UNIVERSITÀ DI PISA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Laurea Triennale in Ingegneria Informatica

Analisi del traffico di rete e riconoscimento spiegabile dei relativi attacchi informatici

Relatore:
Ing. Antonio Luca Alfeo
Prof. Mario G.C.A. Cimino

Candidato:
Leonardo Ceccarelli

ANNO ACCADEMICO 2023/2024

Sommario

L'Intelligenza Artificiale sta rivoluzionando il settore della cybersecurity, rendendo possibile il rilevamento di intrusioni nei sistemi di rete con tecniche avanzate di machine learning. I tradizionali sistemi di Intrusion Detection (IDS), basati su regole statiche, spesso faticano a individuare attacchi sofisticati e in continua evoluzione. Per questo motivo, l'integrazione di modelli di apprendimento automatico rappresenta una soluzione promettente per migliorare l'accuratezza e la reattività nella protezione delle reti informatiche.

Questa tesi si concentra sull'applicazione di modelli di machine learning per la rilevazione delle intrusioni, con un focus sull'interpretabilità delle decisioni attraverso l'uso dell'Explainable AI (XAI). In particolare, sono stati analizzati i dati del dataset NSL-KDD, un benchmark ampiamente utilizzato nella ricerca sulla sicurezza informatica. I modelli implementati includono *RandomForest* e *GradientBoosting*, due approcci supervisionati che consentono di classificare il traffico di rete come normale o malevolo.

L'obiettivo dello studio è valutare l'accuratezza dei modelli, identificare le feature più rilevanti per la classificazione e analizzare la capacità della XAI di rendere il processo decisionale dei modelli più trasparente e comprensibile. Attraverso l'uso di tecniche come SHAP (SHapley Additive exPlanations) e Permutation Importance, è stato possibile comprendere quali fattori incidano maggiormente sulla rilevazione delle intrusioni.

I risultati ottenuti dimostrano che l'utilizzo della XAI non solo migliora la fiducia nell'efficacia dei modelli di Intrusion Detection, ma consente anche di individuare pattern di attacco ricorrenti, supportando così analisti di sicurezza e amministratori di rete nel migliorare le difese informatiche.

Indice

Capitolo 1

Introduzione

Negli ultimi anni, l'evoluzione tecnologica ha portato a una crescente interconnessione tra dispositivi e reti informatiche. Questo ha favorito la digitalizzazione di processi industriali, finanziari e amministrativi, ma ha anche esposto sistemi informatici a **minacce sempre più sofisticate**. Gli attacchi informatici, come il furto di dati sensibili, il sabotaggio di infrastrutture e la diffusione di malware, rappresentano una sfida critica per aziende e istituzioni governative.
L'aumento del numero di dispositivi connessi, unito alla crescita delle tecnologie IoT e al cloud computing, ha moltiplicato le superfici di attacco per i cybercriminali. Gli aggressori informatici sfruttano vulnerabilità nei sistemi per ottenere accesso non autorizzato, interrompere operazioni critiche o esfiltrare dati sensibili. Di fronte a queste minacce, la **cybersecurity tradizionale** basata su firewall e antivirus non è più sufficiente a garantire la protezione delle infrastrutture digitali.
Per far fronte a questi rischi, le organizzazioni si affidano a sistemi di **Intrusion Detection (IDS)**, ovvero strumenti in grado di monitorare il traffico di rete e rilevare attività sospette o malevole. Tuttavia, i metodi tradizionali di rilevamento delle intrusioni, basati su *regole predefinite e firme statiche*, presentano numerosi limiti. Questi sistemi faticano a identificare **attacchi zero-day** (nuove minacce non ancora documentate) e spesso generano **falsi positivi**, creando inefficienze nella gestione della sicurezza. Inoltre, con l'aumento del volume di dati da analizzare, il rilevamento basato su regole diventa meno scalabile e più difficile da mantenere.
Un ulteriore problema dei sistemi di rilevamento tradizionali è la loro incapacità di adattarsi a nuovi schemi di attacco. Ad esempio, un attacco di tipo **Advanced Persistent Threat (APT)**, caratterizzato da attività malevole prolungate nel tempo e difficili da rilevare, può facilmente eludere un IDS statico. Inoltre, i cybercriminali utilizzano tecniche di **evasion**, come la crittografia del traffico o il mascheramento degli indirizzi IP, per rendere inefficaci i sistemi di rilevamento basati su regole fisse.

1.1 Il ruolo dell'Intelligenza Artificiale nella cybersecurity

L'**Intelligenza Artificiale (IA)** ha rivoluzionato il campo della sicurezza informatica grazie alla sua capacità di analizzare grandi volumi di dati e riconoscere schemi anomali che potrebbero indicare un'intrusione. L'uso del **Machine Learning (ML)** ha permesso di sviluppare **sistemi di rilevamento adattivi**, in grado di individuare attacchi noti e nuove minacce analizzando il comportamento del traffico di rete.
A differenza dei tradizionali IDS basati su firme, i modelli di **Machine Learning** possono:

- **Apprendere** dalle minacce precedenti per migliorare le capacità di rilevamento.
- **Generalizzare** a nuovi tipi di attacchi senza richiedere aggiornamenti manuali.
- **Ridurre il numero di falsi positivi**, migliorando l'efficienza operativa.

Uno degli aspetti più interessanti dell'IA applicata alla cybersecurity è la capacità di **identificare anomalie** nel traffico di rete anche in assenza di un pattern di attacco noto. Modelli di **apprendimento non supervisionato** possono rilevare comportamenti sospetti senza essere addestrati su attacchi specifici, offrendo un vantaggio significativo rispetto ai metodi tradizionali.
Un esempio pratico è il rilevamento di attacchi **DDoS (Distributed Denial of Service)**, in cui un gran numero di richieste viene inviato a un server per sovraccaricarlo e renderlo inutilizzabile. Un modello di Machine Learning, analizzando il traffico di rete, può individuare un picco anomalo di richieste in un breve periodo e segnalare tempestivamente la minaccia.

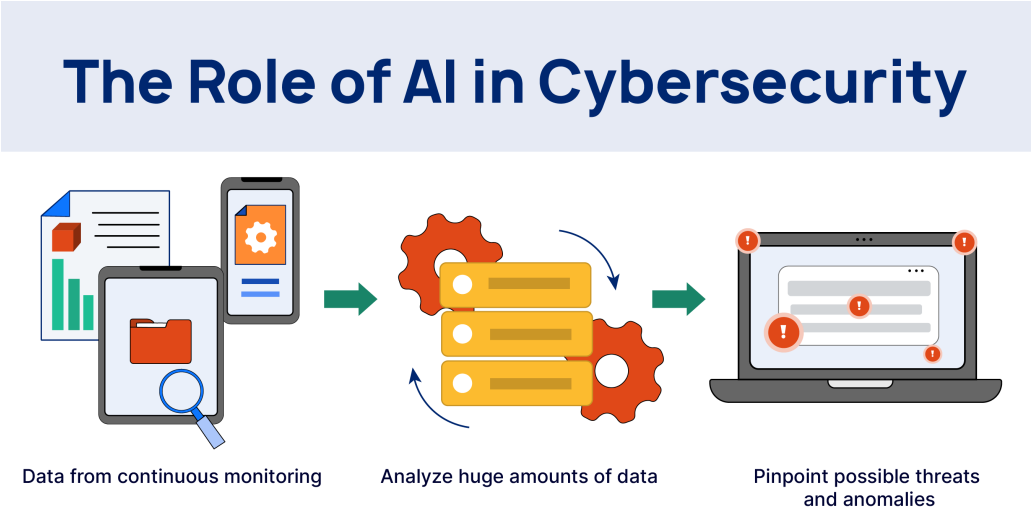

Figura 1.1: Il ruolo dell'AI nella Cybersecurity

1.2 L'importanza della Explainable AI (XAI) per la sicurezza

In questo contesto, l'**Explainable Artificial Intelligence (XAI)** gioca un ruolo fondamentale. Spesso, i modelli di IA utilizzati per il rilevamento delle intrusioni sono delle *black box*, ovvero prendono decisioni senza offrire spiegazioni comprensibili per gli analisti di sicurezza. Questo rappresenta un problema critico, in quanto la mancanza di trasparenza può rendere difficile l'interpretazione degli alert e la loro gestione operativa.
Le tecniche di **XAI** permettono di interpretare i risultati dei modelli, rendendo più trasparente il processo decisionale e facilitando la comprensione delle minacce. Ad esempio, l'uso di strumenti come **SHAP (SHapley Additive Explanations)** e **Permutation Importance** consente di determinare quali feature del traffico di rete abbiano contribuito maggiormente alla classificazione di un evento come **normale o malevolo**.
Un'applicazione concreta della XAI in cybersecurity è la capacità di fornire spiegazioni su **quali parametri di rete** hanno portato a classificare un'operazione come sospetta. Se un modello segnala un'anomalia basata sul numero di connessioni simultanee o sulla durata delle sessioni,

