Università degli Studi di Pisa

Facoltà di Scienze matematiche, Fisiche e Naturali

Corso di studi in Informatica

Rilevazione di anomalie di rete mediante analisi su serie temporali

Candidato: Salvatore Costantino

Relatore: Luca Deri

Anno accademico 2018-2019

1. Introduzione

Il mondo odierno è sempre più interconnesso: se da un lato questo fenomeno porta innumerevoli vantaggi sociali, culturali, economici e finanziari, dall'altro si evidenziano problemi legati alla **sicurezza**, **privacy** e **gestione** degli host connessi in rete, che devono essere opportunamente gestiti; è stato infatti stimato che il costo economico dovuto ad attacchi informateci supera ampiamente i 100 miliardi di dollari annui [1]: c'è quindi una reale e concreta necessità di rilevare e mitigare in tempi brevi minacce, e più in generale anomalie di rete. La gestione delle problematiche di rete si articola essenzialmente in due fasi:

- **rilevazione dell'anomalia**, ovvero ciò che si discosta in modo più o meno netto dalla normalità; il concetto di "normalità" a volte non è semplice da definire, e viene modellato in maniera diversa in base alle varie tecniche di rilevazione: per esempio, in questo lavoro di tesi, l'enfasi è posta sul fattore tempo;
- **gestione e mitigazione dell'anomalia**, la quale può essere effettuata manualmente o per mezzo di procedure informatiche automatizzate: alcuni applicativi registrano le varie attività di rete sospette, in modo da lasciare degli allarmi ai soggetti interessati alle analisi di rete; altri agiscono in modo più deciso, cercando autonomamente di mitigare o risolvere la situazione anomala che si è verificata. Ovviamente gli approcci sopra citati possono coesistere.

Esistono varie soluzioni che tentano di risolvere il problema della rilevazione e gestione di anomalie, ognuna con i suoi pregi e difetti. In particolare, in questo lavoro di Tesi viene proposta una nuova metodologia per rilevare e mitigare anomalie, che vuol essere d'aiuto ai soggetti addetti alla gestione delle reti.

1.1. Struttura della Tesi

La Tesi è organizzata come segue:

- La **sezione 2** tratta lo stato dell'arte relativo alle soluzioni esistenti per individuare anomalie di rete;
- Gli obiettivi e la motivazione del lavoro svolto vengono discusse rispettivamente nella **sezione 3** e nella **sezione 4**;
- La **sezione 5**, illustra brevemente le novità introdotta da questo lavoro di tesi
- Nella **sezione 6** vengono analizzate le scelte progettuali effettuate, e viene descritta l'architettura del software proposto;
- La validazione della soluzione implementata viene discussa nella **sezione 7**;
- Si termina con i lavori futuri e le conclusioni, presentati rispettivamente nella **sezione 8 e 9**;
- Nella **sezione 10** è presente la bibliografia;
- Nella **sezione 11** è possibile trovare il link al sistema realizzato;

2. Stato dell'Arte

Attualmente, per rilevare anomalie di rete vengono impiegati dei sistemi chiamati *intrusion detection system* (**IDS**). Essi si possono dividere in due macro-categorie [2]:

- *signature-based IDS*: individuano anomalie basandosi su pattern comportamentali e strutturali specifici di attacchi noti. Leggere varianti di attacchi e minacce non note non vengono rilevate da questa tipologia di applicativo, quindi si ha un alto tasso di falsi negativi. D'altra parte, essi hanno un basso numero di falsi positivi [3];
- *anomaly-based IDS*: apprendono il comportamento dei parametri di un sistema, ed individuano un'anomalia quando il comportamento corrente differisce in modo netto da ciò che si è appreso durante la fase di training, oppure quando un certo parametro viene classificato come anomalia, grazie a dati di training etichettati. In generale, più dati si hanno nella fase di apprendimento, migliore sarà la capacità di rilevazione. Per costruzione, sono in grado di rilevare anche minacce non note, ma sono affetti da un alto tasso di falsi positivi.

Siccome la Tesi si propone l'obiettivo di rilevare anomalie su serie temporali, vengono considerati anche sistemi che non sono catalogati propriamente come IDS, ma nascono con l'idea di rilevare eventi anormali in serie temporali; in tal modo è possibile confrontare il software prodotto con le tecniche attualmente usate per analizzare le serie temporali.

2.1. Signature-based Intrusion Detection System

I signature-based IDS come **snort** vengono configurati mediante un insieme di regole per individuare pacchetti malevoli; solitamente quando una regola "matcha" un pacchetto viene intrapresa un'azione (alert, drop ecc.). In snort, per esempio, le regole hanno la seguente forma [4]:

```
alert tcp any -> 192.168.1.0/24 70
(content:"|00 01 86|"; msg: "mountd access";)
```

dove il testo all'esterno delle parentesi costituisce la *rule header*, mentre all'interno compaiono le *rule options*; in particolare:

- all'interno dell'intestazione compaiono l'azione da intraprendere, il protocollo, gli indirizzi IP e le porte sorgente e destinazione;
- le opzioni permettono di filtrare più in profondità i vari pacchetti, e consentono di aggiungere delle informazioni di log: in questo caso si specifica la stringa da ricercare nel payload ed il testo da stampare in caso di matching.

Suricata è un ulteriore esempio di signature-based IDS, il cui formato delle regole è praticamente uguale a quello visto per snort.

Zeek (ex Bro) è un IDS che offre anche la funzionalità di rilevazione di anomalie basandosi su firma: esso non è un classico signature-based intrusion detection system, ma utilizza altre tecniche più complesse di individuazione di minacce come l'analisi comportamentale [5].

L'architettura di Zeek è costituita da due livelli principali: l'*event engine* e il *policy script interpreter*.

L'event engine trasforma il flusso di pacchetti, che riceve dal livello sottostante, in eventi ad alto livello: essi descrivono e riportano le informazioni legate ad una certa attività di rete, senza eseguire nessun tipo di analisi sull'evento generato; tale analisi (e/o la corrispondente azione) viene invece svolta dal policy script interpreter, che per ogni evento generato dal livello appena discusso, invoca l'event handler corrispondente, scritto nel linguaggio di scripting di Zeek.

In tale applicativo, un esempio firma (di ovvia semantica) è il seguente:

```
signature my-first-sig {
    ip-proto == tcp
    dst-port == 1078
    payload /.*root/
    event "Found root!"
}
```

Qualora ci fosse corrispondenza tra firma e (un) pacchetto della connessione, l'evento *signature_match* verrebbe generato:

```
event signature_match(state: signature_state, msg: string,
                      data: string)
```

dove:

- *state* contiene informazioni più dettagliate sulla connessione che ha generato l'evento
- il contenuto di *msg* è "Found root!"
- *data* contiene l'ultima parte del payload che ha matchato con l'espressione regolare ".*root"*

A questo punto lo script che gestisce l'evento generato, effettuerà le opportune analisi ed azioni (per esempio potrebbe generare un alert).

Come si può vedere, queste tecniche di rilevazione di anomalie prevedono pattern troppo rigidi in grado di rilevare solo minacce già note: piccolissime variazioni in un attacco conosciuto o nuovi attacchi non vengono individuati da questa famiglia di IDS, poiché non ne è stata scritta ancora la corrispondente firma; il numero di falsi positivi è ovviamente molto contenuto, in quanto tali applicativi sono costruiti per individuare pattern anomali noti.

