TrafficML

–

Rilevamento di attacchi sulla rete con Machine Learning

Sistemi Intelligenti per la Comunicazione Digitale

a.a. 2024-2025

Corso di laurea in

Informatica e Comunicazione Digitale – Sede di Taranto

A cura di: Mongelli Antonio

Docenti: Casalino Gabriella

Zaza Gianluca

Sommario

Documentazione Progetto: **TrafficML**

Contesto del Progetto

Il progetto si concentra sull'addestramento di un modello di machine learning supervisionato per automatizzare il processo di rilevamento e di risposta al traffico maligno in rete.

TrafficML è un sistema di analisi del traffico di rete basato su tecniche di **machine learning supervisionato** per la rilevazione automatica di attacchi informatici. Utilizzando dataset reali di traffico (come **CIC-IDS2017, BCCC-CIC-IDS2017, CIC-DDoS2019, MACCDC2012**), il progetto addestra un modello di classificazione addestrato sul dataset CIC-IDS2017 in grado di distinguere tra traffico **benigno** e **maligno** ricevuto da Suricata, un IDS (Intrusion Detection System), che monitora T-pot (Honeypot), categorizzando diversi tipi di attacchi noti come DDoS, PortScan, Brute Force, Web Attack e altri.

Obiettivo

L'interfaccia grafica proposta mira a:

1. **Riconoscere in tempo reale potenziali attacchi osservando pattern nel traffico.**
    - Sfruttando decine di feature derivate da flussi TCP/IP, come dimensioni dei pacchetti, frequenze di trasmissione, distribuzioni temporali e flag TCP per modellare il comportamento del traffico. In questo modo, è in grado di distinguere con alta precisione i pattern associati ad attacchi noti, come DDoS, PortScan o Brute Force, da quelli benigni.
2. **Fornire spiegazioni interpretabili delle predizioni tramite SHAP values**
    - Essendo la scarsa interpretabilità uno dei limiti comuni nei sistemi di machine learning, TrafficML affronta questo problema integrando SHAP (SHapley Additive exPlanations), una tecnica che attribuisce un punteggio di contributo a ciascuna feature rispetto alla predizione del modello. Questo permette agli analisti di comprendere perché una certa connessione è stata classificata come attacco, aumentando la fiducia nelle decisioni del sistema e facilitando il debug o l'aggiornamento delle policy di sicurezza.
3. **Automazione di azioni difensive come logging, allarmi e blocchi IP**

    - TrafficML non si limita alla semplice classificazione, ma può essere esteso per interagire con l'infrastruttura di rete. In presenza di un attacco rilevato, il sistema può:

        - **Registrare l'evento (logging)** per analisi future
        - **Generare notifiche in tempo reale** tramite integrazione con Wazuh (SIEM - Security Information and Event Management)
        - **Bloccare automaticamente l'IP sorgente** o aggiornare le regole del firewall per mitigare la minaccia

    - Questa integrazione operativa trasforma TrafficML in un componente attivo nella catena di difesa di una rete aziendale, con capacità di risposta immediata.

Glossario

- **Intrusion Detection System (IDS):**

  Sistema che monitora il traffico di rete o i log di sistema per rilevare attività sospette o dannose. Può essere basato su firme (signature-based) o su anomalie (anomaly-based).

- **Honeypot:**

Sistema o una componente hardware o software, utilizzata come esca per attrarre gli attacchi informatici, consentendo di studiare i metodi degli aggressori.

- **Security Information and Event Management (SIEM):**

  Piattaforma che centralizza il monitoraggio della sicurezza, raccogliendo e analizzando log da diversi dispositivi per rilevare minacce e supportare la risposta agli incidenti.

- **TCP (Transmission Control Protocol):**

  Protocollo di trasporto orientato alla connessione, usato per garantire la consegna affidabile dei dati su reti IP.

- **IP (Internet Protocol):**

  Protocollo di rete responsabile dell'indirizzamento e dell'instradamento dei pacchetti di dati.

- **Machine Learning:**

  Tecnica dell'intelligenza artificiale che consente ai sistemi di apprendere automaticamente da dati passati e fare predizioni o decisioni senza essere esplicitamente programmati.

- **Apprendimento supervisionato:**

  Task di machine learning che consiste nell'apprendere una funzione a partire da dati di training etichettati

- **Logging:**

  Processo di registrazione degli eventi che si verificano all'interno di un sistema informatico, utile per auditing, debugging e analisi forensi.

- **SHAP (SHapley Additive exPlanations):**

  Tecnica di interpretabilità che assegna a ogni feature un valore che rappresenta il suo contributo alla predizione finale di un modello ML.

- **DoS (Denial of Service):**

  Attacco che mira a rendere un servizio non disponibile sovraccaricandolo con traffico.

- **DDoS (Distributed Denial of Service):**

  Versione distribuita del DoS, condotta da molteplici dispositivi per amplificare l'impatto.

- **PortScan:**

  Tecnica utilizzata per scoprire le porte aperte su un host, utile sia per l'amministrazione che per preparare attacchi.

- **Brute Force:**

  Attacco che tenta sistematicamente tutte le combinazioni possibili per violare una password o accedere a un sistema.

- **Heartbleed:**

  Vulnerabilità nella libreria OpenSSL che permette di leggere porzioni di memoria del server, esponendo dati sensibili come password o chiavi private.

- **Infiltration:**

  Attacco in cui l'aggressore riesce a entrare nella rete interna e muoversi lateralmente o esfiltrare dati senza essere rilevato.

Scelta dell'Algoritmo

Nel contesto di questo progetto, l'obiettivo principale è la rilevazione e la categorizzazione automatica di diversi tipi di traffico di rete, distinguendo tra traffico benigno e varie tipologie di attacchi (ad esempio DDoS, PortScan, Brute Force, ecc.). Questa natura del problema presuppone la presenza di etichette ("label") note per ciascun esempio nel dataset, che indicano la classe di appartenenza di ciascun flusso di traffico.