gli analisti di sicurezza possono esaminare questi aspetti per verificare la presenza di un attacco reale, evitando di rispondere a falsi allarmi.
Un ulteriore vantaggio della XAI è la sua utilità per la conformità alle normative sulla protezione dei dati. Con l'introduzione di regolamenti come il **GDPR (General Data Protection Regulation)**, molte organizzazioni devono essere in grado di spiegare perché e come un sistema di IA ha preso determinate decisioni. L'uso di tecniche di XAI consente di garantire la trasparenza richiesta dalle normative, migliorando al tempo stesso la fiducia nei sistemi di cybersecurity basati su Machine Learning.
L'obiettivo di questa ricerca è quindi quello di **applicare modelli di Machine Learning alla cybersecurity** per migliorare il rilevamento delle intrusioni, garantendo al contempo trasparenza e interpretabilità tramite l'uso della XAI.

Figura 1.2: Alcuni Use Cases della XAI

Capitolo 2

Related Works

L'**Intelligenza Artificiale (IA)** sta diventando un elemento chiave nel campo della **cybersecurity**, rivoluzionando il modo in cui le minacce informatiche vengono rilevate e contrastate. Le attuali scoperte nell'ambito dell'**apprendimento automatico** permettono di sviluppare sistemi di **Intrusion Detection (IDS)** più avanzati, in grado di individuare attacchi malevoli e comportamenti anomali nel traffico di rete con maggiore precisione rispetto ai tradizionali metodi basati su regole e firme statiche.
Negli ultimi anni, la crescita esponenziale della connettività e l'aumento della complessità delle reti informatiche hanno reso le organizzazioni più vulnerabili agli attacchi informatici. Minacce come il **phishing, gli attacchi DDoS, le escalation di privilegi e le intrusioni non autorizzate** sono diventate sempre più sofisticate e difficili da rilevare con le metodologie tradizionali. I sistemi IDS basati su regole, infatti, spesso non sono in grado di riconoscere **attacchi zero-day**, ovvero minacce mai osservate prima, e generano un elevato numero di **falsi positivi**, causando inefficienze operative e rallentamenti nell'identificazione delle reali minacce.

2.1 L'IA per il rilevamento delle intrusioni

In questo contesto, l'**Intelligenza Artificiale** può essere implementata per **analizzare grandi volumi di traffico di rete in tempo reale** e individuare automaticamente anomalie che potrebbero segnalare un'intrusione. Gli algoritmi di **Machine Learning** sono in grado di **apprendere dai dati storici** e migliorare progressivamente la loro capacità di distinguere il traffico normale da quello malevolo. Tuttavia, l'introduzione di modelli di IA nella cybersecurity presenta una sfida cruciale: la **trasparenza e l'interpretabilià** delle decisioni prese dal sistema. Se un modello di machine learning classifica un'attività come un attacco informatico, è fondamentale capire **su quali fattori si è basato** per emettere quel verdetto.
Secondo lo studio [2], l'integrazione dell'**Explainable Artificial Intelligence (XAI)** nei sistemi di **Intrusion Detection** è essenziale per garantire che le previsioni dei modelli di IA non siano delle semplici *"black box"*, ma che possano essere comprese dagli analisti di sicurezza e dai professionisti IT.

2.2 Sfide e soluzioni dell'XAI nella cybersecurity

Uno dei principali ostacoli all'adozione dell'IA nella cybersecurity è la necessità di **bilanciare la precisione predittiva con la trasparenza del modello**. Modelli altamente complessi, come le reti neurali profonde o gli ensemble boosting, offrono un'alta accuratezza ma spesso risultano **opachi** nel loro processo decisionale. Inoltre, i modelli di IA necessitano di **grandi quantità di dati di qualità** per essere addestrati correttamente: dataset incompleti o sbilanciati possono

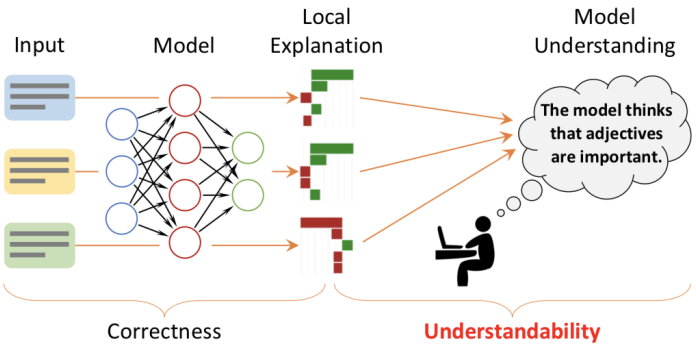

Figura 2.1: Ruolo dell'Explainable AI per la comprensione dei modelli

portare il modello a **fare previsioni errate**, con il rischio di non riconoscere attacchi reali o, al contrario, di segnalare erroneamente attività legittime come intrusioni.
L'uso di tecniche di **Explainable AI** può aiutare a mitigare questi problemi, permettendo di analizzare e interpretare il comportamento del modello. Strumenti come **SHAP (SHapley Additive Explanations)** e **LIME (Local Interpretable Model-agnostic Explanations)** consentono di individuare quali caratteristiche del traffico di rete abbiano avuto il maggiore impatto nella classificazione di un evento come un'intrusione. Questo non solo aiuta gli esperti di sicurezza a comprendere meglio le minacce informatiche, ma permette anche di **affinare le strategie di difesa** migliorando il training dei modelli.
Secondo lo studio [1], la crescente diffusione dell'IA nella sicurezza informatica permette di ridurre drasticamente i tempi di rilevamento degli attacchi e di adattarsi dinamicamente alle nuove minacce. Tuttavia, senza strumenti di interpretabilità, il rischio è che le decisioni prese dai modelli non siano affidabili e che i professionisti del settore non possano **validare le previsioni** prima di intraprendere azioni correttive.

2.3 Obiettivi della ricerca

L'analisi condotta in questa tesi mira a **valutare l'efficacia di modelli di Machine Learning per la rilevazione delle intrusioni**, evidenziando il ruolo dell'XAI nel migliorare la comprensione delle previsioni del modello. Attraverso l'uso del dataset **NSL-KDD**, verranno addestrati e confrontati due modelli di machine learning: **Random Forest e Gradient Boosting**. L'obiettivo è determinare **quale modello offra la migliore combinazione tra accuratezza e interpretabilità** e identificare le feature del traffico di rete che contribuiscono maggiormente alla classificazione delle minacce.
L'integrazione della **Explainable AI** nei sistemi IDS rappresenta un passo fondamentale per rendere le reti più sicure, affidabili e resilienti agli attacchi. La capacità di spiegare le decisioni dei modelli di IA non solo aumenta la fiducia nell'automazione della cybersecurity, ma consente anche ai professionisti di prendere decisioni più informate e reattive nella gestione delle minacce informatiche.

Capitolo 3

Design e Implementazione

3.1 Design

Seguendo gli obiettivi di questa tesi, in questa sezione verranno messi in chiaro i procedimenti seguiti per realizzare un software in grado di addestrare modelli di Machine Learning per la rilevazione delle intrusioni in rete, sperimentare diverse parametrizzazioni e valutare l'impatto delle feature nelle decisioni del sistema.
Il design del sistema è stato strutturato in modo modulare, suddividendo le diverse fasi del processo in componenti indipendenti ma interconnessi. Questo approccio ha permesso di mantenere flessibilità nell'implementazione e facilitare l'integrazione di nuovi modelli o tecniche di analisi. I principali moduli sviluppati includono:

- **Preprocessing dei dati**: encoding delle variabili categoriche, conversione delle feature, suddivisione del dataset.
- **Addestramento dei modelli**: applicazione di algoritmi di classificazione per individuare le intrusioni.
- **Valutazione delle performance**: utilizzo di metriche di accuratezza, precisione e recall per confrontare le prestazioni dei modelli.
- **Analisi delle feature**: interpretazione dell'importanza delle variabili utilizzate nel processo decisionale.

