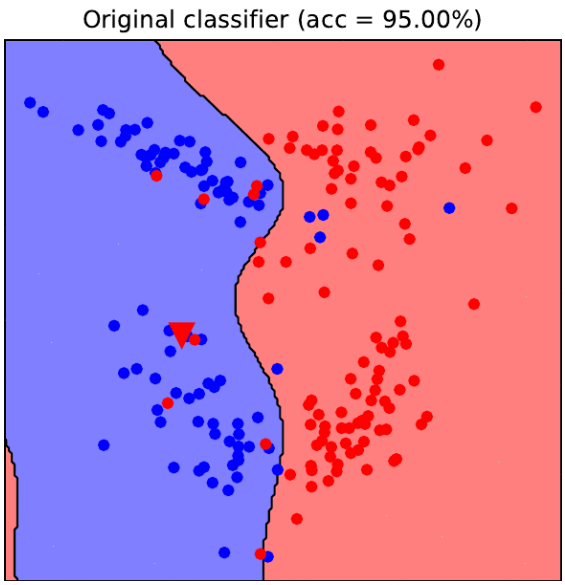

Figure 9: Backdoor poisoning attack against SVM model

Le Figure 8 e 9 illustrano come i confini decisionali di un modello di regressione logistica e di un modello SVM cambiano durante un attacco di avvelenamento *backdoor*. In questo esempio, si è scelto un punto dati dalla classe blu (*true label*) che si voleva che il modello prevedesse come classe rossa (punto *backdoor*). Questo punto è raffigurato come un triangolo rosso. Si nota che l'attacco di avvelenamento *backdoor* ha richiesto meno punti dati sull'avvelenamento per avere successo – 21 per la

regressione logistica e 12 per SVM. Per questo attacco, è necessario generare solo i punti dati di avvelenamento della classe bersaglio (rosso), che sono complessivamente più raggruppati rispetto all'attacco *denial-of-service*. A differenza di quest'ultimo, l'attacco *backdoor* è più efficace contro il modello SVM piuttosto che contro il modello di regressione logistica per la stessa ragione data sopra. Poiché i modelli SVM con kernel RBF hanno una capacità maggiore rispetto ai modelli di regressione logistica, il loro confine decisionale può adattarsi meglio alle anomalie nel set di formazione e creare "eccezioni" nelle loro previsioni. D'altra parte, richiede più punti dati avvelenati per spostare il confine decisionale lineare del modello di regressione logistica per adattarsi a queste anomalie.
La ragione di ciò è che la previsione del modello di regressione logistica dovrà contraddire l'etichetta per un gran numero di punti dati puliti nel set di formazione, e quindi sono necessari più punti dati avvelenati per affrontare questo problema.

PROPOSTE PER CONTRASTARE GLI ATTACCHI DI DATA POISONING

Ci sono diverse misure che i ricercatori consigliano di adottare per cercare di difendere i modelli dagli attacchi di *data poisoning*. Come già osservato nel paragrafo sulla vulnerabilità, esse riguardano soprattutto la protezione dei dati, l'addestramento del modello per renderlo più robusto e il miglioramento della sicurezza informatica.

Per proteggere i dati possono essere implementati dei controlli degli accessi per far sì che solo le persone autorizzate possano accedere al dataset. Anche la crittografia si rivela utile per garantire la sicurezza dei dati durante la trasmissione degli stessi. Meritevoli di protezione sono soprattutto i dati sensibili in quanto se divulgati possono diventare un punto di ingresso per immettere i dati avvelenati e per questo deve essere

garantita la protezione della rete, della struttura, degli *endpoint*[7] e delle persone[8]. Non meno importante è l'utilizzo di fonti sicure dalle quali raccogliere i dati e diversificare quest'ultimi per avere una rappresentazione più realistica della realtà.
Nella fase di preelaborazione dei dati spesso si creano ulteriori campioni per diversificare il dataset, si introduce il *random noise* e si standardizzano i dati per evitare che gli aggressori sfruttino le variazioni di scala per attaccare. Inoltre, si utilizza la convalida incrociata per creare più *fold* sia per l'addestramento che per la fase di test in modo che il modello riesca a generalizzare bene.

Nella fase di progettazione e addestramento per rafforzare il modello si modificano le architetture, si regolano gli iperparametri e si cerca di ridurre l'impatto degli attacchi allenando il modello su dati di addestramento suddivisi in piccoli campioni e preparati come descritto in precedenza, testando anche la capacità del modello di rispondere bene alle variazioni dei dati. Inoltre, il monitoraggio continuo permette di effettuare una verifica delle prestazioni del modello in quanto una loro riduzione può indicare la presenza di un possibile attacco che si può mitigare utilizzando delle tecniche di rilevamento di dati anomali con successiva eliminazione.

