FindOutlier
Progetto finale di Gestione di Reti a.a. 2020/2021

Dipendenze di Python
Easysnmp (https://pypi.org/project/easysnmp/)
Plotly (https://plotly.com/python/getting-started/)

I link sono delle guide all'utilizzo delle librerie e alla loro installazione.

Requisiti
Python 3.x
SNMP (https://martinsblog.dk/linux-how-do-i-enable-snmp-on-ubuntu/)

Esecuzione del programma
Prima di eseguire il programma è importante digitare il comando

- *export MIBS=ALL*

per settare le MIB environment variables. Se questo passo viene saltato, il programma ritornerà un messaggio di errore.

Per eseguire il programma bisogna trovarsi nel path corretto, ovvero nella cartella in cui è salvato il file .py, e lanciare il comando

- *python3 outlier.py*

Se si volesse arrestare l'esecuzione prima della terminazione dell'esecuzione basterà digitare

- *ctrl + C*

Chiaramente non si otterrà alcun tipo di risultato.

Funzionamento
Il programma ha un host predefinito su cui lavorare e usa il suo indirizzo IP per fare il polling delle sue porte attive, grazie a una snmp walk. Per recuperare queste porte, la snmp walk esplora il MIB ifOperStatus per ogni porta e memorizza il numero delle porte che risultano attive.

Successivamente, il programma procede valutando l'utilizzo di ogni porta eseguendo due snmp get (per ogni porta) distanziate da un intervallo di tempo della durata di 4 secondi. Entrambe le get ritornano il valore del MIB ifOutOctets, ovvero il numero di otteti trasmessi (gruppi di otto bits) da una certa interfaccia; l'utilizzo medio di una porta è calcolato con la seguente formula:

- *outOctets(2) - outOctets(1) / 4*

(4 sono i secondi passati tra una get e la successiva).

Dopo questi passi preliminari, la funzione find_outliers fa la gran parte del lavoro. Per prima cosa, per poter settare il lower e l'upper limit, il programma calcola Q1 e Q3, i percentili utilizzati nel calcolo dell'IQR (InterQuartile Range). L'IQR è una misura di variabilità, basata sul dividere il data set in quartili; questi quartili dividono il data set in quattro parti uguali e sono chiamati Q1, Q2 e Q3 (Q1 e Q3 sono i soli usati dal programma per raggiungere il suo obiettivo).

Per calcolare il lower e l'upper limit ho dovuto cambiare leggermente la regola standard per trovare gli outliers con l'IQR: poiché è impossibile che un'interfaccia trasmetta un numero negativo di otteti, il lower limit non può avere un valore negativo, che avrebbe se calcolato con la formula:

- *Q1 - 1.5 * IQR*

Quindi, per evitare questo problema e per trovare le porte che stanno effettivamente inviando pochi bits, il programma calcola il lower limit con questa formula:

- *Q1 / 1.5*

Si tratta di una scelta personale, dove la decisione di dividere è stata presa per avere un valore ragionevolmente basso e che non coincidesse esattamente con il percentile.

La regola standard è comunque usata per calcolare l'upper limit:

- *Q3 + 1.5*IQR*

Infine, il programma chiama una funzione che crea un grafico usato per mostrare i dati dell'analisi appena svolta.