2.2. Anomaly-based Intrusion Detection System

I vari anomaly-based IDS usano diverse tecniche di rilevazione, ma in generale condividono le seguenti fasi o livelli [6]:

- *Parametrizzazione*: le metriche o i parametri da analizzare vengono rappresentati in una certa forma stabilita a priori, consona alle analisi da effettuare;
- *Training o allenamento*: il comportamento normale (e anormale) del sistema viene appreso e ne viene costruito il modello corrispondente;
- *Rilevazione*: il modello costruito viene confrontato con il traffico attuale alla ricerca di istanze anomalie.

Gli anomaly-based IDS possono essere suddivisi in tre tipi principali: *statistical-based* (il comportamento del sistema è trattato da un punto di vista statistico), *knowledge-based* (si cerca di apprendere il comportamento desiderato, utilizzando i dati di sistema disponibili (protocolli utilizzati, traffico di rete, specifiche ecc.)) e *machine learning-based* (viene costruito un modello rappresentante il comportamento normale (e talvolta anormale) delle istanze da analizzare.

2.2.1. Statistical-based

Gli IDS che appartengono a questa famiglia, catturano il traffico di rete e costruiscono un modello che rappresenta il suo comportamento stocastico. Possono essere profilate varie metriche come il traffico in entrata/uscita, flussi, pacchetti in entrata/uscita eccetera; per ogni metrica individuata si possono considerare centinaia di descrittori del traffico di rete, come la media, varianza, quantili, funzione di distribuzione, eccetera [7].
Si ha un'anomalia quando il comportamento statistico attuale si discosta "troppo" dal modello creato.
Le prime tecniche statistiche si basavano sulla costruzione di modelli ad una variabile, i cui parametri erano rappresentati da variabili aleatorie gaussiane indipendenti. A queste variabili veniva associato un valore di treshold (ovvero un valore soglia, il cui superamento rappresenta un evento eccezionale), entro il quale, il comportamento veniva considerato normale (utilizzando per esempio test statistici che si basano sulla *three-sigma-rule*). Successivamente furono proposte delle soluzioni che prevedevano la correlazione tra più variabili, che si dimostrarono essere più precisi dell'approccio illustrato precedentemente. Altre tecniche effettuano analisi su serie temporali, considerando gli eventi accaduti in un certo lasso temporale e mettendoli in relazione con ciò che si è osservato precedentemente.

Un esempio di test statistico è il test chi quadrato, che utilizza la variabile test chi-quadro, così definita [8]:

$$\chi^2 = \sum_{i=1}^{n} \frac{(\mathrm{X}_i - \mathrm{E}_i)^2}{\mathrm{E}_i}$$

Dove *n* è il numero delle variabili da considerare, X_i è il valore dell'i-esima variabile osservate ed E_i è il suo valore atteso. Fisando l'errore tollerato e considerando le tavole della distribuzione chi-quadro è possibile stabilire se il campione di osservazioni considerato presenta delle anomalie.

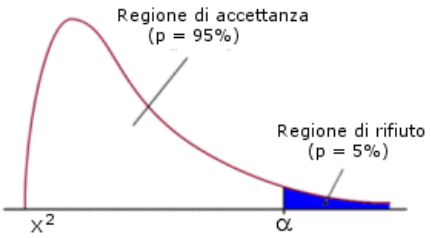


Figura 1: Test statistico

Uno dei grandi vantaggi dei sistemi statistical-based, è dovuto al fatto che non richiedono alcuna conoscenza preventiva sul sistema che stanno analizzando, ma ne assimilano via via il comportamento basandosi sulle osservazioni effettuate; d'altra parte, un attaccante potrebbe generare traffico in modo da far assimilare all'IDS un comportamento anomalo, in modo che un eventuale attacco non venga rilevato. Generalmente gli applicativi appartenenti a questa famiglia di IDS possiedono una notevole capacità di rilevazione di attività anomale.

2.2.2. Knowledge-based

Fanno parte di questi famigli i cosiddetti "expert-system". Essi classificano i dati in input secondo un insieme di regole, in tre fasi distinte: nella prima fase vengono estratti dai dati di training le classi e i relativi attributi rilevanti, ovvero gli oggetti che costituiscono il dominio d'interesse. Successivamente, dal modello costruito nella prima fase, vengono dedotte le regole di classificazione dei dati. Nell'ultima fase, in base alle procedure stabilite precedentemente, vengono classificati i dati in input da controllare.

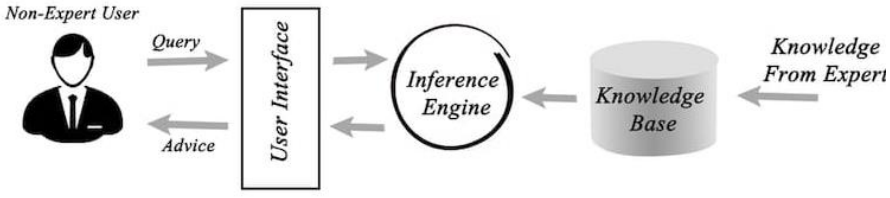


Figura 2: Expert System

Un'altra tipologia di knowledge-based IDS, sono i sistemi basati sulle specifiche: il modello desiderato viene costruito da esperti del settore in base a delle regole (le specifiche), cercando di catturare il normale comportamento del sistema; più la

specifica risulta completa, migliore è la capacità di rilevazione. Le specifiche possono essere modellate tramite strumenti formali come gli automi a stati finiti, usati soprattutto per rappresentare le attività di rete.

La robustezza e la flessibilità sono i punti di forza dei knowledge-based IDS; la costruzione di una buona base di conoscenza risulta però assi difficoltosa e dispendiosa in termini di tempo.

2.2.3. Machine Learning-based

Le tecniche basate sul machine learning, si propongono come obiettivo la costruzione di modelli in grado di classificare la natura di nuove istanze che devono essere analizzate. I modelli vengono costruiti durante la cosiddetta fase di training, utilizzando i dati del training set, che solitamente sono etichettati. Esistono anche (più raramente) delle soluzioni che prevedono dati non etichettati, come mostra il lavoro svolto nella costruzione del NIDS (network IDS) Kitsune [9].

Quest sistemi hanno la capacità di migliorare la loro capacità di predizione, man mano che acquisiscono nuove informazioni. D'altra parte, gli applicativi machine-learning-based richiedono un numero considerevole di risorse spazio-temporali, soprattutto nella fase di training.

Le cosiddette *reti Bayesiane* modellizzano le relazioni probabilistiche tra le metriche d'interesse. Esse sono usate per rilevare anomalie, riuscendo a codificare le interdipendenze tra le variabili in gioco e a predire nuovi eventi. È stato dimostrato che i risultati ottenibili con reti bayesiane son comparabili con i sistemi basati su treshold, utilizzando però un numero di risorse considerevolmente maggiore; inoltre la capacità di rilevazione è dipendente dal dominio d'interesse a cui sono applicate.

Anche *le catene di markov* vengono usate nella rilevazione di anomalie; una catena di markov è composta da un insieme di stati, interconnessi da alcune probabilità di transizione, che determinano la topologia e la capacità del modello. Durante la fase di training vengono apprese le probabilità di transazione (da uno stato della catena ad un altro) in base al comportamento del sistema da monitorare. La rilevazione di anomalie viene effettuata confrontando lo score della sequenza di eventi osservata, con un valore di threshold fissato.