Per questo motivo, si è scelto di adottare un approccio di classificazione supervisionata anziché ricorrere a tecniche di clustering o a metodi non supervisionati. Il clustering, infatti, sarebbe stato più adatto in assenza di etichette o in presenza di un obiettivo esplorativo per individuare pattern o anomalie sconosciute nei dati. Tuttavia, nel nostro caso, il compito principale è quello di assegnare in modo automatico e preciso la corretta categoria di attacco o di traffico benigno, sfruttando la conoscenza pregressa fornita dalle etichette disponibili. L'approccio di classificazione, inoltre, permette di valutare le performance in modo oggettivo tramite metriche quantitative, garantendo sia la riproducibilità sia la confrontabilità dei risultati.

Infine, la scelta di un algoritmo supervisionato e in particolare di una pipeline strutturatan permette di:

- sfruttare appieno la ricchezza informativa delle etichette;
- ottimizzare le performance rispetto alle classi di interesse;
- facilitare l'interpretazione delle decisioni del modello grazie a strumenti di spiegabilità;
- rendere più agevole l'integrazione della soluzione in un contesto operativo di cybersecurity, dove la precisione nella classificazione delle minacce è fondamentale.

1. Data Preparation

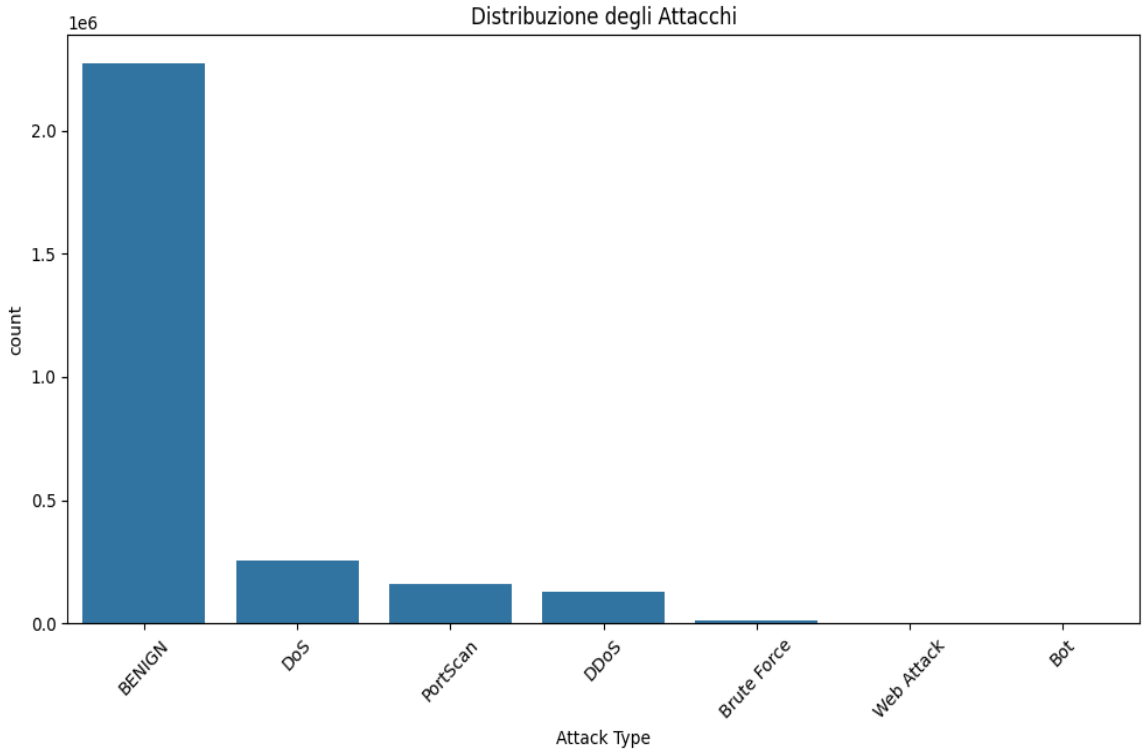
Dataset

Per garantire una copertura ampia e rappresentativa delle diverse tipologie di traffico e di attacchi, sono stati utilizzati molteplici file CSV provenienti dal dataset CICIDS2017. Ogni file rappresenta uno scenario specifico di traffico di rete (ad esempio, attacchi DDoS, PortScan, traffico benigno, ecc.). L'unione di questi file, effettuata tramite caricamento e concatenazione, consente di costruire un dataset complessivo che include sia il traffico ordinario sia molteplici scenari di attacco, aumentando la robustezza e la generalizzabilità del

modello. Questa scelta permette di addestrare e testare l'algoritmo in contesti realistici e variabili, riflettendo la complessità di un ambiente di rete reale.

Pulizia e preparazione etichette

Dopo il caricamento dei dati, si è proceduto a un'attenta fase di pulizia: uniformazione delle colonne, rimozione di duplicati e normalizzazione delle etichette. La mappatura delle etichette tramite un dizionario personalizzato ("attack_map") ha avuto l'obiettivo di ridurre la frammentazione delle classi, accorpando categorie simili e mantenendo solo quelle con sufficiente rappresentatività statistica. Questa decisione è fondamentale per evitare che il modello venga influenzato da classi troppo rare o poco significative, migliorando così la stabilità delle predizioni e l'interpretabilità dei risultati. Inoltre, la pulizia delle etichette e la loro normalizzazione sono passaggi cruciali per garantire coerenza nei dati, condizione imprescindibile per ogni pipeline di machine learning affidabile.