3.1.1 Classificazione

Gli algoritmi di classificazione sono tecniche utilizzate per assegnare un'etichetta o una classe a un determinato insieme di dati, in modo da poterli analizzare e sfruttare per prendere decisioni automatizzate.
Nel contesto di questa ricerca, si è fatto uso di algoritmi di classificazione supervisionata: essi richiedono un dataset di addestramento etichettato, ossia un insieme di dati per i quali è già nota la classe di appartenenza. L'algoritmo utilizza questi dati per apprendere un modello predittivo capace di classificare correttamente nuovi dati non ancora etichettati.
Nel caso specifico, il compito del classificatore è quello di distinguere tra traffico di rete normale e attività malevole, rilevando le intrusioni informatiche con maggiore accuratezza rispetto agli approcci tradizionali basati su firme statiche.
Un'importante caratteristica dei modelli utilizzati è la loro capacità di identificare pattern nascosti nei dati, individuando anomalie anche quando non sono presenti regole esplicite.

Ad esempio, mentre un sistema basato su firme potrebbe non riconoscere una nuova variante di attacco DDoS, un modello di Machine Learning può rilevare un comportamento anomalo basandosi sulle metriche di traffico.

3.1.2 Modelli

Nell'ambito dell'Intelligenza Artificiale, un modello è una rappresentazione matematica che consente di estrarre informazioni dai dati e di creare sistemi intelligenti in grado di fare previsioni, prendere decisioni e adattarsi a nuove situazioni.
In questa ricerca, sono stati utilizzati due modelli di Machine Learning ampiamente diffusi in ambito di Intrusion Detection:

- **Random Forest**: un metodo basato su alberi decisionali che utilizza il concetto di ensemble learning per migliorare la robustezza del modello.
- **Gradient Boosting**: un modello iterativo che ottimizza progressivamente la capacità predittiva tramite una serie di alberi deboli successivi.

L'adozione di due modelli differenti ha permesso di confrontare le loro prestazioni su diversi aspetti, tra cui accuratezza, interpretabilità e capacità di generalizzazione. Mentre il *Random Forest* si distingue per la sua robustezza e stabilità, il *Gradient Boosting* offre una maggiore precisione nelle predizioni, ma con un costo computazionale superiore.

Random Forest

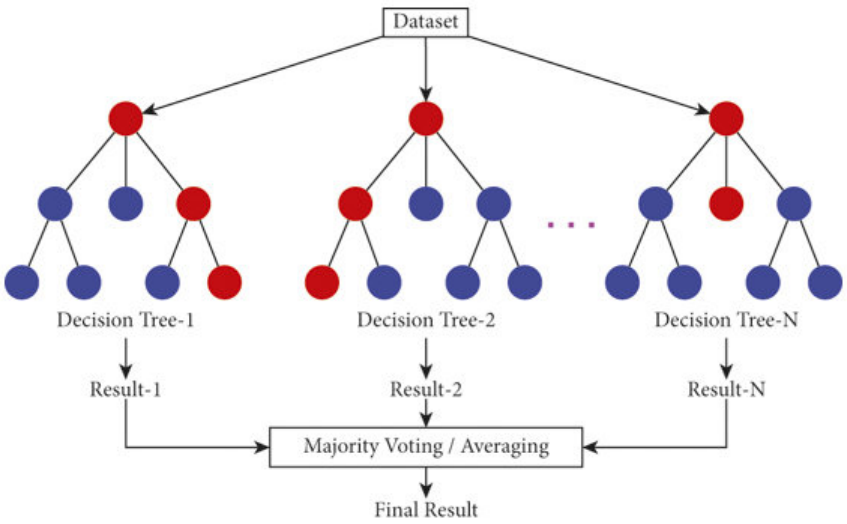

Figura 3.1: Schema riassuntivo del funzionamento di RandomForest

Il *Random Forest* è un algoritmo di apprendimento supervisionato che appartiene alla categoria degli algoritmi di *ensemble learning*. Questo approccio combina più modelli deboli (alberi decisionali) per ottenere una previsione più accurata e stabile.

Il funzionamento dell'algoritmo si basa su tre concetti fondamentali:

- **Bootstrap Aggregating (Bagging)**: ciascun albero è addestrato su un sottoinsieme casuale dei dati di addestramento.
- **Selezione casuale delle feature**: ogni nodo dell'albero viene diviso scegliendo solo un sottoinsieme di feature, rendendo gli alberi diversificati tra loro.
- **Aggregazione delle previsioni**: per la classificazione, la previsione finale viene ottenuta tramite una votazione a maggioranza tra gli alberi.

L'uso di molteplici alberi decisionali riduce il rischio di overfitting rispetto a un singolo Decision Tree, migliorando la capacità di generalizzazione del modello. Tuttavia, può risultare computazionalmente costoso quando si utilizza un elevato numero di alberi.
Un aspetto interessante del *Random Forest* è la sua capacità di gestire dataset con feature numeriche e categoriche senza richiedere trasformazioni particolari. Questo lo rende particolarmente utile nell'analisi del traffico di rete, dove le variabili possono avere natura eterogenea, come protocolli di comunicazione, dimensioni dei pacchetti e frequenza delle connessioni.

Gradient Boosting

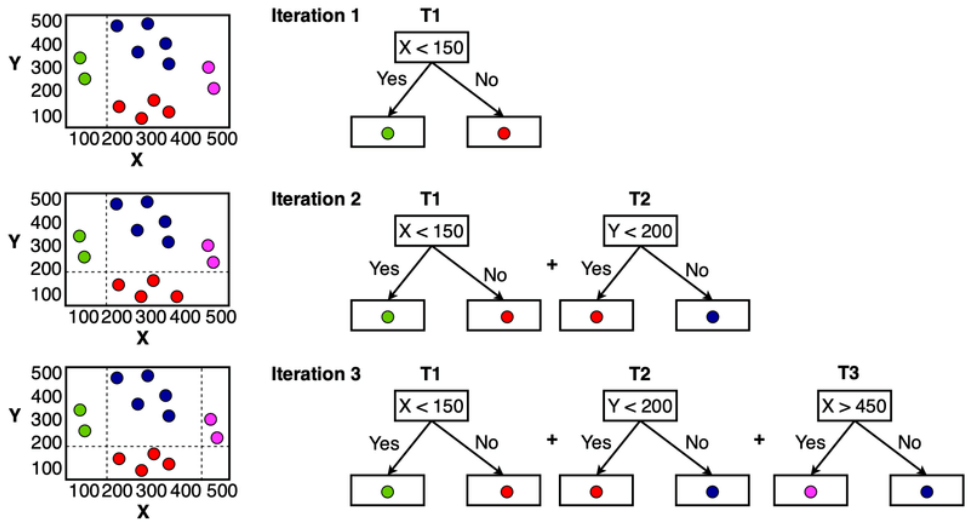

Figura 3.2: Schema riassuntivo del funzionamento di GradientBoosting

Il *Gradient Boosting* è un metodo di boosting che costruisce una sequenza di alberi decisionali in modo iterativo, ottimizzando progressivamente la capacità predittiva.
A differenza del Random Forest, in cui ogni albero lavora indipendentemente, nel Gradient Boosting ogni nuovo albero viene addestrato per correggere gli errori commessi dai modelli precedenti.
Il funzionamento può essere riassunto nei seguenti passi:

1. Viene costruito un primo albero decisionale per effettuare la previsione iniziale.
2. Si calcola l'errore residuo tra la previsione del modello e il valore reale.
3. Un nuovo albero viene addestrato per correggere gli errori commessi dal precedente.
4. Il processo viene ripetuto iterativamente fino alla convergenza del modello.

Il Gradient Boosting tende a fornire previsioni molto accurate, ma può essere più sensibile al rumore nei dati rispetto al Random Forest e richiede una corretta scelta degli iperparametri per evitare l'overfitting.
In questa ricerca, sono state esplorate diverse configurazioni di iperparametri per migliorare le prestazioni del modello, tra cui il numero di alberi (*n_estimators*), la profondità massima degli alberi (*max_depth*) e il tasso di apprendimento (*learning rate*). L'ottimizzazione di questi parametri ha consentito di bilanciare la precisione delle previsioni con il tempo di addestramento.