Le *reti neurali artificiali*, modellando il concetto di neuroni e sinapsi presenti nel cervello umano, sono molto utilizzate nell'ambito della rilevazione di intrusione, grazie alla loro flessibilità e capacità di adattamento in base alle nuove informazioni che ricevono. Esse sono capaci di apprendere, grazie ad esempi etichettati o meno, il comportamento corretto, e talvolta anche quello anomalo, di un determinato sistema. Uno degli elementi che accomuna le varie reti neurali esistenti, è il fatto che la decisione presa non risulta umanamente interpretabile, in quanto l'apprendimento consiste nel trovare la configurazione ottimale di alcuni parametri in modo da minimizzare una certa funzione obiettivo.

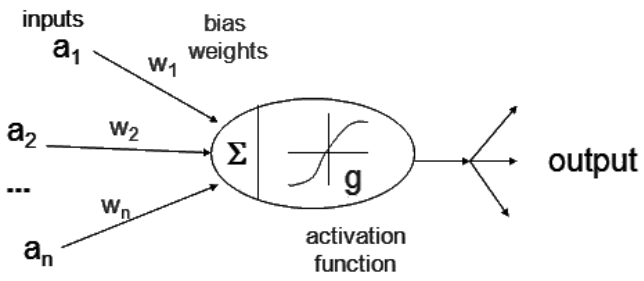


Figura 3: Neurone di una rete neurale

Anche la tecnica del clustering è usata per rilevare anomalie in un insieme di dati: essa opera raggruppando i dati osservati in alcuni agglomerati (cluster) in base alla loro reciproca somiglianza. La tecnica più comune è quella di scegliere un rappresentante di ogni cluster (centroide) e man mano che vengono osservati nuovi dati, si inseriscono nel cluster rappresentato da centroide più vicino; i punti che sono "troppo" diversi dai centroidi individuati, sono considerati anomalie. Un esempio di algoritmo di clustering è il K-NN (k-nearest neighbors), che opera partizionando i dati in base ai k punti più vicini, solitamente utilizzando la distanza euclidea.

Altre tecniche note impiegano la logica fuzzy e gli algoritmi genetici.

2.3. Anomaly-detection su Serie Temporali

Vari modelli appartenenti a questa categoria sono stati proposti nella letteratura statistica [10]: autoregressive moving average (ARMA), autoregressive integrated moving average (ARIMA), vector autoregression (VARMA), Cumulative SUM Statistics (CUSUM), media mobile esponenziale, holt-winters (HW) [11], reti neurali artificiali LSTM [12].

Si possono individuare due tipi di tecniche principali per la rilevazione di anomalie su serie temporali: analisi su un insieme di serie temporali e analisi su una singola serie temporale.
Nel primo caso, dato un database di serie temporali vogliamo individuare tutte le serie temporali anomale. L'assunzione di base è che la maggior parte delle serie temporali nel database sono normali, mentre pochissime sono anomale. L'idea di base è quella di costruire (tramite processo di apprendimento) un modello che rappresenti il comportamento riassuntivo di tutte le serie temporali, che vengono poi confrontate singolarmente con il modello (supervisionato o non supervisionato) creato.

Un possibile approccio non supervisionato, detto discriminativo, si basa sulla definizione di una funzione di somiglianza (come la lunghezza normalizzata dell'LCS (longest common subsequence)) che consente di confrontare sequenze di valori delle serie temporali. Una volta che tale funzione viene definita si procede al processo di clustering a seguito del quale vengono assegnati degli score alle sequenze delle serie temporali in base alla distanza dal relativo centroide.

Un altro approccio non supervisionato, chiamato parametrico, consiste nella costruzione di un modello che sintetizza il comportamento delle serie temporali. Una sequenza di valori viene detta anomala se la sua probabilità di generazione è molto bassa in base al modello generato. Modelli che appartengono a tale categoria sono gli automi a stati finiti, modelli di Markov e modelli di Markov nascosti.

Se i dati delle serie temporali risultano etichettati (ovvero sappiamo se una sotto-sequenza di valori è anomala o no) possono essere usati vari metodi per la rilevazione di anomalie come le reti neurali, support vector machines (SVMs), alberi di decisione, reti di Elman etc.

Altre tecniche che possono essere applicate ad un database di serie temporali consistono nel rilevare finestre temporali anomale, o sotto-sequenze anormali rispetto ad una data serie temporale di test, opportunamente costruita.

Data una singola serie temporale, si può ricercare un singolo particolare elemento come anomalia o individuare un'intera sottosequenza anomala.
Per individuare punti anomali è possibile utilizzare modelli predittivi, approcci basati sulla somiglianza ad un determinato profilo, l'individuazione dei cosiddetti devianti e la tecnica del clustering.

Utilizzando modelli predittivi, il punteggio anomalo per un punto della serie temporale viene calcolato considerando la distanza dal valore predetto dal modello di predizione che sintetizza il comportamento della serie temporale che si vuole analizzare.

L'approccio basato sulla somiglianza ad un determinato profilo consiste nel mantenere le caratteristiche normali della serie temporale (profilo), e confrontare ogni nuovo punto con il profilo costruito per rilevare anomalie.

I devianti sono punto anomali da un punto di vista della minima lunghezza necessaria per descrivere serie temporale: se la rimozione di un punto *P* da una sequenza temporale permette di descrivere la sequenza stessa in modo significantemente succinto, allora il punto *P* è anomalo.

Per rilevare sottosequenze anomale, si procede nel calcolare la distanza (per esempio si può usare la classica distanza euclidea) tra le varie sotto sequenza che non si sovrappongono della serie temporale in analisi.

Alcune soluzioni commerciali che effettuano analisi su serie temporali, sono **Datadog** [13] e **SignalFx** [14]; tali software rilevano anomalie su serie temporali utilizzando approcci a soglia fissa, statistici e basati sul machine-learning, in base alle esigenze dell'utente.

3. Obiettivi

Lo scopo di questa Tesi è la realizzazione di un sistema automatico di analisi di serie temporali, capace di rilevare alcune problematiche di rete, efficiente in spazio e in tempo, utilizzabile all'interno di un'intera rete per monitorare gli host che ne fanno parte. Inoltre, esso deve prevedere dei meccanismi per segnalare ed eventualmente mitigare le anomalie rilevate.

Si vogliono rilevare le anomalie, grazie all'apprendimento del comportamento passato delle metriche a medio-lungo termine, oppure tramite il confronto delle metriche a breve termine con un profilo comportamentale che modella il concetto di normalità.

Il sistema proposto si colloca quindi nella famiglia degli anomaly-based IPS (intrusion prevention system); in particolare esso combina la semplicità e l'efficienza dell'approccio statistico (usando preliminarmente alcune soglie fisse note) nel breve termine, con la potenza e la precisione del machine learning nel medio-lungo periodo: si è realizzato un ibrido tra le famiglie statistical-based e machine learning-based discusse precedentemente nello stato dell'arte ed in particolare si intende rilevare i punti anomali di ogni serie temporale (che viene analizzata in modo indipendente dalle altre) utilizzando modelli predittivi e tecniche basate sulla somiglianza a profili considerati normali.

Per quanto riguarda la capacità di rilevazione, viene adottato un comportamento che intende minimizzare il più possibile la quantità di falsi positivi, ovvero le rilevazioni di anomalie quando queste non sono realmente presenti: vogliamo quindi massimizzare la precisione *P*, ovvero il seguente rapporto [15]:

$$P = \frac{TP}{TP + FP}$$

dove TP rappresenta il numero dei veri positivi (rilevazione di anomalie esistenti) e FP il numero dei falsi positivi (rilevazione di anomalie non esistenti).