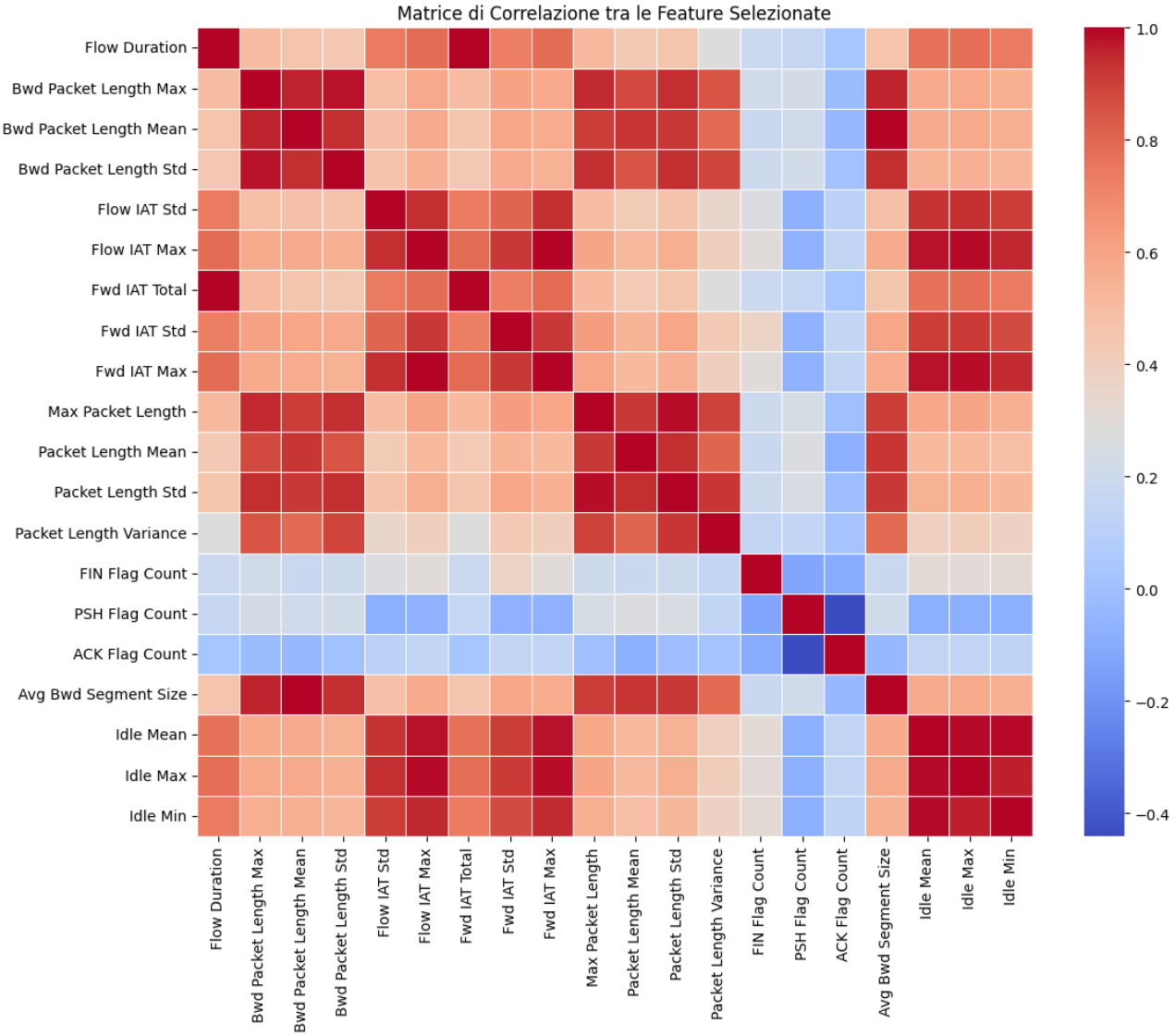
Feature Engineering

Il processo di feature engineering ha incluso diverse operazioni:

- Rimozione di colonne non informative (come Flow ID, indirizzi IP e Timestamp), poiché queste variabili non contengono informazioni predittive rilevanti per la classificazione degli attacchi, ma rischiano di introdurre rumore o bias non desiderati.
- Codifica delle variabili categoriche tramite label encoding, per garantire la compatibilità con gli algoritmi di machine learning che richiedono input numerici.
- Imputazione dei valori mancanti utilizzando la media, scelta che consente di mantenere la coerenza statistica delle variabili senza introdurre distorsioni significative.
- Applicazione della normalizzazione tramite MinMaxScaler: questa operazione è particolarmente importante in presenza di feature con scale diverse, perché consente di

evitare che alcune variabili dominino il processo di apprendimento, favorendo una convergenza più stabile e prestazioni migliori del modello.

- Selezione delle 20 feature più rilevanti tramite SelectKBest con test chi-quadro, per ridurre la dimensionalità del problema, migliorare l'efficienza computazionale e focalizzare l'attenzione sulle variabili realmente discriminanti.



Ogni passo è stato pensato per ottimizzare la qualità e la coerenza del dataset, riducendo il rischio di overfitting e migliorando la comprensibilità delle decisioni del modello.

2. Training Model

Divisione dei Dati

La suddivisione dei dati è stata effettuata tramite il metodo train_test_split con stratificazione, riservando il 20% dei dati al test set. La stratificazione sulla label è stata una scelta fondamentale: in presenza di un dataset multiclass e sbilanciato, è essenziale che tutte le classi siano rappresentate proporzionalmente sia nel training set sia nel test set. Questo

garantisce che la valutazione delle performance del modello sia affidabile e non distorta dalla presenza di classi rare solo in una delle due partizioni. La dimensione del test set (20%) bilancia bene la necessità di avere sufficienti dati per la validazione e la necessità di mantenere un numero adeguato di esempi per l'addestramento.

