Alberi di Decisione per Intrusion Detection

Lorenzo Pesci

2 novembre 2018

1 Introduzione

Lo scopo di questo elaborato è quello di utilizzare diverse implementazioni di *Decision Trees* relativamente al problema dell'Intrusion Detection.

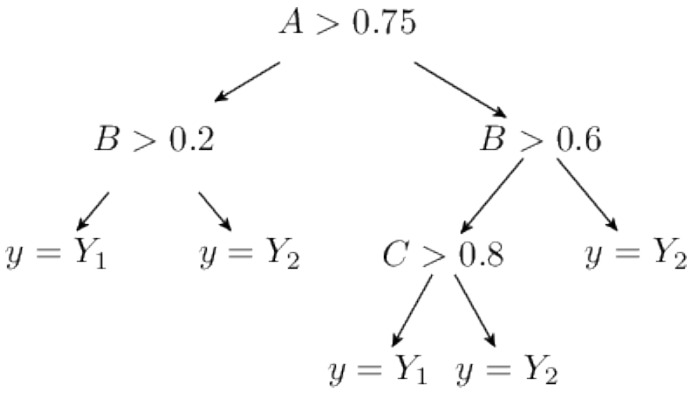

Figura 1: Un esempio di albero di decisione. [*ce.unipr.it*]

1.1 Alberi di Decisione

Nell'ambito del *Machine Learning* un metodo semplice ed efficace per risolvere problemi di classificazione è quello di utilizzare gli alberi di decisione.

Un albero di decisione, costruito a partire da un insieme di dati iniziali (**dataset**), è un albero in cui ogni nodo interno è associato ad una particolare "domanda" su una certa caratteristica dei dati. Da questo nodo escono tanti archi quanti sono i possibili valori che la caratteristica può assumere, fino a raggiungere le foglie che indicano la categoria associata alla decisione.

Lo scopo è quello di allenare questo albero su alcuni esempi di dati (*training set*) in modo da poter classificare altri dati non precedentemente visionati (*test set*) con un certo livello di confidenza.

Un problema comune con gli alberi di decisione è che spesso si adattano fin troppo bene ad alcune caratteristiche specifiche solo del *training set*, e questo porta ad un calo delle prestazioni sul *test set*.

1.2 Intrusion Detection

L'Intrusion Detection è un sistema che viene utilizzato per rilevare e contrastare gli attacchi informatici che avvengono in un sistema o in una determinata rete. Generalmente un intruso è definito come un sistema, un programma o una persona che tenta di entrare in un sistema di informazione o eseguire un'azione non legalmente consentita.
In particolare in questo esercizio ci concentreremo su quattro tipologie di attacchi:
-Denial of Service Attacks (DoS): è un tipo di attacco in cui l'hacker rende le risorse di calcolo o di memoria troppo occupate o troppo piene per servire legittime richieste di rete e quindi negare agli utenti l'accesso a una macchina. (es. apache)
-Remote to User Attacks (R2L): un attacco da remoto a utente è un attacco in cui un utente invia pacchetti a una macchina su Internet, a cui non ha accesso per esporre le vulnerabilità delle macchine e i privilegi di exploit che un utente locale avrebbe sul computer. (es. xlock)
-User to Root Attacks (U2R): sono exploit in cui l'hacker si avvia sul sistema con un normale account utente e tenta di sfruttare le vulnerabilità nel sistema per ottenere privilegi di super utente. (es. perl)
-Probing: è un attacco in cui l'hacker esegue la scansione di una macchina o di un dispositivo di rete al fine di determinare punti deboli o vulnerabilità che potrebbero essere sfruttati in seguito per compromettere il sistema. Questa tecnica è comunemente usata nel data mining. (es. portsweep)

2 Esperimento

L'esperimento è stato condotto utilizzando il data set della KDD Cup 1999. Questo set di dati è stato fornito dal MIT Lincoln in collaborazione con l'Air Force Lan ed è utile per verificare e utilizzare il classificatore di sistema.

Questo è il set di dati utilizzato per la terza competizione internazionale di scoperta della conoscenza e strumenti di data mining, che si è svolta in concomitanza con la KDD-99. La quinta conferenza internazionale sulla scoperta della conoscenza e il data mining. L'obiettivo della competizione era costruire un rilevatore di intrusione di rete, un modello predittivo in grado di distinguere tra connessioni "cattive", chiamate intrusioni o attacchi e connessioni "buone" normali. Questo database contiene un set standard di dati da verificare, che include un'ampia varietà di intrusioni simulate in un ambiente di rete militare.

L'obiettivo dell'esperimento è quello di riprodurre con maggior accuratezza possibili le tabelle 2 e 3 proposte in (Amor et al.2004).