Altre misure statistiche della performance di rilevazione, che verranno prese severamente in esame, sono la sensitività e la specificità:

$$SENSITIVITY = \frac{TP}{TP + FN}$$

$$SPECIFICITY = \frac{TN}{TN + FP}$$

dove FN è il numero dei falsi negativi e TN è il numero dei veri negativi.

4. Motivazioni

Il lavoro di Tesi intende estendere la classe degli anomaly-based IDS, costruendo un sistema innovativo capace di monitorare il comportamento di un insieme di host in modo completo ed efficiente.

L'enfasi della rilevazione è posta sull'analisi avanzata di serie temporali, soprattutto nel medio-lungo termine dove i concetti di trend e di stagionalità risultano di fondamentale importanza.
Il fattore tempo è fondamentale per capire profondamente il comportamento di una metrica di rete, e quindi per rilevarne eventuali anomalie; per esempio, è necessario modellare esplicitamente il fatto che alcune metriche di rete presentino dei picchi fisiologici (i quali possono eventualmente ripetersi periodicamente), che non sono da intendersi come anomalie: se si considera un host situato in un ambiente lavorativo, è normale che esso non faccia alcun traffico nel weekend o nell'orario di chiusura, e presenti quindi un improvviso aumento di traffico non appena un dipendente inizi nuovamente a lavorare.

I valori relativi ad alcune coppie di metriche, sono solitamente proporzionali tra loro, mantenendo un rapporto più o meno costante nel tempo: in tal caso è necessario innanzitutto effettuare un controllo preliminare sul valore del rapporto stesso, e successivamente applicare degli algoritmi per verificare l'eventuale presenza di picchi nella serie temporale rappresentante il comportamento del rapporto considerato.

Riassumendo, le motivazioni principali per le quali è stato effettuato questo lavoro di Tesi sono le seguenti:

- Molte tecniche per rilevare anomalie (per esempio, quelle basate su rete neurali) risultano relativamente costose e complesse per essere applicate ad un numero consistente di host, soprattutto per le analisi nel breve termine; si vuole quindi creare un applicativo capace di eseguire analisi su un gran numero di host, e che quindi deve risultare efficiente in spazio e tempo;
- Nella letteratura, raramente il tempo assume un ruolo centrale nella rilevazione di anomalie; invece in questo lavoro di Tesi, esso assume un ruolo fondamentale poiché ci consente di caratterizzare con molta

precisione una metrica di rete, e quindi di individuare sue eventuali anomalie.

- Le tecniche esistenti di rilevazioni di anomalie su serie temporali molto spesso non eseguono nessun controllo o ragionamento preliminare sui dati da analizzare; in questo lavoro di Tesi, prima di applicare le tecniche di rilevazione di anomalie, si vuole eseguire un processo di ripulitura dei dati e/o di correlazione tra metriche, in modo da poter eseguire analisi più intelligenti e precise (come è stato fatto per le metriche a breve termine).

5. Contributo Originale

Il contributo originale di questo lavoro di Tesi alla letteratura corrente, consiste nella realizzazione di una nuova metodologia per la rilevazione di anomalie su serie temporali, che si vuol distinguire dalle soluzioni già esistenti per le tecniche e gli algoritmi impiegati.

In particolare, per effettuare le rilevazioni a breve termine è stato considerato un efficiente indicatore, impiegato per effettuare analisi sui mercati finanziari; il suo utilizzo per la rilevazione di anomalie su serie temporali, non appare in letteratura.

Inoltre, per le analisi a medio-lungo termine è stato considerato un nuovo modello di predizione su serie temporali, non ancora menzionato in nessuna ricerca di rilievo riguardante la rilevazione di anomalie su serie temporali.

Grazie al software di monitoraggio di rete ntop, il sistema realizzato dispone di una visibilità sulle metriche di rete ad ampio raggio; ciò ha permesso la selezione e la correlazione di metriche in modo innovativo, rispetto allo stato dell'arte corrente.

6. Architettura della soluzione

In questa sezione vengono discusse, le scelte progettuali adottate in questo lavoro di Tesi.
In particolare, viene spiegato in modo dettagliato cosa si è deciso di monitorare, e con quali metodologie vengono effettuate le varie analisi di rete e l'eventuale mitigazione delle minacce rilevate.
Viene inoltre illustrata l'architettura del sistema di rilevazione di anomalie proposto.

6.1. Metriche

L'analisi delle metriche individuate per questo lavoro di Tesi, consente di individuare alcuni problemi di rete comunamente riscontrati nelle reti odierne; dove necessario, sono state messe in correlazione coppie di metriche in modo da poter effettuare le analisi in modo più preciso e significativo.

Sono state individuate due macro-famiglie di metriche da analizzare:

- *Metriche a breve termine*: esse sono essenzialmente composte da coppie di metriche correlate tra loro: data la coppia di metriche (x, y), viene considerato il rapporto x/y; ipotizziamo in modo ragionevole, che numeratore e denumeratore siano "più o meno" proporzionali tra loro, quindi importanti disproporzionalità verranno considerate anomale; inoltre è possibile controllare la presenza di anomalie, tramite delle soglie fisse sul valore del rapporto;
- *Metriche a medio-lungo termine*: la loro caratteristica principale è data dal fatto che esse presentano un trend e delle stagionalità dipendenti dal tempo; Queste metriche verranno analizzate da un algoritmo che sfrutta in modo cruciale il loro comportamento temporale passato.

Precisiamo ulteriormente che entrambe le famiglie di metriche dipendono dal tempo di osservazione, in altre parole data la metrica (o coppia di metriche) x, siamo

interessati alla funzione $f_x(t)$ che ne esprime il comportamento in funzione del parametro t (tempo).

Tutte le metriche (che verranno discusse successivamente) sono state scelte in seguito ad una lunga ed attenta analisi delle loro proprietà: tra le innumerevoli metriche esistenti, se ne sono scelte alcune tra le più rilevanti ai fini delle analisi che si vogliono effettuare; infatti le metriche considerate appaiono comunamente nel traffico odierno e molto spesso sono affette da anomalie; inoltre, possedendo le caratteristiche delle sopra citate famiglie di metriche, è relativamente semplice individuare un algoritmo capace rilevare anomalie, basandosi sul comportamento atteso della metrica.

I valori delle metriche sono stati ottenuti utilizzando il software di monitoraggio di rete ntop. Alcune delle metriche (o meglio, i contatori relative ad esse e le corrispondenti serie temporali) erano già presenti in ntop, altre sono state implementate appositamente per questo lavoro di Tesi; infatti, sono state create le serie temporali relative a flussi host unreachable, pacchetti ARP, statistiche e pacchetti TCP, pacchetti ICMP echo, e pacchetti DNS.

6.1.1. Metriche a Breve Termine

Le metriche a breve termine vengono analizzate dal sistema ogni cinque minuti, alla ricerca di picchi o valori anomali; inoltre è necessario un periodo di almeno un'ora per inizializzare il comportamento delle metriche di rete considerate, al fine della rilevazione di eventuali pendenze sospette. Come detto in precedenza, vengono analizzate coppie di metriche, opportunamente correlate.