Bilanciamento

Il dataset originale presenta un marcato sbilanciamento tra le classi (il traffico benigno è molto più numeroso rispetto agli attacchi). Questo squilibrio rischia di portare il modello a favorire la classe maggioritaria, penalizzando la capacità di rilevare attacchi, che spesso sono l'oggetto principale di interesse. Per affrontare questo problema, è stato impiegato il RandomUnderSampler per riequilibrare le classi, riducendo la numerosità delle classi maggioritarie e portando tutte le classi alla stessa dimensione. Questa scelta, seppur a scapito della perdita di alcune istanze della classe prevalente, è necessaria per garantire che il modello apprenda a riconoscere efficacemente anche gli attacchi meno frequenti, migliorando la sua utilità pratica in scenari reali dove la detection degli attacchi è critica.

Addestramento modello

Per l'addestramento è stato scelto il Random Forest Classifier, con 100 alberi e profondità massima di 20. Il Random Forest è stato selezionato per la sua comprovata efficacia sui dati tabellari, la capacità di gestire variabili eterogenee (numeriche e categoriche) e di fornire una buona robustezza rispetto all'overfitting grazie alla combinazione di più alberi decisionali. L'impostazione di 100 alberi e una profondità limitata rappresenta un compromesso tra accuratezza e complessità computazionale, garantendo tempi di addestramento accettabili e una buona generalizzazione. Si è scelto di addestrare il modello sia sui dati originali sia su quelli bilanciati, per confrontare l'impatto del bilanciamento sulle performance e per valutare la reale capacità del modello di gestire situazioni sia tipiche che critiche dal punto di vista della distribuzione delle classi.

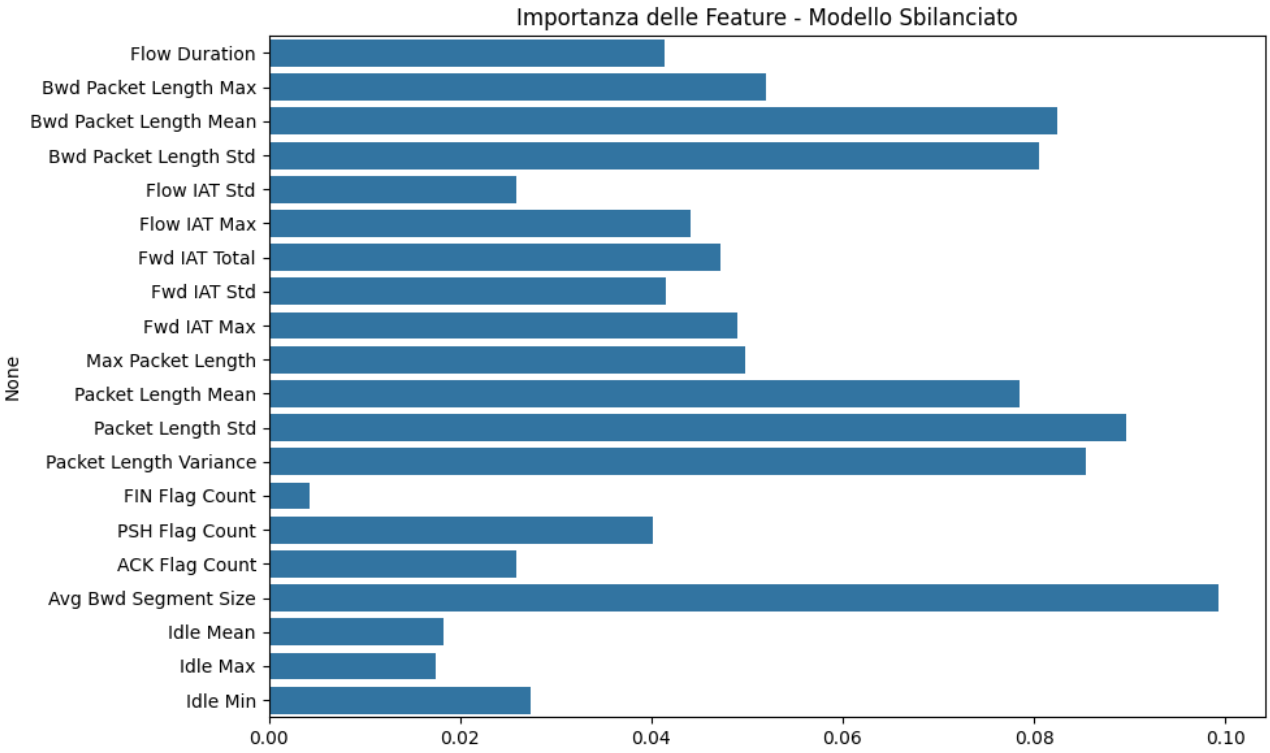
4. Test Model & Metriche

Strategia di Test

Per garantire una valutazione rigorosa e trasparente delle prestazioni del modello, sono state adottate due distinte strategie di test: una basata sul test set con la distribuzione naturale del dataset (sbilanciata) e una sul test set reso bilanciato tramite tecniche di under-sampling.