3.1.3 Parametrizzazioni

Gli iperparametri controllano il comportamento del modello e devono essere ottimizzati per migliorare le sue prestazioni. L'ottimizzazione degli iperparametri è spesso eseguita con tecniche come *Grid Search* o *Random Search*, in cui si testano diverse combinazioni per selezionare quelle che garantiscono la migliore accuratezza.
Per questa ricerca, sono stati scelti i seguenti iperparametri:

Random Forest:

- *n_estimators*: numero di alberi nella foresta.
- *max_depth*: profondità massima degli alberi per evitare overfitting.
- *min_samples_leaf*: numero minimo di campioni richiesti in una foglia.
- *max_features*: numero massimo di feature considerate per la divisione di un nodo.

Gradient Boosting:

- *n_estimators*: numero di alberi utilizzati nel boosting.
- *learning_rate*: tasso di apprendimento che regola la velocità di aggiornamento del modello.
- *max_depth*: profondità degli alberi utilizzati nel boosting.
- *subsample*: percentuale dei dati di addestramento usati in ogni iterazione per ridurre il rischio di overfitting.

3.1.4 Feature Importances

Un aspetto fondamentale dell'analisi è stato valutare quali feature del traffico di rete influenzino maggiormente le previsioni del modello. Sono stati utilizzati due metodi principali:

- **Feature Importance nativa del Random Forest**: questo metodo assegna un punteggio a ciascuna feature in base al suo impatto sulle decisioni del modello, misurando la riduzione dell'impurità (*Gini Importance*). Le feature con valori più alti sono quelle che maggiormente contribuiscono alla classificazione degli eventi come normali o malevoli.
- **SHAP (SHapley Additive Explanations)**: tecnica avanzata basata sulla teoria dei giochi che permette di comprendere il contributo di ogni feature alla predizione finale, fornendo spiegazioni dettagliate delle decisioni del modello. A differenza della *Gini Importance*, SHAP offre una spiegazione più dettagliata per ogni singola previsione.

La combinazione di questi due metodi ha permesso di evidenziare quali caratteristiche del traffico di rete sono più indicative di una possibile intrusione. Ad esempio, feature legate alla dimensione dei pacchetti scambiati, alla durata delle connessioni e alla frequenza degli accessi a determinati servizi sono risultate particolarmente rilevanti per il rilevamento delle anomalie. Questo tipo di analisi è essenziale per migliorare l'affidabilità dei modelli e garantire una maggiore trasparenza nel processo decisionale.

3.2 Implementazione

L'intero software è stato implementato in Python, sfruttando librerie dedicate all'analisi dei dati e al Machine Learning, tra cui:

- **Scikit-learn**: implementazione degli algoritmi di classificazione, utilizzato per la creazione, l'addestramento e la validazione dei modelli *Random Forest* e *Gradient Boosting*.
- **Pandas**: gestione e manipolazione dei dataset, consentendo operazioni di pulizia, filtraggio e trasformazione delle feature prima dell'addestramento.
- **Seaborn e Matplotlib**: visualizzazione dei risultati tramite grafici e matrici di confusione, utili per l'analisi delle prestazioni e l'interpretazione delle feature importances.
- **SHAP**: analisi delle feature importances, fornendo un'interpretazione dettagliata delle decisioni del modello e migliorando la comprensione delle dinamiche sottostanti alle predizioni.

Il codice è stato organizzato in una pipeline strutturata che include le seguenti fasi principali:

1. **Preprocessing**: encoding delle variabili categoriche e suddivisione del dataset.
2. **Addestramento**: applicazione di tecniche di ottimizzazione degli iperparametri per massimizzare l'accuratezza e ridurre il rischio di overfitting.

3. **Valutazione**: confronto tra i due modelli (*Random Forest* e *Gradient Boosting*) sulla base delle metriche di classificazione e analisi delle matrici di confusione.

4. **Interpretazione**: utilizzo delle tecniche di XAI per comprendere il ruolo delle feature e fornire maggiore trasparenza agli analisti di sicurezza informatica.

3.3 Use Case

Gli attori coinvolti nel processo sono:

- **Data Scientist**: si occupa dell'addestramento, ottimizzazione e interpretazione del modello.
- **Analista di sicurezza**: utilizza il sistema per monitorare la rete e identificare minacce informatiche.

I principali casi d'uso relativi al lavoro di tesi sono i seguenti:

3.3.1 Addestramento del Modello

Attore principale: Data Scientist
Descrizione: Il Data Scientist seleziona il dataset e applica le tecniche di preprocessing necessarie per la preparazione dei dati. Una volta pulito il dataset, procede con l'addestramento del modello *Random Forest* e *Gradient Boosting*, valutandone le prestazioni attraverso metriche di accuratezza e classificazione.
Post-condizione: Il modello è stato addestrato e pronto per essere testato sul traffico di rete.

3.3.2 Ottimizzazione del Modello

Attore principale: Data Scientist
Descrizione: Il Data Scientist esegue un'ottimizzazione degli iperparametri per migliorare le prestazioni del modello. Utilizza tecniche come *Grid Search* e *Random Search* per selezionare i parametri migliori, riducendo il rischio di overfitting e massimizzando la capacità predittiva del sistema.
Post-condizione: Il modello è ottimizzato e pronto per il deployment.

3.3.3 Monitoraggio del Traffico di Rete

Attore principale: Analista di sicurezza
Descrizione: L'analista di sicurezza utilizza il modello per analizzare il traffico di rete in tempo reale. Il sistema classifica le connessioni come normali o malevole e segnala eventuali anomalie sospette.
Post-condizione: Le attività sospette sono state rilevate e l'analista può intraprendere azioni correttive.

3.3.4 Interpretazione delle Decisioni del Modello

Attore principale: Data Scientist / Analista di sicurezza
Descrizione: Per comprendere meglio il funzionamento del modello e garantire maggiore trasparenza, il Data Scientist applica tecniche di Explainable AI (*SHAP, Feature Importance*). L'analista di sicurezza utilizza queste informazioni per prendere decisioni basate sulle feature più influenti.
Post-condizione: Le decisioni del modello sono state spiegate, migliorando la fiducia e l'interpretabilità dei risultati.

3.3.5 Aggiornamento del Modello

Attore principale: Data Scientist
Descrizione: Poiché le minacce informatiche evolvono nel tempo, il Data Scientist aggiorna periodicamente il modello con nuovi dati per migliorare la capacità di rilevamento di attacchi emergenti.
Post-condizione: Il modello è stato aggiornato con nuovi dati ed è pronto per rilevare nuove tipologie di minacce.

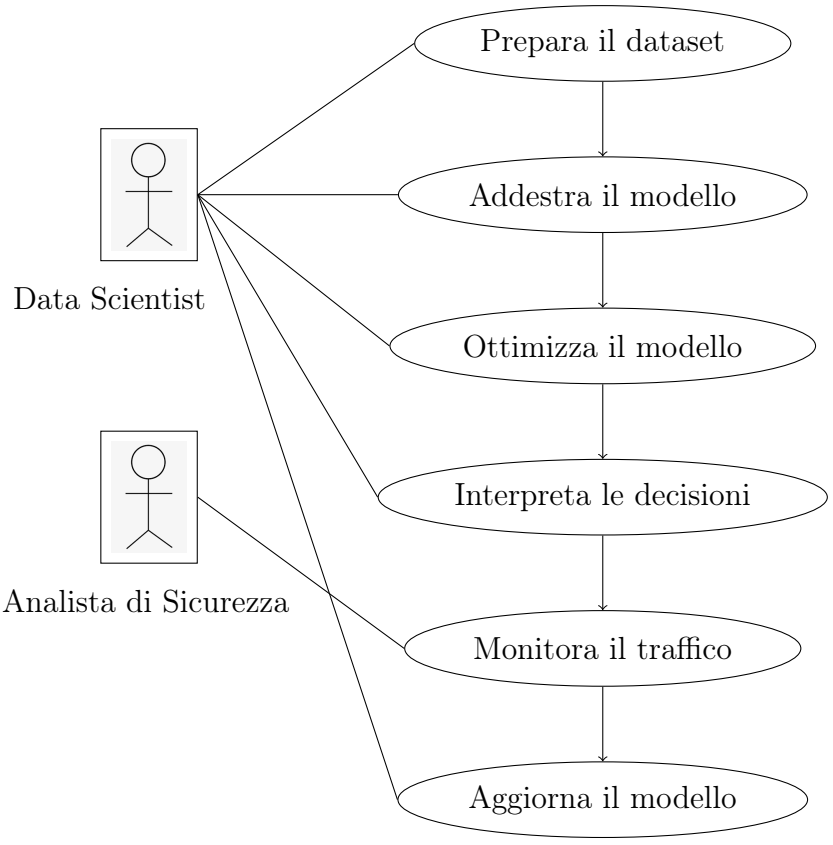

Figura 3.3: Diagramma UML dei Casi d'Uso

3.3.6 Casi d'uso per il Data Scientist

Il *Data Scientist* gioca un ruolo fondamentale nella progettazione, addestramento e ottimizzazione dei modelli di Machine Learning per la rilevazione delle intrusioni nei sistemi di rete.
In questa sezione verranno descritti i principali casi d'uso relativi alle sue attività.