In particolare, data la coppia di contatori (x, y) *relativi alle metriche* $(M_{x,}\, M_y)$ *da analizzare, consideriamo il loro rapporto e le variazioni di esso:*

$$\frac{\Delta x}{\Delta y} = \frac{x_{t_f} - x_{t_i}}{y_{t_f} - y_{t_i}} = r_{t_i}, \qquad \Delta r = r_{t_f} - r_{t_i}$$

Le metriche a breve termine (o meglio, le coppie di metriche) che vengono prese in esame sono le seguenti:

- **Rapporto tra risposte DNS ricevute e query DNS inviate**: poiché in situazioni normali, ad ogni query del protocollo di risoluzione degli indirizzi

corrisponde una e una sola risposta, idealmente il loro rapporto deve essere uguale ad uno, o leggermente minore di uno a causa della non affidabilità del protocollo UDP; valori maggiori indicano sicuramente la presenza di un problema che va investigato: potrebbe essere dovuto ad una cattiva configurazione di rete, o nel peggiore dei casi ad attacchi di tipo flooding [16].

- **Rapporto tra risposte DNS con errore ricevute e risposte DNS totali ricevute**: ricevere occasionalmente riposte DNS con codice d'errore può essere del tutto fisiologico, ma è necessario confrontarle numericamente con le risposte DNS totale ricevute: un'elevata percentuale di errori rispetto alle risposte corrette e/o incrementi improvvisi del valore del rapporto, possono indicare un'errata impostazione di rete o problematiche relative al name server o al resolver, che devono essere gestite in tempi brevi.
- **Rapporto tra i bytes del protocollo DNS inviati e i pacchetti DNS inviati**: tale rapporto esprime la dimensione media dei pacchetti DNS inviati (query e risposte inviate); a causa della non affidabilità del protocollo UDP, per evitare frammentazioni del pacchetto in transito, si mantiene la dimensione del pacchetto UDP (compreso il payload) minore o uguale a 576 bytes, che è il minimo valore del MTU (maximum transmission unit). Quindi, se la dimensione media dei pacchetti DNS è maggiore di 576 bytes o si rileva un aumento consistente del valore del rapporto, potrebbe trattarsi di "data exfiltration" [17] o di un attacco DNS flooding di tipo amplificazione.
- **Rapporto tra i bytes del protocollo DNS ricevuti e i pacchetti DNS ricevuti**: valgono tutte le osservazioni fatte nel punto precedente, con la differenza che in questo caso, in caso di rilevazione anomala, il problema potrebbe essere l'infiltrazione di dati o l'essere vittima di un attacco DNS flooding.
- **Rapporto tra risposte ICMP echo e richieste ICMP echo:** I pacchetti ICMP di tipo echo vengono comunamente inviati per verificare lo stato d'attività di un host connesso alla rete; idealmente il rapporto risposte e richieste è uguale a uno, ma potrebbe anche essere minore di uno a causa di eventuali richieste perse o host inattivi; valori maggiori indicano certamente

un problema, e nel caso peggiore potrebbe essere un sintomo di un attacco di tipo ICMP flooding;

- **Rapporto tra flussi ICMP port unreachable come client e flussi totali come server:** Un flusso in uscita di tipo port unreachable viene generato da un host quando esso viene contattato su una porta sulla quale non è presente alcun servizio attivo; inviare occasionalmente pacchetti ICMP port unreachable è del tutto normale, ma è necessario confrontarli numericamente con i flussi totali ricevuti: se il loro rapporto supera il 50%, oppure aumenta in maniera importante, allora è necessario investigare la situazione in quanto l'host analizzato potrebbe essere sotto attacco port scan, tramite protocollo UDP [18]; infatti, quando il rapporto supera il valore 0.50 si ha una situazione in cui più di un host su due che contatta il sistema analizzato, trova il servizio desiderato non attivo; questo è proprio il classico scenario di un attacco di tipo port scan, in cui l'host malevolo va alla ricerca di servizi vulnerabili sulla macchina scelta come vittima; ovviamente la rilevazione di un'anomalia non indica con la massima certezza la presenza di un port scan; per esempio, potrebbero esserci problemi con un servizio noto installato sull'host, che non risulta raggiungibile per una qualche ragione che va investigata.
- **Rapporto tra flussi ICMP port unreachable come server e flussi totali come client:** sono valide le stesse osservazioni fatte nel punto precedente; l'host anomalo potrebbe aver eseguito un port scan, oppure potrebbe aver cercato di accedere ad un servizio remoto non più raggiungibile. Si noti che in questo caso, si stanno considerando i flussi port unreachable ricevuti.
- **Rapporto tra flussi ICMP host unreachable come client e Flussi totali come server:** Un flusso host unreachable in uscita viene generato quando un host (solitamente un router) non riesce ad instradare un pacchetto, in base all'indirizzo IP di destinazione; un rapporto il cui valore supera il valore 0.50 o presenta un rapido incremento, è da considerarsi anomalo; infatti un valore del rapporto maggiore del 50%, indica che circa un host su due che ha inizializzato una connessione con il sistema analizzato, ha inviato pacchetti con indirizzo IP di destinazione non inoltrabile, e ciò è chiaramente un'anomalia; la rilevazione di problematiche su tale metrica,

possono indicare la presenza di attività malevola di un worm, il quale nelle fasi iniziali provvede a generare (spesso in modo randomico) degli indirizzi IP, alla ricerca di host con vulnerabilità da sfruttare. Spesso non esiste una rotta per gli indirizzi IP generati, e quando i worm provano a contattarli, qualche router, che non riesce ad inoltrare l'indirizzo, invia un pacchetto ICMP di tipo host unreachable [19]. L'anomalia rilevata non è chiaramente imputabile alla sola attività di un worm, ma potrebbe essere causata, per esempio, da eventuali problemi dei protocolli di routing.

- **Rapporto tra flussi ICMP host unreachable come server e flussi totali come client:** valgono le analisi espresse nel punto precedente, precisando che se un host presenta anomalie in questa metrica, allora esso potrebbe essere stato infettato da un worm e quindi potrebbe aver iniziato la fase di ricerca degli indirizzi IP, poiché i messaggi ICMP di tipo host unreachable sono diretti verso il sistema analizzato.
- **Rapporto tra richieste ARP inviate e risposte ARP ricevute**: idealmente ad ogni richiesta ARP corrisponde una solo risposta ARP, quindi il valore del rapporto dovrebbe essere pari ad uno; se il valore del rapporto è maggiore di uno, significa che alcune richieste ARP non hanno ricevuto risposte, e ciò risulta anomalo; tale problematica potrebbe essere dovuta ad un'errata configurazione di rete o all'esecuzione di un network discovery. Infatti, quando viene eseguito un network discovery, vengono inviate richieste ARP in cui si richiede di risolvere i vari IP appartenenti al blocco di indirizzi locali; se un indirizzo IP non è presente nella rete locale, allora la richiesta ARP non riceverà risposta, e quindi il rapport tra richieste e risposte risulterò maggiore di uno.
- **Rapporto tra pacchetti TCP ritrasmessi, persi, out of order e pacchetti TCP totali: i** pacchetti TCP persi, ritrasmessi e out of order, non sono rari durante una comunicazione TCP; ma se il valore del rapporto è troppo alto o presenta incrementi improvvisi, allora potrebbero esserci alcuni problemi di rete, come la congestione del traffico.
- **Rapporto tra flussi sospetti e flussi totali:** un flusso è considerato sospetto quando presenta le seguenti caratteristiche:
    - Connessione TCP lenta

- Scambio lento di dati (possibile Slowloris)
- Goodput basso
- Lunga durata
- Volume del traffico elevato
- Host client e/o server presenti in una blacklist
- Problemi con il protocollo TCP

se tali flussi occorrono occasionalmente, non devono destare particolari sospetti; se invece se ne hanno troppi rispetto al totale dei flussi, o il valore del rapporto presenta un aumento importante, allora ciò potrebbe indicare una possibile attività malevola in atto.