Table 2: PCC's relative to five classes

TRAINING SET	TESTING SET
Decision tree	
99.99% (99.99%)	92.28% (91.81%)

Figura 2: Tabella 2 (Amor et al. 2004)

Table 3: Confusion matrices relative to five classes

Decision tree					
→	Normal	DOS	R2L	U2R	Probing
Normal (60593)	**99.50%** (**99.43%**)	0.13% (0.14%)	0.01% (0.02%)	0.01% (0.02%)	0.36% (0.39%)
DOS (229853)	2.76% (2.94%)	**97.24%** (**96.57%**)	0.00% (0.10%)	0.00% (0.00%)	0.00% (0.39%)
R2L (16189)	96.55% (75.77%)	0.02% (2.79%)	**0.52%** (**0.45%**)	0.15% (4.27%)	2.76% (16.71%)
U2R (228)	79.82% (23.25%)	2.63% (0.00%)	1.75% (5.26%)	**7.89%** (**13.60%**)	7.89% (57.89%)
Probing (4166)	19.54% (15.22%)	5.16% (6.67%)	0.34% (0.19%)	0.00% (0.00%)	**74.96%** (**77.92%**)
PCC	92.06% (92.80%)				

Figura 3: Tabella 3 (Amor et al. 2004)

3 Implementazione

Il progetto è stato sviluppato con Python versione 3.7 e richiede l'installazione delle librerie *pandas* e *sklearn*.

3.1 preprocessing.py

Nel file *preprocessing.py* vengono mappate tutte le possibili tipologie di attacchi con l'intero corrispondente. In particolare *Normal Attack = 0*, *DOS Attack = 1*, *R2L Attack = 2*, *U2R Attack = 3* e *Probing = 4*.

Inoltre poichè gli algoritmi di costruzione di un albero di decisione classificano correttamente gli esempi del training set, a condizione che nel training set non siano presenti dati rumorosi. Per far fronte a questa problematica è stata applicata la codifica ***one hot*** alle variabili categoriche corrispondenti alle colonne *flag*, *protocol_type* e *service*, in modo da ridurre il rumore dei dati.

3.2 dt_predict.py

Nel file *dt_predict* vengono effettuate le predizioni sui dati.
Viene utilizzato *DecisionTreeClassifier*, preso dalla libreria *sklearn* che contiene il classificatore basato sugli alberi di decisione. Come criterio di decisione è stato usato *entropy*, come splitter è stato usato *random*. Inoltre è stata fissata la massima profondità raggiungibile dall'albero (max_depth=15) e il numero minimo di campioni richiesti in una foglia (min_samples_leaf=6). Questi parametri sono stati scelti in base ai parametri contenuti nella pagina Data Mining Techniques for Intrusion Detection.

3.3 data_names.py

Nel file *data_names* sono presenti due liste, una contenente i features names e l'altra contente gli attribute names.

3.4 Test

Sono stati effettuati numerosi test, utilizzando le due diverse tipologie di criteri per gli alberi di decisione, *entropy* e *gini*, le due diverse tipologie di split, *best* e *random*. Sono stati inoltre modificati gli interi assegnati a *max_ depth* e *min_ samples_ leaf*, avendo cura di combinare tutte le possibilità.

4 Risultati

Di seguito è riportato il miglior risultato ottenuto dai vari test:

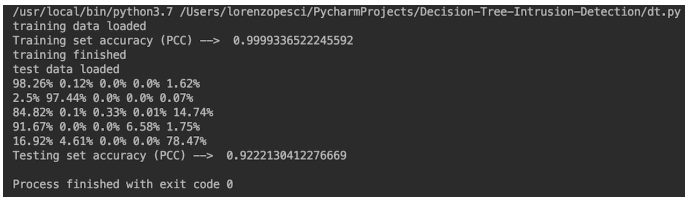

Figura 4: Miglior test effettuato

Riportiamo i dati ottenuti in due distinte tabelle:

TRAINING SET	**TESTING SET**
99.99%	92.22%

Tabella 1: Dati della Figura 4, riportati in un grafico.

$\longrightarrow$	**Normal**	**DOS**	**R2L**	**U2R**	**Probing**
Normal	**98.26%**	0.12%	0.0%	0.0%	1.62%
DOS	2.5%	**97.44%**	0.0%	0.0%	0.07%
R2L	84.82%	0.1%	**0.33%**	0.01%	14.74%
U2R	91.67%	0.0%	0.0%	**6.58%**	1.75%
R2L	16.92%	4.61%	0.0%	0.0%	**78.47%**

Tabella 2: Dati della Figura 4, riportati in un grafico.

5 Conclusione

Dall'analisi del set di dati della KDD Cup 1999, è apparso evidente che erano necessari rivelatori specializzati per classificare i vari tipi di attacchi informatici che avvengono in un sistema o in una rete. Gli attacchi di tipo DoS o Probing si sono dimostrati molto facili da classificare utilizzando modelli semplici. Invece gli attacchi più rari come R2L e U2R necessitano di rivelatori più sofisticati.