- **Addestramento del modello**: il Data Scientist si occupa della selezione del dataset e della preparazione dei dati, assicurandosi che il modello possa apprendere in modo efficace dai pattern di traffico di rete. L'uso di tecniche di bilanciamento delle classi consente di evitare distorsioni nei risultati.

- **Ottimizzazione del modello**: per migliorare l'efficacia del sistema, il Data Scientist sperimenta diverse configurazioni di iperparametri sui modelli Random Forest e Gradient Boosting. Attraverso tecniche di ottimizzazione come la *Grid Search* e la *Random Search*, seleziona le combinazioni di parametri che massimizzano le prestazioni del modello in termini di accuratezza, precisione e recall. Inoltre, può valutare l'impatto dell'aumento della complessità del modello sull'overfitting.

- **Estrazione di informazioni dal modello**: un aspetto critico del lavoro del Data Scientist è l'interpretazione delle decisioni del modello. Utilizzando metodi di *Explainable AI* (XAI), come le *Feature Importances* e *SHAP*, egli è in grado di comprendere il peso delle diverse feature del traffico di rete sulle predizioni effettuate dal sistema. Questo processo aiuta a identificare eventuali bias nel modello e fornisce una maggiore trasparenza agli analisti di sicurezza informatica.

- **Monitoraggio e aggiornamento del modello**: il comportamento delle minacce informatiche evolve nel tempo, rendendo necessario un monitoraggio continuo delle prestazioni del modello. Il Data Scientist analizza periodicamente le predizioni effettuate, confrontandole con nuovi dati raccolti in tempo reale, e aggiorna il modello per garantire una capacità di rilevamento efficace anche contro attacchi emergenti e sconosciuti.

- **Valutazione della robustezza del modello**: un ulteriore compito del Data Scientist è testare la robustezza del modello simulando attacchi avversari (*Adversarial Machine Learning*) per individuare eventuali vulnerabilità. Questa pratica permette di migliorare la resilienza del sistema contro tecniche di evasione avanzate utilizzate dai cybercriminali.

Capitolo 4

Case Study

Il caso di studio di questa tesi riguarda l'analisi di dati di traffico di rete con l'obiettivo di rilevare intrusioni informatiche e attacchi alla sicurezza dei sistemi. Per questo scopo, è stato utilizzato il dataset **NSL-KDD**, una versione migliorata del dataset KDD'99, comunemente impiegato nella ricerca sugli *Intrusion Detection System* (IDS). L'obiettivo è quello di sviluppare modelli di *Machine Learning* in grado di distinguere tra traffico di rete normale e attività malevole, garantendo un miglioramento rispetto agli approcci tradizionali basati su firme statiche.
L'utilizzo di modelli di apprendimento automatico per la cybersecurity si è dimostrato particolarmente efficace nell'individuare pattern di attacco nascosti nei dati di traffico di rete. A differenza delle tecniche tradizionali basate su regole statiche, i modelli di Machine Learning sono in grado di adattarsi a nuove minacce, migliorando la capacità di rilevamento di attacchi emergenti. Inoltre, il ricorso a tecniche di *Explainable AI* ha reso più trasparente il processo decisionale, permettendo agli analisti di sicurezza di interpretare meglio i risultati e di adottare misure correttive più mirate.

4.1 Il Dataset

Il dataset **NSL-KDD** contiene una serie di connessioni di rete etichettate, classificate come traffico normale o attacco. Ogni connessione è descritta attraverso 41 feature che rappresentano aspetti delle comunicazioni di rete, come durata della connessione, numero di pacchetti inviati e ricevuti, protocolli di trasmissione utilizzati e statistiche sulle interazioni tra host.
Le etichette delle connessioni indicano se il traffico è *normale* o appartiene a una delle seguenti classi di attacco:

- **Denial of Service (DoS)**: attacchi volti a sovraccaricare le risorse di un sistema rendendolo inaccessibile.
- **Probing (Probe)**: tentativi di raccolta di informazioni sulla rete per identificare vulnerabilità sfruttabili.
- **User to Root (U2R)**: attacchi in cui un utente malintenzionato tenta di ottenere privilegi amministrativi.
- **Remote to Local (R2L)**: attacchi in cui un utente remoto tenta di ottenere accesso non autorizzato a una macchina della rete.

Il dataset NSL-KDD è stato scelto rispetto al suo predecessore KDD'99 poiché elimina problemi di ridondanza nei dati, riducendo il rischio di bias nel training dei modelli. Nel dataset KDD'99, infatti, era presente una quantità eccessiva di istanze duplicate, che potevano portare i modelli di Machine Learning a sovradattarsi ai dati di addestramento e a perdere capacità di

generalizzazione. NSL-KDD rappresenta quindi una soluzione più bilanciata per testare algoritmi di rilevamento delle intrusioni.

4.2 Preprocessing

Affinché i modelli di *Machine Learning* possano operare efficacemente, è necessario che i dati siano opportunamente pre-elaborati. Il preprocessing del dataset è stato articolato nelle seguenti fasi:

4.2.1 Separazione delle Feature e delle Etichette

Il primo passo è stato separare le feature indipendenti (X) dalla variabile target (y). Quest'ultima rappresenta la classe di ciascun campione, indicata come *normal* o *anomaly*. Poiché alcune etichette contenevano spazi vuoti, è stata effettuata un'operazione di pulizia tramite il metodo `str.strip()`, garantendo una corretta identificazione delle classi.

4.2.2 Codifica delle Feature Categoriche

Il dataset NSL-KDD include alcune feature categoriche, come il tipo di protocollo (*TCP, UDP, ICMP*) e il servizio di rete (*HTTP, FTP, SSH, ecc.*). Poiché i modelli di *Machine Learning* operano su dati numerici, queste feature sono state convertite in valori numerici utilizzando la classe `LabelEncoder` della libreria *scikit-learn*. Questa tecnica ha permesso di assegnare un codice numerico univoco a ogni categoria, consentendo al modello di elaborare correttamente le informazioni.

4.2.3 Suddivisione del Dataset

Per garantire una corretta valutazione delle prestazioni del modello, il dataset è stato suddiviso in due insiemi:

- **Training set**: contiene il 70% dei dati e viene utilizzato per l'addestramento del modello.
- **Test set**: comprende il restante 30% e viene impiegato per la valutazione delle prestazioni del modello su dati mai visti prima.

La suddivisione è stata effettuata tramite la funzione `train_test_split` di *scikit-learn*, utilizzando l'opzione `stratify=y` per mantenere le proporzioni originali tra classi e prevenire uno sbilanciamento dei dati.

4.2.4 Dimensione del Dataset Prima e Dopo il Preprocessing

Le operazioni di preprocessing non hanno comportato la rimozione di campioni, ma hanno trasformato alcune feature categoriche in numeriche. La dimensione del dataset è rimasta invariata in termini di numero di campioni, mentre le feature sono state convertite in un formato numerico adatto ai modelli di *Machine Learning*.

- **Dimensione iniziale del dataset:** 125968 campioni, 42 feature (di cui alcune categoriche).
- **Dopo il preprocessing:** 125968 campioni, 41 feature (tutte numeriche).