Si noti che alcune delle metriche sopra discusse sono state classificate anomale quando il loro valore aumenta in modo improvviso: ciò è dovuto alla ragionevole supposizione precedentemente fatta, circa la quasi proporzionalità tra numeratore e denumeratore delle metriche considerate.

6.1.2. Metriche a Medio-Lungo Termine

Le metriche a medio-lunghe termine, a differenza delle metriche a breve termine, sono costituite da una singola componente (non coppie di componenti) e non sono quindi presenti correlazioni tra metriche, seppure possono essere corredate da ulteriori controlli che coinvolgono altre metriche, come sarà spiegato in seguito.

Le metriche vengono controllate dal sistema ogni ora, alla ricerca di comportamenti inattesi; inoltre i valori delle metriche devono preventivamente essere osservati per un periodo minimo di tre settimane, per poter poi rilevare eventuali anomalie, scelta che verrà approfondita nella sezione riguardante la validazione. In particolare, le metriche scelte possiedono delle caratteristiche ben precise, ovvero presentano un trend e delle stagionalità in funzione del tempo: per esempio, i valori associati ad una certa metrica possono crescere, decrescere o oscillare in un certo periodo temporale, ed il comportamento generale può ripetersi nel tempo, individuando quindi stagionalità ed eventuali sub-stagionalità della metrica. Ovviamente le caratteristiche di tali metriche, ci guidano nella scelta di un algoritmo che sfrutti le proprietà sopra descritte.

Le metriche a medio-lungo termine, analizzate in questa Tesi, sono le seguenti:

- **Bytes inviati**: i bytes che sono stati inviati, ovvero il traffico in uscita di un host;
- **Bytes ricevuti**: i bytes che sono stati ricevuti, ovvero il traffico in entrata di un host;
- **Flussi come client**: i flussi in uscita di un host, ovvero le comunicazioni che esso ha inizializzato;
- **Flussi come server**: i flussi in entrata di un host, ovvero le richieste di comunicazione che esso ha ricevuto.

Sulle seguenti metriche possono essere fatte le seguenti osservazioni: se ci si discosta troppo dal consueto trend e stagionalità della metrica sotto esame, potrebbero essere presenti problematiche di rete più o meno gravi: l'host analizzato ha infatti presentato un cambiamento di comportamento importante, per una qualche ragione che va investigata dall'amministratore di rete; per esempio l'host analizzato potrebbe essere diventato un flooder (esecutore di un attacco di tipo inondazione), o essere vittima di un attacco flooding.

Come accennato inizialmente, è possibile corredare la rilevazione di anomalie, con ulteriori controlli che considerano la presenza dei seguenti protocolli o eventi rilevati da ntop:

- **Protocolli sconosciuti**: protocolli la cui natura non è nota, ovvero non corrisponde a nessun protocollo noto a NDPI (Network Deep Packet Inspection), una libreria di ispezione approfondita di pacchetti, utilizzata da ntop;
- **Protocolli di accesso remoto:** molti di questi protocolli possono essere presi di mira da malintenzionati, che ne sfruttano le vulnerabilità per prendere il controllo di host remoti;
- **Comunicazione con host catalogati come malware:** si tratta di host inseriti in delle apposite blacklist, catalogati come malware;
- **Comunicazione con host catalogati come responsabili di attività di mining:** si tratta di host presenti in delle blacklist, catalogati come responsabili di attività di mining.

Quindi, nel caso in cui venga rilevata un'anomalia sulle metriche a medio-lungo termine (bytes inviati e ricevuti, flussi come server e come client), si può eventualmente controllare se nello stesso periodo temporale in cui si è verificata la problematica, ci sia stato o meno traffico in entrata o in uscita (in base alla metrica anomala) relativo alle categorie di protocolli e eventi elencati sopra: in questo modo la precisione di rilevazione migliora notevolmente, riducendo il numero di eventuali falsi positivi, dovuti ad un cambio di comportamento del tutto fisiologico dell'host sotto analisi.

6.2. Rilevazione delle Anomalie

Rilevare un'anomalia significa individuare eventi o valori che per qualche loro caratteristica non possono essere considerati normali. Ai fini delle analisi che si vogliono effettuare è essenziale, quindi, definire il concetto di normalità: senza definire ciò che viene considerato normale, è impossibile definire ciò che può essere considerato anormale. Questa semplice considerazione è il punto di partenza per tutti i ragionamenti effettuati successivamente.

Nella fattispecie, ci si preoccupa di definire quali sono i parametri di normalità delle metriche di rete che si vogliono analizzare, e per far ciò è importante capire la natura delle metriche stesse.

Alcune volte è realtivamente facile individuare anomalie di alcune metriche, soprattutto quando esse non sono analizzate come entità a sé stanti, ma vengono correlate ad altre metriche; per esempio, si considerino le query DNS inviate e le risposte DNS ricevute: se il valore del rapporto tra risposte e richieste risultasse maggiore di uno, ci sarebbe certamente un qualche problema; in tal caso è immediato utilizzare banali tecniche a soglia fissa, per rilevare l'anomalia.

In altri casi è certamente più complesso, e le metodologie utilizzate sono maggiormente soggette al problema dei falsi positivi; per esempio, si considerino le risposte DNS di tipo errore e le risposte DNS totali; si possono preliminarmente applicare delle tecniche a soglia fissa, e se queste danno esito negativo (ovvero assenza di anomalie), utilizzare delle tecniche più complesse che rilevino incrementi importanti nel rapporto delle metriche sopra considerate.

Altre volte non è conveniente o non è possibile correlare più metriche al fine della rilevazione di anomalie di rete, come nel caso dei bytes inviati ovvero il traffico totale in uscita; una possibile idea potrebbe essere quella di correlare tale metrica con i bytes ricevuti e assumere una certa proporzionalità tra loro, ma ciò nello scenario moderno è un'assunzione poco veritiera, in quanto in base al servizio richiesto il rapporto tra le due metriche può variare in maniera abbastanza netta, senza essere sintomo di attività malevola.
In tal caso è invece necessario apprendere il comportamento della singola metrica, e se in futuro il suo comportamento dovesse mutare profondamente allora potrebbe esserci un'anomalia.

In estrema sintesi, il comportamento normale delle metriche può essere definito da:

- Non superamento dei valori soglia
- Comportamento più o meno costante nel tempo
- Comportamento futuro coerente con quello passato

In particolare, i primi due punti definiscono il comportamento normale delle metriche a breve termine, mentre l'ultimo punto definisce il comportamento atteso delle metriche a medio-lungo termine.

6.2.1. Treshold

Tale tecnica si applica unicamente alle metriche a breve termine, per i motivi discussi precedentemente.