Tra le tecniche più utilizzate per mitigare gli effetti di un attacco vi è la data *sanitization*, ossia la sanificazione dei dati di addestramento effettuata eliminando quelli dannosi. Quando quest'ultimi presentano delle caratteristiche diverse rispetto a tutti gli altri è possibile utilizzare delle tecniche di clustering o di rilevamento degli *outliers* per individuarli ed eliminarli. Tuttavia, per alcuni attacchi questa tecnica non può essere utilizzata. A volte è lo stesso modello ad essere sanificato individuando e rimuovendo eventuali backdoor.

[7] Computer desktop, server, dispositivi mobili che si connettono e scambiano informazioni con una rete.

[8] Sono i quattro principi informatici di base indicati dalla Cyber Maturity Model Certification che garantiscono la protezione dei dati sensibili come indicato sul sito https://blogs.manageengine.com/active-directory/log360/2024/01/30/data-poisoning-prevention-strategies-to-keep-your-data-safe.html#:~:text=Model%20monitoring%2C%20routine%20data%20validation,software%20applications%20depend%20on%20this

Per potersi difendere implementando queste procedure è fondamentale conoscere il modello in modo approfondito ma anche avere accesso all'insieme di addestramento, all'insieme di test, agli iperparametri e alla procedura di addestramento.

Da quanto detto nel paragrafo emerge che per difendersi da un attacco e ridurre gli effetti è necessario proteggere tutte le fasi della formazione di un modello dalla raccolta e gestione dei dati fino al monitoraggio delle prestazioni ma è la fase di addestramento quella che può permette di rendere il modello capace di essere accurato anche in presenza di dati manipolati.

CONCLUSIONI

Alla conclusione di questo project work, si possono trarre delle considerazioni riguardanti i temi affrontati nei due capitoli. In primo luogo, l'analisi delle criticità del *machine learning* porta ad affermare che il fenomeno del *data poisoning* è un potenziale problema del presente-futuro. Una tecnica di attacco che va a manipolare i dati utilizzati per l'addestramento di un modello di machine learning allo scopo di influenzare le prestazioni o compromettere la sicurezza è sicuramente un fenomeno da tenere d'occhio.

L'analisi che si è messa in luce è quella degli attacchi *denial-of-service* e *backdoor* (i primi riducono l'*accuracy* del modello nel suo complesso; i secondi lo fanno per input specifici). Sono stati rappresentati entrambi gli attacchi a modelli di SVM e regressione logistica, per notare le differenze: l'attacco *denial-of-service* è più efficace contro il modello di regressione logistica; l'attacco *backdoor* è più efficace contro il modello di SVM. Il motivo di questi risultati è quello per cui i modelli complessi sono più resistenti ad attacchi *denial-of-service*, in quanto richiedono più punti dati avvelenamento da compromettere ed è difficile ridurre l'*accuracy* nel suo complesso.
Il grande problema del *data poisoning* è quello per cui una volta che il modello viene avvelenato, il recupero di quest'ultimo non è una procedura banale.

Le proposte per contrastare gli attacchi sono l'adozione di tecniche di rilevamento e filtraggio dei dati dannosi durante il processo di addestramento, l'implementazione di controlli, la tecnica di *sanitization* e l'adozione di approcci robusti per la sicurezza informatica.

REFERENCES

Biggio B., Cinà E.A., Demontis A., Grosse K., Pelillo M., Roli F., *Machine Learning Security aganist Data Poisoning: Are We There Yet?*, 2024

Cinà A. E. et al, Wild Patterns Reloaded: *A Survey of Machine Learning Security against Training Data Poisoning ACM Computing Surveys*, 2023

Heinrich K., Janiesch C., Zschech P., *Machine learning and deep learning*, 2021

Sarker I.H., *Machine Learning, Real-World Applications and Research Directions*, 2021

https://www.cybersecurity360.it/outlook/data-poisoning-pericolo-ai/

https://www.diritto.it/intelligenza-artificiale-data-poisoning-avvelenare/

https://www.agendadigitale.eu/sicurezza/data-poisoning-cosi-si-inquina-lintelligenza-artificiale/

https://www.cybersecurity360.it/nuove-minacce/lintelligenza-artificiale-puo-essere-attaccata-cose-il-data-poisoning-e-come-difendersi/

https://www.giurismatico.it/il-data-poisoning-e-il-suo-impatto-sullecosistema-ai/

https://labs.withsecure.com/publications/data-poisoning-in-action

https://www.glasswall.com/blog/defending-the-future-a-guide-to-fortifying-ai-against-data-poisoning-attacks