Capitolo 5

Risultati Sperimentali

In questo capitolo vengono analizzati i risultati ottenuti dai modelli *Random Forest* e *Gradient Boosting* applicati al dataset **NSL-KDD**. L'obiettivo principale è valutare le prestazioni dei modelli in termini di accuratezza e capacità di rilevare intrusioni nella rete. Inoltre, verranno esaminate le feature più influenti nelle decisioni dei modelli.

5.1 Metriche e grafici

Per valutare le prestazioni dei modelli, sono state utilizzate le seguenti metriche:

- **Accuracy Score**: misura la percentuale di predizioni corrette rispetto al totale delle predizioni effettuate. È calcolata come segue:

$$Accuracy = \frac{\text{numero di predizioni corrette}}{\text{numero totale di predizioni}} \tag{5.1}$$

  Un valore elevato di accuracy indica buone prestazioni globali del modello, ma è importante considerare anche metriche più specifiche per comprendere come il modello si comporta sulle singole classi.

- **Classification Report**: include precisione, recall e F1-score per ogni classe, fornendo un'analisi dettagliata delle performance del modello. Questo permette di valutare non solo la capacità del modello di classificare correttamente le istanze, ma anche di comprendere se alcune classi vengano penalizzate, ad esempio nel caso di dataset sbilanciati.

- **Matrice di confusione**: rappresenta il numero di predizioni corrette e errate suddivise per classe, permettendo di identificare eventuali errori sistematici. Analizzando la matrice di confusione possiamo osservare se il modello tende a confondere particolari classi, ad esempio se classifica erroneamente attacchi di tipo *Probe* come normali, il che potrebbe avere implicazioni significative per la sicurezza della rete.

- **Feature Importance**: misura l'impatto di ciascuna feature nelle previsioni del modello, evidenziando quali caratteristiche sono più influenti per la classificazione. Ad esempio, potrebbe emergere che parametri come il numero di pacchetti inviati o la durata della connessione siano determinanti nella distinzione tra traffico normale e attacchi.

- **Distribuzione delle classi originali:** La distribuzione mostra una lieve differenza nel numero di campioni tra le due classi. Tuttavia, è fondamentale verificare se il dataset presenta uno sbilanciamento significativo, poiché un'eccessiva differenza tra il numero di istanze normali e quelle di attacco potrebbe influenzare negativamente l'efficacia del modello, portandolo a favorire la classe più rappresentata.

Per visualizzare le prestazioni dei modelli, sono stati generati diversi grafici, tra cui la curva ROC (*Receiver Operating Characteristic*), che mostra la relazione tra il tasso di veri positivi e il tasso di falsi positivi per vari livelli di soglia decisionale. Un'area sotto la curva (*AUC*) più alta indica una migliore capacità del modello di distinguere tra traffico normale e malevolo.
Inoltre, è stata analizzata la distribuzione delle predizioni errate al fine di comprendere quali tipi di attacco risultano più difficili da identificare. Questo aspetto è cruciale poiché alcune categorie di attacco, come *User to Root* (U2R), presentano un numero limitato di esempi nel dataset, rendendo la loro classificazione più complessa. Strategie di riequilibrio, come l'oversampling o il cost-sensitive learning, potrebbero essere adottate in futuro per migliorare la rilevazione di queste minacce.
Infine, sono stati confrontati i tempi di esecuzione dei due modelli. Il *Random Forest* ha mostrato tempi di inferenza più rapidi rispetto al *Gradient Boosting*, rendendolo una scelta potenzialmente più adatta per applicazioni in tempo reale, dove la velocità di risposta è fondamentale per contrastare intrusioni in tempo utile.

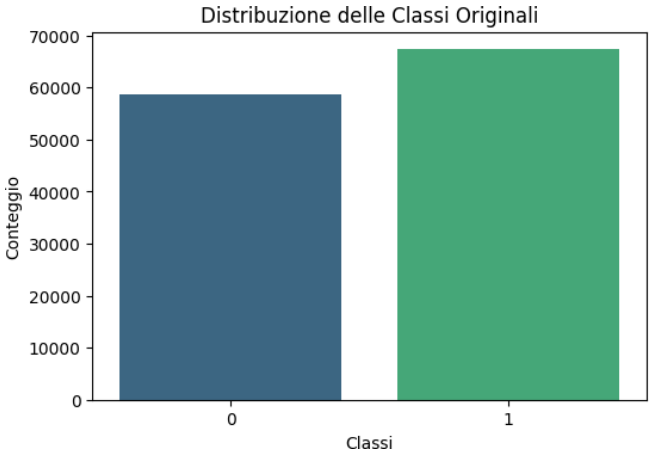

Figura 5.1: Distribuzione delle classi originali

5.2 Prestazioni dei modelli

5.2.1 Random Forest

Il modello *Random Forest* ha ottenuto un'accuratezza complessiva del **99.79%**. Di seguito, i risultati principali:

- **Matrice di confusione**:
  - Classe "anomaly": 17.518 campioni correttamente classificati, 70 predetti erroneamente come "normal".
  - Classe "normal": 20.192 campioni correttamente classificati, 11 predetti erroneamente come "anomaly".

- **Classification Report**: precisione, recall e F1-score superiori al 99% per entrambe le classi.
- **Distribuzione delle classi originali**: il dataset presenta una lieve differenza nel numero di campioni tra le due classi, ma il modello riesce comunque a mantenere un'ottima accuratezza.

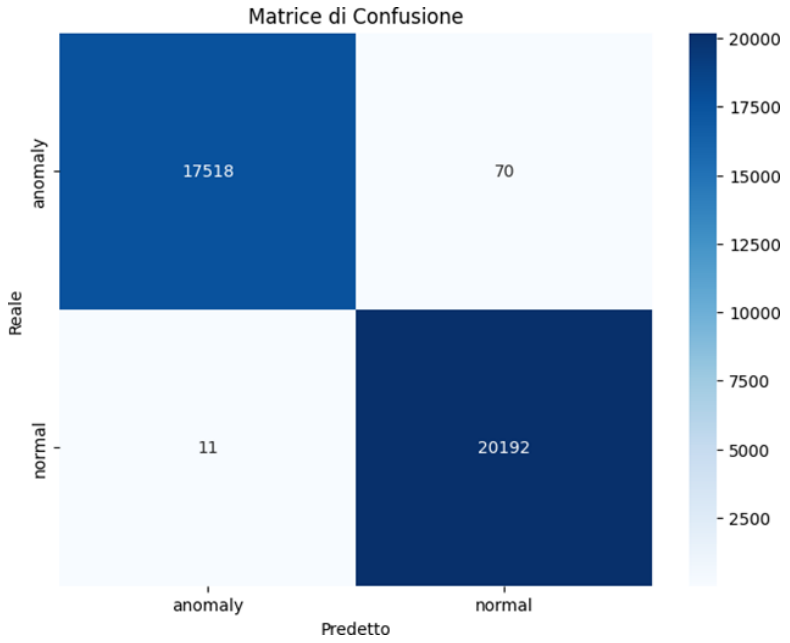

Figura 5.2: Matrice di Confusione relativa a RandomForest

5.2.2 Gradient Boosting

Il modello *Gradient Boosting* ha ottenuto un'accuratezza leggermente superiore, pari a **99.87%**. I risultati principali sono:

- **Matrice di confusione**:
  - Classe "anomaly": 17.553 campioni correttamente classificati, 35 predetti erroneamente come "normal".
  - Classe "normal": 20.189 campioni correttamente classificati, 14 predetti erroneamente come "anomaly".
- **Classification Report**: precisione, recall e F1-score superiori al 99% per entrambe le classi.