I sistemi a soglia (treshold) vengono utilizzati per individuare con ottima precisione anomalie nelle metriche di rete.
Sono state individuate tre tipi di soglia, che si applicano a diverse tipologie di metriche a breve termine:

- **Soglia fissata al valore di 1** per le seguenti metriche: rapporto risposte e query DNS, rapporto richieste e risposte ICMP echo e rapporto richieste e risposte ARP; valori maggiori ad uno di tali rapporti, indicano la presenza di un problema, come spiegato nella sezione 6.1.1.
- **Soglia fissata al valore di 576 (bytes),** per le seguenti metriche: rapporto bytes DNS inviati e pacchetti DNS inviati e rapporto bytes DNS ricevuti e

pacchetti DNS ricevuti; la soglia quindi limita la dimensione media dei pacchetti DNS inviati e ricevuti, individuando quindi eventuali fughe o infiltrazioni di dati usando il protocollo DNS in modo improprio. Siccome il protocollo DNS si poggia sul protocollo di livello trasporto UDP, per evitare la frammentazione del pacchetto, solitamente si mantiene la dimensione del payload DNS minore o uguale a 512 bytes in modo da rientrare nel valore minimo assunto dall' MTU (576 bytes), ed evitare quindi la frammentazione del pacchetto.

- **Soglia fissata al valore di 0.50 per le altre metriche a breve termine**: se il valore di un rapporto risulta maggiore di 0.50, allora si ha chiaramente un'anomalia che necessita di ulteriori controlli; per esempio, se il rapporto tra flussi sospetti e flussi totali supera il valore imposto dalla soglia, si è in una condizione anomala; infatti il 50% dei flussi totali risulterebbe sospetto. Analoghi ragionamenti valgono per le tutte altre metriche a breve termine, non menzionate nei due punti precedenti.

6.2.2. RSI

I valori di treshold riescono a identificare in modo abbastanza preciso un'anomalia; per alcune metriche, quando non viene superato il valore di soglia, ci si trova in una sorta di zona grigia in cui è difficile stabilire se la metrica sia o meno anomala. In tal caso è necessario controllare la presenza di incrementi ripidi del valore della metrica in esame; chiaramente, tale tecnica viene applicata a quasi tutte le metriche a breve termine, che presentano, per ipotesi, un comportamento più o meno costante nel tempo: le metriche che vengono controllate mediante la soglia fissa con valore pari a 1, non vengono analizzate dall'RSI poiché l'eventuale rilevazione di picchi non si configura come anomalia; per esempio un incremento del rapporto tra pacchetti ICMP echo ricevuti ed inviati in un certo periodo di tempo t, potrebbe essere dovuto ad un rapporto registrato al tempo $t-1$ in cui si è effettuato il ping su un host non attivo: siccome al tempo $t-1$ il rapporto tende a 0, l'invio di pacchetti ICMP echo di tipo richiesta al tempo t ad un host attivo, determinerà un aumento del valore del rapporto e quindi il sollevamento di un falso allarme.

Per rilevare picchi nelle serie temporali, viene utilizzato un indicatore statistico chiamato **RSI** (Relative Strength Index) [20]; esso è uno degli oscillatori più popolari dell'analisi tecnica, ovvero dello studio dei prezzi dei mercati finanziari.
Esso è chiamato oscillatore poiché varia tra due valori ovvero tra 0 e 100; siccome in ambito finanziario è usato per valutare la velocità del movimento dei prezzi, sfruttiamo questa sua caratteristica per analizzare la velocità con cui variano i valori associati ad una metrica di rete;

Si noti la capacità dell'RSI nel rilevare la presenza di picchi nel grafico rappresentante il prezzo di un titolo azionario, in figura 4:

Figura 4: Calcolo dell'RSI

È possibile fissare un valore limite entro il quale il valore dell'RSI calcolato deve sottostare, per far si che la metrica che stiamo analizzando non presenti picchi troppo ripidi; tale caratteristica, fa dell'RSI il candidato ideale per la rilevazione di incrementi anomali su serie temporali.
Nei mercati finanziari si considerano normali, valori dell'RSI che oscillano tra 30 e 70 (nella figura 4, tali limiti sono rappresentati dalle due barre orizzontali che attraversano la curva in blu).
Per le analisi sulle metriche a breve termine, non viene preso in considerazione alcun limite inferiore al valore dell'RSI, in quanto, per come sono state definite le metriche, un trend negativo non indica la presenza di anomalie: la metrica si vuole tenere sotto controllo, è sempre il numeratore del rapporto; quindi all'aumentare del valore del rapporto, si ha un aumento percentuale del numeratore rispetto al denumeratore, e quindi una possibile anomalia.

Come detto precedentemente, è necessario definire un limite superiore per rilevare le metriche che crescono troppo velocemente: è stato deciso fissare un valore di threshold pari a 80, in quanto ciò garantisce un buon trade-off tra falsi positivi e falsi negativi; quest'ultimo aspetto verrà approfondito nella sezione riguardante la validazione dei modelli.
Un ulteriore parametro da scegliere per calcolare l'RSI è il numero dei periodi, ovvero il numero di variazioni di rapporti consecutivi temporalmente: se si sceglie di analizzare x periodi, allora ci serviranno $x + 1$ punti (ovvero valori di rapporti tra metriche) per poter calcolare le x variazioni richieste. In questo lavoro di Tesi si considerano 50 periodi, per ragioni che saranno spiegate nella sezione di validazione.

La formula per il calcolo dell'RSI è la seguente:

$$RSI = 100 * U/(U + D)$$

Dove U è la media delle differenze positive tra punti consecutivi nel periodo considerato, e D è la media delle differenze negative tra punti consecutivi nel periodo considerato. Sia N il numero di periodi, ed x un array di dimensione $N + 1$ contenente i valori della metrica da analizzare, temporalmente consecutivi, allora in formule U e D si possono esprimere come:

$$U = \frac{\sum_{t=2}^{N+1} \max(0,\ x_t\ -\ x_{t-1})}{N}$$

$$D = \frac{\sum_{t=2}^{N+1} |\min(0,\ x_t\ -\ x_{t-1})|}{N}$$

Si noti, che per ogni metrica a breve termine di ogni host analizzato è necessario mantenere in memoria $N + 1$ valori, in modo che all'arrivo del prossimo punto venga effettuato uno shift di una posizione verso sinistra degli $N + 1$ punti precedenti e venga inserito il nuovo valore per ricalcolare il valore dell'RSI. Si può gestire tale inconveniente calcolando la formula precedente solo per i primi N + 1 punti, e per i successivi valori si procede approssimando il valore RSI esatto, calcolando U e D con la seguente formula [21]:

$$U = (U_{old} * (N - 1) + \max(0, x_{new} - x_{last}))/N$$

$$D = (D_{old} * (N - 1) + |\min(0, x_{new} - x_{last})|)/N$$

Si noti che tale approssimazione è simile a quella effettuata nel calcolo dalla media mobile esponenziale. In tale lavoro di Tesi si usa quindi una variante dell'RSI, che consente un importante risparmio di memoria a discapito di una minima perdita di precisione.

Il calcolo dell'RSI è ovviamente molto efficiente in tempo, in quanto sono necessarie poche operazione aritmetiche di base: nel calcolo del primo valore dell'RSI sono necessarie N addizioni e due operazioni moltiplicative, mentre successivamente, per calcolare i valori approssimati, bisogna effettuare quattro operazioni moltiplicative e una somma.
Il costo in tempo, è pari al numero di punti necessari per calcolare il primo valore dell'RSI, ovvero $N + 1$.