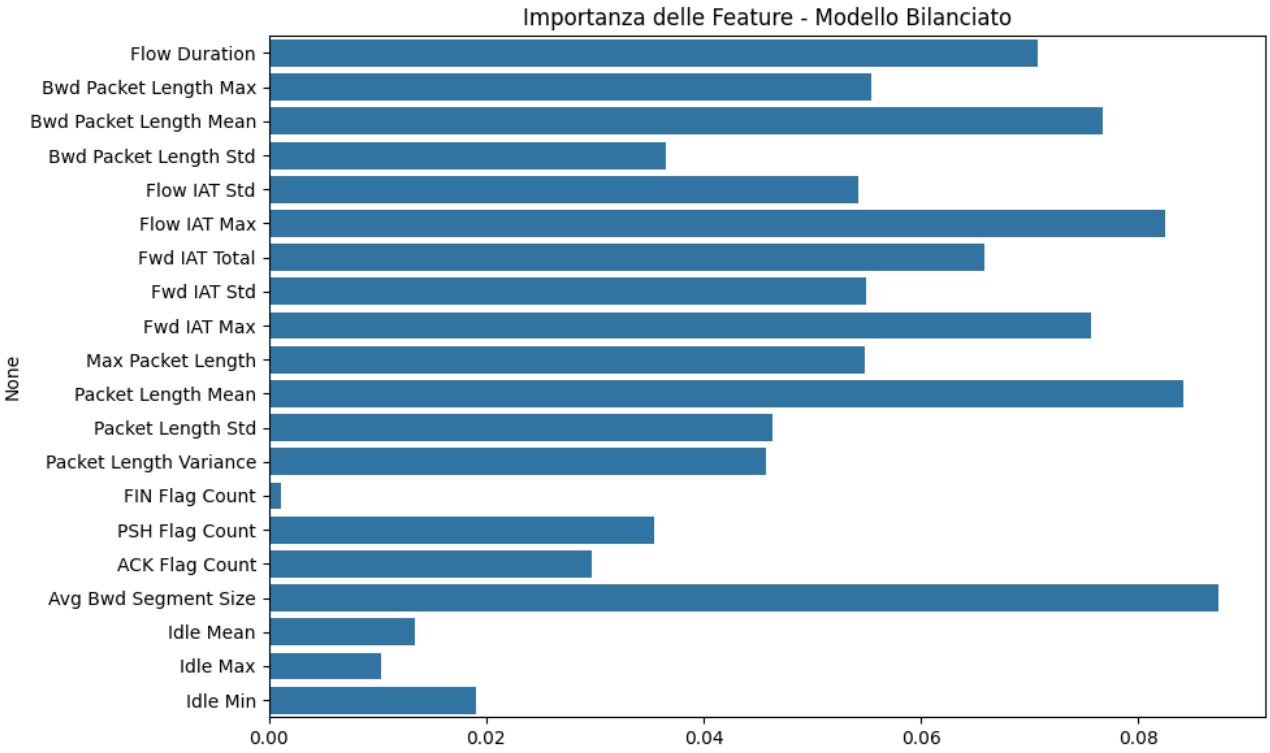
Test sul set sbilanciato:

Questa valutazione riflette il comportamento del modello in condizioni reali, dove il traffico benigno è di gran lunga più frequente rispetto agli attacchi. Testare sul set sbilanciato permette di stimare come il modello si comporterebbe nell'uso reale, evidenziando la capacità di mantenere un basso tasso di falsi positivi senza trascurare le classi minoritarie.