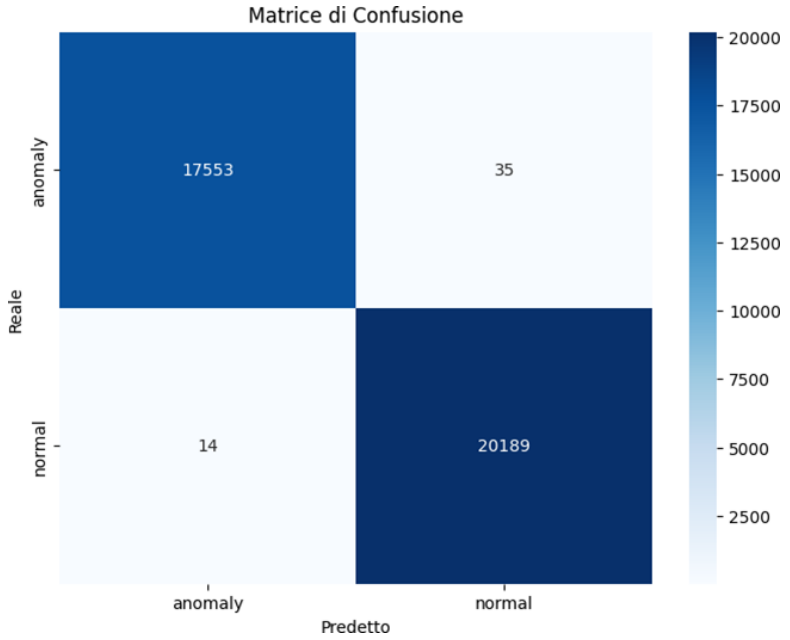

Figura 5.3: Matrice di Confusione relativa a GradientBoosting

5.3 Feature Importance

Per comprendere meglio le decisioni prese dai modelli, è stata analizzata l'importanza delle feature. Di seguito, il confronto tra le 15 feature più significative per *Random Forest* e *Gradient Boosting*.

5.3.1 Random Forest - Top 15 Feature Importance

Il modello *Random Forest* ha identificato le seguenti 15 feature come le più importanti per il processo decisionale.

- La feature più determinante è **Column5**, con un valore di importanza significativamente superiore rispetto alle altre. Questo conferma l'importanza del volume di dati trasferiti nel rilevamento delle intrusioni.
- **Column6** e **Column12** sono anch'esse molto rilevanti e contribuiscono significativamente alle previsioni del modello, suggerendo che gli attacchi interni e i segnali di compromissione sono aspetti critici della sicurezza.
- La feature **Column4_SF** emerge come una delle più influenti. Questo indica che la classificazione dei protocolli di comunicazione gioca un ruolo importante nella distinzione tra traffico normale e malevolo.
- Altre feature chiave includono **Column33, Column34, Column39 e Column23**, che hanno un impatto minore ma comunque rilevante. La loro presenza nella top 15 evidenzia la complessità del fenomeno da analizzare.

- Le feature **Column3_private** e **Column3_http** risultano meno influenti rispetto alle prime posizioni, ma rimangono tra i primi 15 fattori decisionali. Questo suggerisce che, pur non essendo determinanti in senso assoluto, le informazioni relative al tipo di servizio utilizzato possono essere utili nella rilevazione di schemi anomali.

Osservazione: Il modello *Random Forest* utilizza un'ampia gamma di feature per effettuare le previsioni, senza una forte dipendenza da un numero ristretto di variabili. Questo lo rende robusto a eventuali variazioni nel dataset, ma al tempo stesso richiede un'interpretazione attenta dell'importanza relativa delle diverse feature.

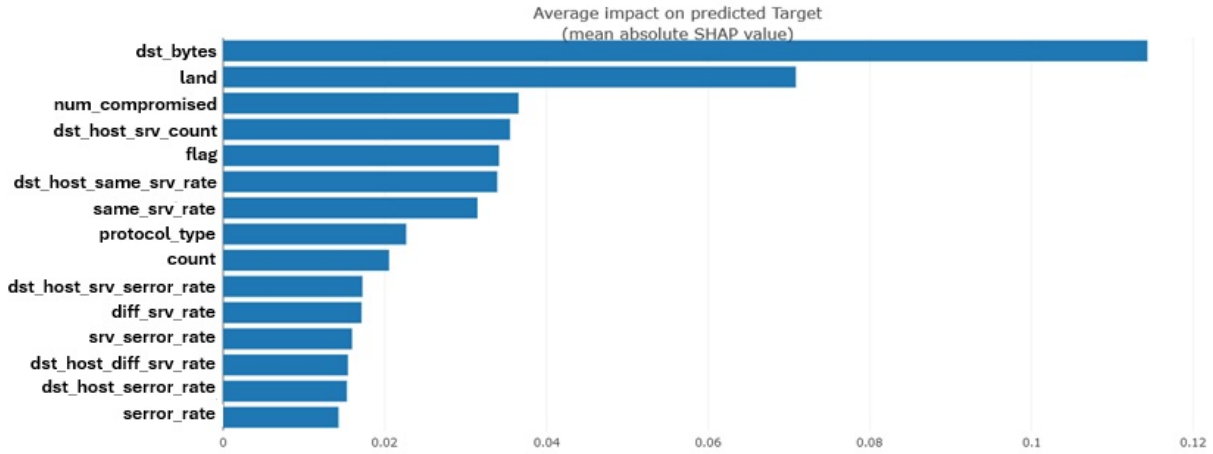

Figura 5.4: Random Forest - Top 15 Feature Importance

5.3.2 Gradient Boosting - Top 15 Feature Importance

Analogamente a *Random Forest*, il modello *Gradient Boosting* ha individuato le seguenti feature come le più influenti:

- Anche per *Gradient Boosting*, la feature **Column5** è la più importante, con un peso ancora più accentuato rispetto a *Random Forest*. Questo indica che il modello attribuisce un'attenzione particolare ai volumi di dati scambiati, potenzialmente migliorando la sensibilità al rilevamento di attacchi basati sul traffico anomalo.
- **Column6** mantiene un'elevata rilevanza, ma ha un impatto inferiore rispetto alla prima feature, suggerendo che l'identificazione degli attacchi interni rimane un elemento importante ma meno dominante.
- Le feature **Column33** e **Column3_private** sono molto più influenti rispetto a quanto osservato con *Random Forest*, il che può suggerire che *Gradient Boosting* ponga maggiore enfasi sulla natura del servizio di rete utilizzato.

- **Column23, Column34, Column38 e Column30** continuano ad avere un peso rilevante, sebbene con proporzioni leggermente diverse rispetto a *Random Forest*.
- **Column2_icmp** e **Column3_http**, pur avendo un'importanza inferiore rispetto alle feature numeriche principali, rientrano comunque nella top 15.

Osservazione: Il modello *Gradient Boosting* enfatizza un numero più ristretto di feature chiave rispetto a *Random Forest*, suggerendo che il modello attribuisce maggiore peso a pattern specifici nei dati.

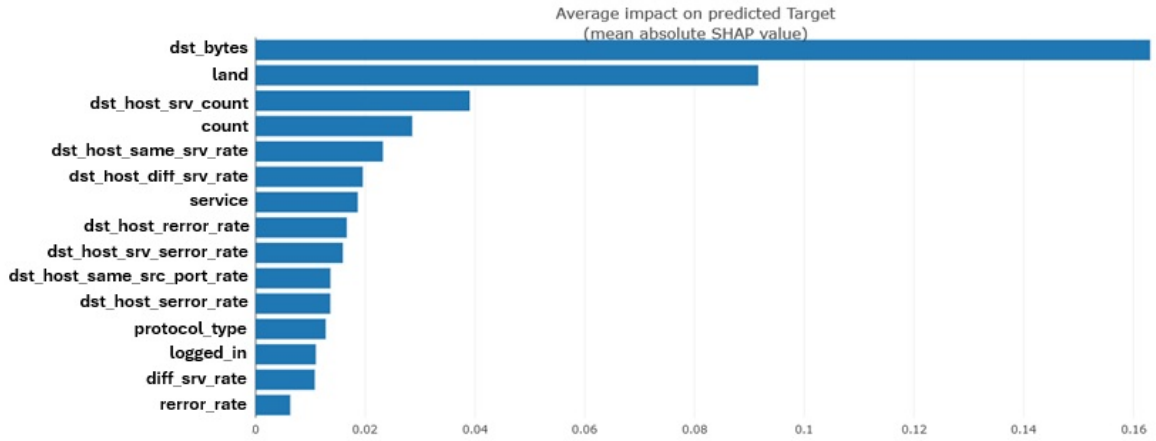

Figura 5.5: Gradient Boosting - Top 15 Feature Importance

Capitolo 6

Conclusioni

L'obiettivo di questa ricerca era sviluppare un sistema di rilevamento delle intrusioni basato su modelli di *Machine Learning*, capace di analizzare il traffico di rete e individuare attività anomale con elevata accuratezza. L'integrazione di tecniche di *Explainable Artificial Intelligence* (XAI) ha consentito non solo di migliorare le capacità di individuazione delle minacce, ma anche di comprendere il processo decisionale dei modelli impiegati, fornendo una visione più chiara sui fattori chiave che determinano la classificazione di un traffico come normale o malevolo.