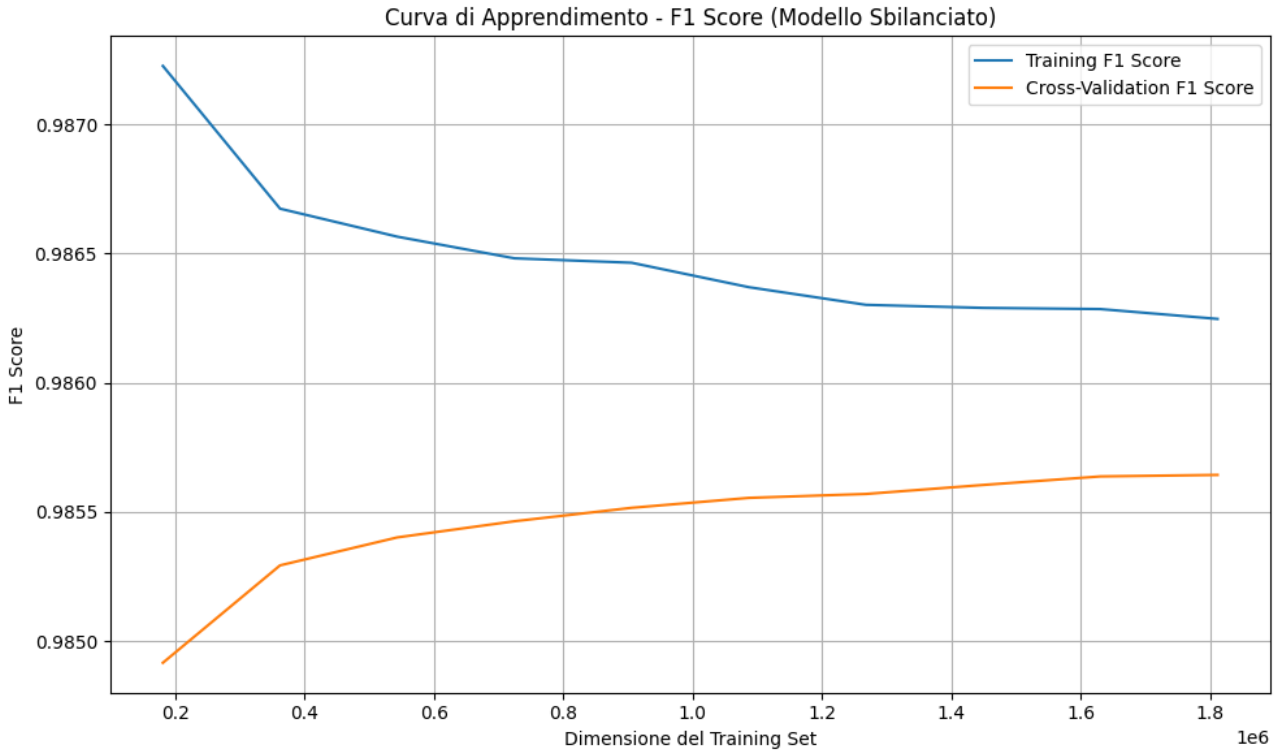


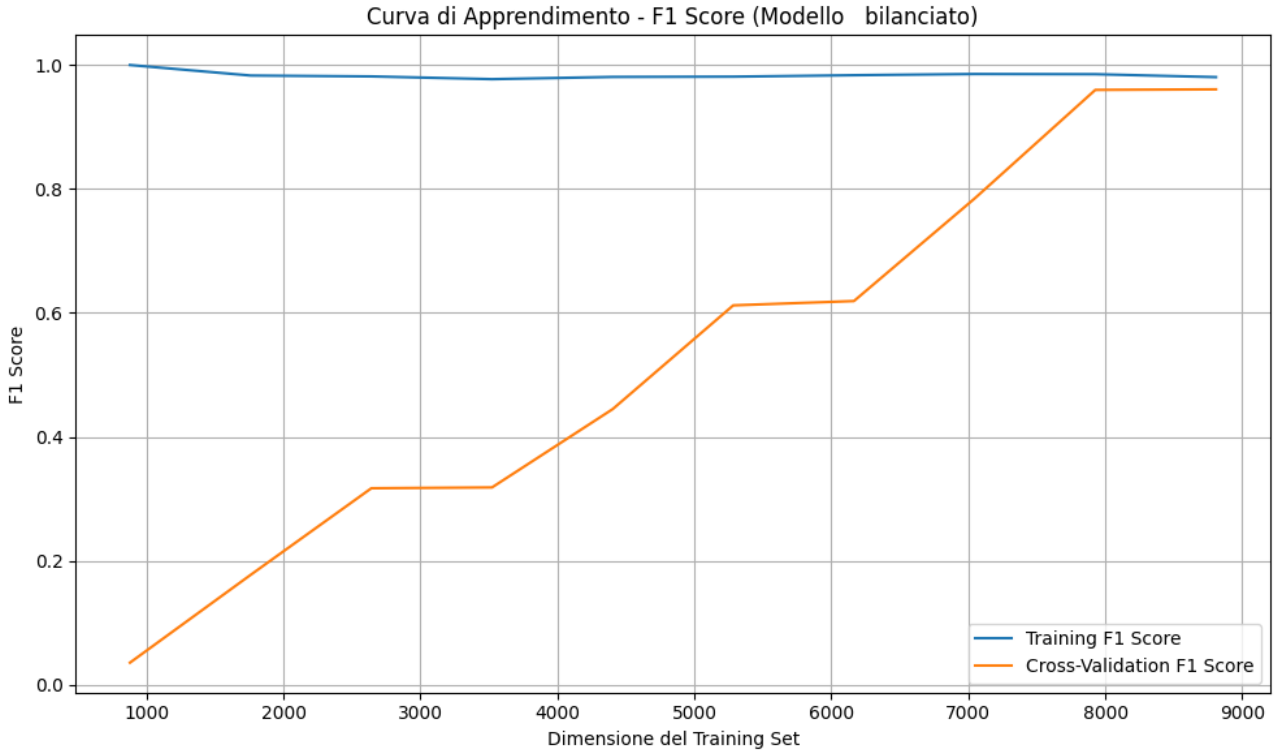
Test sul set bilanciato:

È stata inoltre condotta una valutazione su un test set bilanciato tramite RandomUnderSampler. Questo consente di analizzare in modo più controllato la reale capacità del modello di riconoscere tutte le classi in modo equo, indipendentemente dalla loro frequenza relativa. In particolare, questa strategia aiuta a identificare eventuali difficoltà nella classificazione delle classi minoritarie.



L’utilizzo di entrambe le strategie permette di ottenere un quadro completo: da un lato la performance in scenari realistici, dall’altro la capacità intrinseca del modello di distinguere ogni classe senza essere influenzato dallo sbilanciamento.





Metriche Utilizzate

Per valutare in modo oggettivo e granulare il comportamento del modello, sono state adottate diverse metriche standard del machine learning, ciascuna con un ruolo specifico:

- Accuracy: Indica la percentuale di predizioni corrette sul totale degli esempi. Rappresenta una misura sintetica della performance generale, ma in presenza di sbilanciamento delle classi può risultare poco rappresentativa dell'effettiva qualità del modello, poiché non distingue tra errori sulle classi maggioritarie e minoritarie.

- Precision: Misura la proporzione di predizioni positive che sono effettivamente corrette. Un alto valore di precision riduce il rischio di falsi allarmi, fondamentale in ambito cybersecurity per non sovraccaricare i sistemi di monitoraggio con segnalazioni inutili.

- Recall (Sensibilità): Indica la capacità del modello di individuare tutte le istanze appartenenti a una classe specifica, in particolare le classi di attacco. Un recall elevato è essenziale per non lasciarsi sfuggire minacce reali.

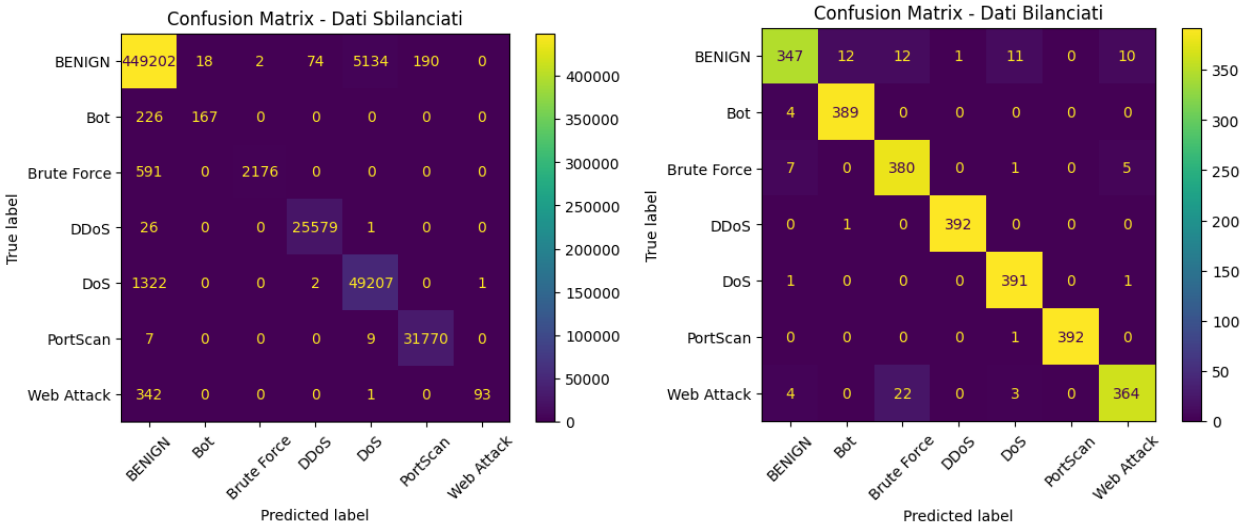
- F1 Score: È la media armonica tra precision e recall. È particolarmente indicato quando è importante bilanciare la capacità di catturare tutte le istanze positive (recall) e la precisione delle predizioni (precision), soprattutto per le classi di attacco che sono di maggiore interesse nel contesto della sicurezza informatica.

- Classification Report: Fornisce una panoramica dettagliata delle metriche principali (precision, recall, F1-score) per ciascuna classe, offrendo così una diagnosi approfondita delle prestazioni su tutte le tipologie di traffico o attacco.

```
--- Test Set (Sbilanciato) ---
Accuracy: 0.9859646023951673
F1 Score: 0.9858148822759902
Precision: 0.9865337275516963
Recall: 0.9859646023951673
              precision    recall  f1-score   support

           0       0.99      0.99      0.99    454620
           1       0.90      0.42      0.58       393
           2       1.00      0.79      0.88      2767
           3       1.00      1.00      1.00     25606
           4       0.91      0.97      0.94     50532
           5       0.99      1.00      1.00     31786
           6       0.99      0.21      0.35       436

    accuracy                           0.99    566140
   macro avg       0.97      0.77      0.82    566140
weighted avg       0.99      0.99      0.99    566140
```

```
--- Test Set (Bilanciato) ---
Accuracy: 0.965103598691385
F1 Score: 0.9648331730445453
Precision: 0.9653567315656155
Recall: 0.965103598691385
              precision    recall  f1-score   support

           0       0.96      0.88      0.92       393
           1       0.97      0.99      0.98       393
           2       0.92      0.97      0.94       393
           3       1.00      1.00      1.00       393
           4       0.96      0.99      0.98       393
           5       1.00      1.00      1.00       393
           6       0.96      0.93      0.94       393

    accuracy                           0.97      2751
   macro avg       0.97      0.97      0.96      2751
weighted avg       0.97      0.97      0.96      2751
```

- Matrice di Confusione: La matrice di confusione offre una visione dettagliata dei risultati delle predizioni, mostrando come le istanze di ciascuna classe reale sono state classificate dal modello. Permette di identificare facilmente quali classi vengono spesso confuse tra loro e di diagnosticare errori sistematici, evidenziando eventuali problematiche nella distinzione tra categorie specifiche. Questo strumento è particolarmente utile per ottimizzare ulteriormente il modello, individuando le classi per cui sono necessari interventi di tuning o raccolta dati aggiuntivi.



L'adozione congiunta di queste metriche consente di valutare sia la performance globale che quella specifica per ogni classe, fornendo un quadro completo e affidabile dell'efficacia del modello in tutti gli scenari possibili.

5. Spiegabilità

Per aumentare la trasparenza e l'affidabilità del modello, è stato utilizzato SHAP (SHapley Additive exPlanations) per analizzare l'importanza delle feature. SHAP è uno degli strumenti più avanzati per la spiegabilità dei modelli di machine learning, poiché permette di quantificare in modo preciso il contributo di ciascuna variabile alla predizione finale, sia a livello globale che per la singola osservazione. Questa scelta è fondamentale in un contesto

come quello della sicurezza informatica, dove è importante non solo ottenere buone prestazioni, ma anche comprendere le motivazioni delle decisioni del modello, individuare eventuali bias e fornire giustificazioni trasparenti agli stakeholder.

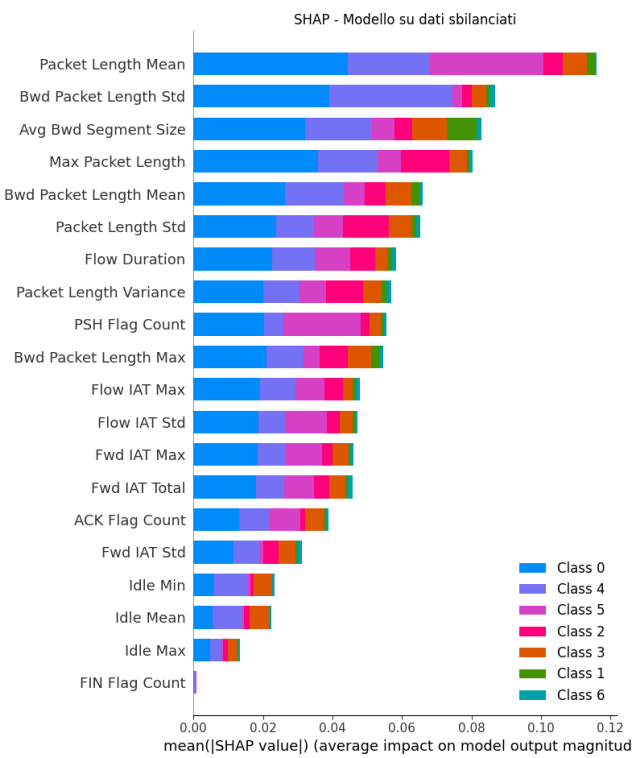
Implementazione pratica:

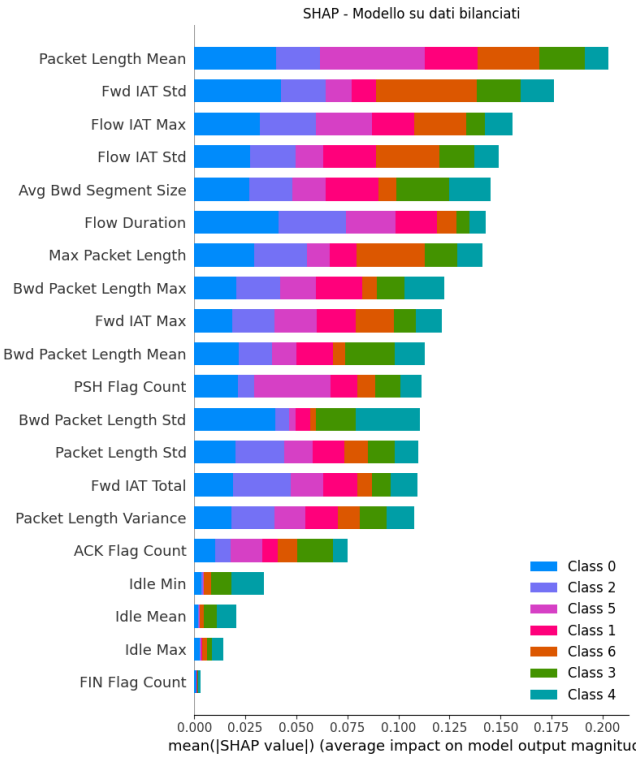
SHAP è stato applicato ai modelli addestrati sia su dati sbilanciati che su dati bilanciati. Vengono generati summary plot che mostrano graficamente l'influenza di ciascuna feature sulle decisioni del modello, permettendo di individuare quali variabili hanno il maggiore impatto nelle classificazioni. Inoltre, l'analisi locale consente di spiegare predizioni specifiche su singoli flussi di traffico, evidenziando il motivo per cui un certo esempio è stato classificato come benigno o come attacco.

Valore aggiunto nel contesto della sicurezza informatica:

L'utilizzo di SHAP è cruciale in ambito cybersecurity, dove la sola accuratezza predittiva non è sufficiente: è essenziale fornire spiegazioni chiare e dettagliate delle decisioni del modello. SHAP consente di:

- Identificare rapidamente quali feature sono maggiormente responsabili di una classificazione, facilitando l'individuazione di pattern sospetti o di possibili bias nel dataset.
- Verificare la coerenza del modello con le conoscenze di dominio, confermando che certe caratteristiche del traffico siano effettivamente indicative di attacchi specifici.
- Individuare eventuali anomalie nelle decisioni, come predizioni influenzate da feature inattese, e intervenire tempestivamente per migliorare la pipeline.





6. Salvataggio del modello

Al termine della fase di addestramento e valutazione, l'intera pipeline di machine learning viene salvata utilizzando la libreria joblib. In particolare, vengono serializzati e memorizzati su disco: il modello Random Forest addestrato, il codificatore delle etichette (LabelEncoder), lo scaler utilizzato per la normalizzazione delle feature (MinMaxScaler), l'imputer per la gestione dei valori mancanti e la lista delle feature selezionate tramite SelectKBest.

Questa strategia garantisce la possibilità di riutilizzare in modo immediato l'intero workflow di preprocessing e predizione su nuovi dati, mantenendo piena coerenza tra la fase di training e quella di inferenza. In altre parole, ogni nuovo dato potrà essere sottoposto agli stessi identici passaggi di trasformazione e selezione delle feature, assicurando che le predizioni del modello siano consistenti e affidabili.

Il salvataggio di tutti gli oggetti della pipeline non solo facilita la riproducibilità degli esperimenti scientifici, ma consente anche una facile portabilità e integrazione del modello in sistemi reali o in ambienti di produzione. Questo approccio minimizza il rischio di errori dovuti a differenze tra ambienti di sviluppo e ambienti operativi, velocizza il deployment e rende possibile l'aggiornamento o il riaddestramento del modello senza dover ricostruire manualmente l'intera catena di preprocessing.

Infine, la scelta di joblib come strumento di serializzazione è motivata dalla sua efficienza sia in termini di velocità che di compatibilità con oggetti complessi di scikit-learn, particolarmente adatta per pipeline articolate.

Conclusioni

Le decisioni prese lungo tutta la pipeline sono state guidate dall'esigenza di costruire un sistema robusto, interpretabile e facilmente riutilizzabile, capace di affrontare in modo efficace le sfide poste dai dati reali e dal contesto applicativo della sicurezza del traffico di rete. Ogni scelta è stata ponderata in funzione dell'affidabilità del risultato, della trasparenza e della possibilità di trasferire il modello in un contesto operativo.

Possibili Miglioramenti Futuri:

- Implementare un sistema di notifiche automatiche per dispositivi mobile (android/iOS) per avvisare degli attacchi che stanno avvenendo.
- Creazione di una dashboard interattiva per visualizzare predizioni e spiegazioni SHAP in tempo reale.