6.1 Analisi delle prestazioni

I risultati sperimentali hanno dimostrato che entrambi i modelli implementati, *Random Forest* e *Gradient Boosting*, hanno raggiunto livelli di accuratezza estremamente elevati.

- **Random Forest**: ha ottenuto un'accuratezza del **99.79%**, confermandosi come un modello solido e ben bilanciato. La sua capacità di combinare molteplici alberi decisionali ha garantito una riduzione dell'overfitting e un'ottima generalizzazione.
- **Gradient Boosting**: ha raggiunto un'accuratezza leggermente superiore, pari a **99.87%**. Grazie alla sua natura iterativa, ha mostrato una migliore capacità di apprendimento dai dati e una maggiore sensibilità ai pattern complessi presenti nel traffico di rete.

Metrica	**Random Forest**	**Gradient Boosting**
Accuracy	99.79%	99.87%
Precisione	99%	99%
Recall	99%	99%
F1-score	99%	99%

Tabella 6.1: Metriche di Performance per i modelli di Machine Learning

Entrambi i modelli hanno dimostrato un'elevata efficacia nel rilevare attacchi informatici, con differenze minime nelle prestazioni. Tuttavia, *Gradient Boosting* ha evidenziato una maggiore selettività nell'uso delle feature, mentre *Random Forest* ha distribuito l'importanza su un numero più ampio di variabili, garantendo una robustezza maggiore in scenari più variabili.

6.2 Interpretazione dei risultati

L'analisi delle *feature importances* ha permesso di identificare le variabili più rilevanti per la classificazione del traffico di rete, fornendo indicazioni utili per la sicurezza informatica.

- **Column5 (dst_bytes)**: Questa feature indica il numero di byte inviati dalla destinazione alla sorgente. È risultata la variabile più influente in entrambi i modelli, suggerendo che il volume di dati ricevuti da una destinazione possa essere un indicatore chiave di attività sospette. Molti attacchi, come il *Denial of Service* (DoS) o i tentativi di esfiltrazione di dati, generano volumi di traffico anomali. Ad esempio, un attacco DoS può essere caratterizzato da un numero di pacchetti molto elevato inviati verso una destinazione senza una corrispettiva risposta adeguata.

- **Column6 (land)**: Questa feature assume valore 1 se l'attacco proviene dalla stessa macchina su cui è destinato il traffico, 0 altrimenti.
  Gli attacchi interni alla rete, come il *spoofing* o il *lateral movement* di un attaccante che ha già compromesso una macchina, possono essere caratterizzati da connessioni interne sospette. L'alta rilevanza di questa feature indica che i modelli hanno appreso a riconoscere tali schemi.

- **Column12 (num_compromised)**: Questa variabile indica il numero di indicatori di compromissione rilevati all'interno di una connessione. Un valore elevato di questa feature è spesso associato a un'intrusione.
  Tipicamente, attacchi come il *buffer overflow* o le *escalation di privilegi*, causano un aumento del numero di file alterati o di tentativi di accesso non autorizzati. I modelli hanno identificato questa feature come fortemente discriminante tra traffico normale e malevolo.

- **Column3_private (service = private)**: Questa feature indica se il servizio utilizzato è un servizio privato (non standard).
  Molte tipologie di attacchi sfruttano porte e protocolli non standard per evitare di essere rilevati dai firewall o dai sistemi IDS tradizionali. Connessioni su servizi privati possono essere segni di attività malevole, come backdoor o accessi remoti non autorizzati.

- **Column3_http (service = http)**: Questa feature indica se la connessione avviene su protocollo HTTP.
  Il traffico HTTP è spesso bersaglio di attacchi come *SQL Injection*, *Cross-Site Scripting* (XSS) e attacchi *brute force* su pagine di login. Un'analisi delle richieste HTTP può fornire importanti indizi su attività anomale, rendendo questa variabile cruciale per il rilevamento delle intrusioni.

L'impiego delle tecniche di XAI, in particolare i valori di SHAP, ha permesso di comprendere come il modello giunga alle proprie decisioni. Questa trasparenza è essenziale per gli analisti di sicurezza, poiché consente di interpretare i risultati con maggiore affidabilità e di migliorare le strategie di difesa contro le minacce informatiche.

Capitolo 7

Appendice A

Librerie

Di seguito sono riportate le librerie utilizzate per l'implementazione del sistema di *Intrusion Detection* basato su Machine Learning:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestClassifier,
    GradientBoostingClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report, confusion_matrix,
    accuracy_score
```

Listing 7.1: Librerie utilizzate

RandomForest

Il modello *Random Forest* è stato implementato con i seguenti parametri:

```
# Inizializzazione del modello
rf_model = RandomForestClassifier(n_estimators=100, max_depth=10,
    class_weight='balanced', random_state=42)

# Addestramento del modello
rf_model.fit(X_train_df, y_train)

# Predizioni e valutazione
y_pred_rf = rf_model.predict(X_test)
accuracy_rf = accuracy_score(y_test, y_pred_rf)

print("\nRandom Forest - Accuracy:", accuracy_rf)
print("\nClassification Report:")
print(classification_report(y_test, y_pred_rf))

# Matrice di confusione
conf_matrix_rf = confusion_matrix(y_test, y_pred_rf)
plt.figure(figsize=(8, 6))
```

```
sns.heatmap(conf_matrix_rf, annot=True, fmt='d', cmap='Blues',
    xticklabels=label_encoder_y.classes_, yticklabels=label_encoder_y.
    classes_)
plt.title('Matrice di Confusione - Random Forest')
plt.xlabel('Predetto')
plt.ylabel('Reale')
plt.show()

# Feature Importance per Random Forest (TOP 15)
if hasattr(rf_model, 'feature_importances_'):
    rf_importances = rf_model.feature_importances_
    indices = np.argsort(rf_importances)[::-1][:15]

    plt.figure(figsize=(10, 6))
    plt.title('Importanza delle Feature - Random Forest (Top 15)')
    plt.bar(range(15), rf_importances[indices], align='center')
    plt.xticks(range(15), X.columns[indices], rotation=90)
    plt.xlabel('Feature')
    plt.ylabel('Importanza')
    plt.tight_layout()
    plt.show()
```

Listing 7.2: Implementazione del modello RandomForest

GradientBoosting

Il modello *Gradient Boosting* è stato implementato con la seguente configurazione:

```
# Inizializzazione del modello
gb_model = GradientBoostingClassifier(n_estimators=100, learning_rate
    =0.1, max_depth=5, random_state=42)

# Addestramento del modello
gb_model.fit(X_train_df, y_train)

# Predizioni e valutazione
y_pred_gb = gb_model.predict(X_test)
accuracy_gb = accuracy_score(y_test, y_pred_gb)

print("\nGradient Boosting - Accuracy:", accuracy_gb)
print("\nClassification Report:")
print(classification_report(y_test, y_pred_gb))

# Matrice di confusione
conf_matrix_gb = confusion_matrix(y_test, y_pred_gb)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_gb, annot=True, fmt='d', cmap='Blues',
    xticklabels=label_encoder_y.classes_, yticklabels=label_encoder_y.
    classes_)
plt.title('Matrice di Confusione - Gradient Boosting')
plt.xlabel('Predetto')
```

```
plt.ylabel('Reale')
plt.show()

# Feature Importance per Gradient Boosting (TOP 15)
if hasattr(gb_model, 'feature_importances_'):
    gb_importances = gb_model.feature_importances_
    indices = np.argsort(gb_importances)[::-1][:15]

    plt.figure(figsize=(10, 6))
    plt.title('Importanza delle Feature - Gradient Boosting (Top 15)')
    plt.bar(range(15), gb_importances[indices], align='center')
    plt.xticks(range(15), X.columns[indices], rotation=90)
    plt.xlabel('Feature')
    plt.ylabel('Importanza')
    plt.tight_layout()
    plt.show()
```

Listing 7.3: Implementazione del modello Gradient Boosting

Bibliografia

[1] A. L. Buczak and E. Guven. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys Tutorials*, 18(2), 2016.

[2] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao. Explainable ai for cybersecurity: Current trends and future directions. *IEEE Communications Surveys Tutorials*, 2021.