Per evitare che l'RSI includa nel calcolo, valori palesemente anomali, che andrebbero ad inquinare le future rilevazioni e lo renderebbero inutilizzabile nel rilevare picchi importanti se il comportamento anomalo persiste, si procede ad eseguire il controllo preliminare con la tecnica precedentemente discussa delle soglie fisse: se un valore è palesemente anomalo, questo non verrà usato nel calcolo dell'RSI, per le motivazioni espresse sopra;

Bisogna fare ulteriori restrizioni sul tipo di valori che contribuiranno al calcolo dell'RSI: siccome si vuole rilevare il comportamento di una metrica, vogliamo che i valori registrati individuino precisamente l'andamento della metrica stessa, e ciò si può ottenere considerando i dati raccolti in condizioni sufficienti di volume del traffico di rete; tali condizioni minime di significatività dei valori considerati, dipendono dalla metrica in esame; per esempio per la rilevazione di flussi anomali si sono stabiliti le seguenti soglie di traffico minimo: 2 flussi con comportamento sospetto oppure 12 flussi totali; al di sotto di questi valori si ritiene che il traffico non sia sufficiente a delineare il reale comportamento della metrica in questione; infatti, abbassando o eliminando tali soglie minime di traffico si è rilevato un elevato numero di falsi positivi individuati dall'RSI.

Per la rilevazione di anomalie riguardanti i messaggi ICMP port unreachable e host unreachable, è stata fatta la seguente scelta (valida per entrambe le tecniche discusse): le analisi non vengono effettuate (i punti delle metriche non vengono analizzati) se si riscontra traffico p2p (peer to peer), poiché esso comporta fisiologicamente l'invio e la ricezione di un elevato numero di messaggi ICMP port

unreachable e host unreachable, e quindi si avrebbe una quantità troppo elevato di falsi positivi.

6.2.3. Prophet

La scelta di un algoritmo in grado di analizzare metriche a medio-lungo termine è avvenuta attraverso una lunga e attenta analisi delle varie alternative esistenti; Inizialmente si è considerato il modello ARIMA (Autoregressive Integrated Moving Average) [22]; esso è un modello statistico relativamente semplice e veloce, ma non è in grado di gestire esplicitamente alcuna stagionalità (che è invece una caratteristica delle metriche che si vogliono analizzare) ed i suoi parametri sono difficili da scegliere a priori senza un processo di fitting.

Successivamente si è analizzato il modello HW (Holt-Winters) [23]. L'HW, come L'ARIMA, è un modello statistico semplice ed efficiente in grado di effettuare predizioni sui valori futuri assunti da una metrica. I suoi parametri possono essere scelti mediante tecniche di training; a differenza dell'ARIMA, esso riesce a catturare la stagionalità della metrica; il problema di questo metodo è la possibilità di modellare solo una singola stagionalità, e come detto in precedenza le metriche che si vogliono analizzare presentano stagionalità a periodicità multiple.

Si è quindi cercato qualcosa di più espressivo, in grado di catturare le caratteristiche delle metriche da analizzare: è stata studiata e testata la rete neurale ricorrente LSTM (Long short-term memory) [24]. Essa si è rilevata in grado di carpire le multi-stagionalità presenti nelle metriche, grazie ad un potente meccanismo di mapping dei valori della serie temporale, presente al suo interno. Il suo costo in spazio e in tempo si è rilevato però troppo oneroso per poter essere impiegato in una rete con migliaia di host da monitorare.
Tutte le tecniche illustrate precedentemente consentono di predire nuovi valori della serie temporale analizzata, ma non incorporano nessun meccanismo che consenta di rilevare eventuali anomalie.
Successivamente la ricerca si è concentrata su modelli con un potere di espressività e costo computazionale intermedio tra i modelli statistici ARIMA e HW, e il modello di deep learning rappresentato dalle reti artificiali LSTM: il sistema scelto si chiama Prophet, e le sue peculiarità vengono sotto discusse.

Le metriche a medio-lungo termine vengono quindi analizzate utilizzando Prophet, un sistema messo a punto dagli sviluppatori di Facebook. Esso consente la creazione di un modello di **regressione**, mediante l'utilizzo di parametri umanamente interpretabili e modificabili da esperti del dominio delle serie temporali da trattare [25]. In breve, Prophet consente la predizione dei valori futuri associati ad una certa metrica, mediante l'apprendimento del comportamento passato della metrica stessa.

In particolare, Prophet utilizza un modello decomponibile in tre sottocomponenti principali: trend, stagionalità e festività, combinati in una delle due forme seguenti:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t \ (additivo)$$

$$y(t) = g(t) * s(t) + h(t) + \varepsilon_t \ (moltiplicativo)$$

dove:

- $g(t)$: è la funzione *trend*, che modella cambiamenti non periodici dei valori della serie temporale;
- $s(t)$: è la funzione *stagionalità*, che modella i cambiamenti periodici (giornalieri, settimanali, annuali, ecc.) della serie temporale;
- $h(t)$: è la funzione *festività*, rappresentante l'effetto delle festività che occorrono in alcuni giorni ben precisi; se una festività si ripete nel tempo, Prophet riuscirà a modellare in modo più preciso il comportamento della metrica analizzata nei giorni di festa specificata; altrimenti, se essa non si ripete, essa verrà modellata ma non verrà inclusa nelle predizioni future;
- ε_t: è il termine rappresentante l'errore, derivante da eventuali variazioni della serie temporale non previsti dal modello; per ipotesi essa viene considerata normalmente distribuita.

In Particolare, la funzione *trend*, $g(t)$, è una componente lineare mentre la funzione *stagionalità*, $s(t)$, si basa sulla serie di Fourier e assume la seguente forma:

$$s(t) = \sum_{n=1}^{N} (a_n \cos(2\pi nt/P) + b_n \sin(2\pi nt/P))$$

dove, N sono i termini della serie di Fourier, P è il periodo considerato (7 giorni per periodicità settimanale, 1 giorno per periodicità giornaliera, ecc.) e i coefficienti $a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_n$, sono $2N$ parametri che dovranno essere appresi durante il processo di training, per approssimare la stagionalità della serie temporale da predire.
Si noti, che incrementando il valore dell'iper-parametro N, il modello rischia di andare in overfitting, con l'effetto di avere una limitata capacità di generalizzazione e quindi di predizione di nuovi valori.
Il fitting del modello alla curva rappresentante la serie temporale reale, viene effettuato tramite la metodologia L-BFGS [26].

Come, accennato all'inizio di questa sezione, con Prophet è possibile definire due tipi di modello, quello moltiplicativo e quello additivo; di default Prophet utilizza il modello additivo dove la componente stagionale viene sommata al trend per effettuare la predizione [27]; tale modello non funziona quando la stagionalità cresce con il trend, ed in tal caso è necessario adottare un modello moltiplicativo. Si notino a proposito i grafici delle figure 5 e 6, generati utilizzando stessi dati di training: in figura 5 si mostra il fitting alla curva della serie temporale reale (curva in blu) con un modello additivo, in figura 6 il fitting effettuato con la variante moltiplicativa; il modello additivo non riesce ad approssimare correttamente la curva della funzione "vera" (punti in nero), poiché è evidente che la stagionalità si amplifica, al crescere del trend. In tal caso è necessario scegliere un modello moltiplicativo. Per le metriche a medio-lungo termine da monitorare si è optato per un modello moltiplicativo, poiché si è rilevato più preciso rispetto a quello additivo; tale scelta verrà giustificata nella sezione riguardante la validazione dei modelli (*Sezione 7.1*).

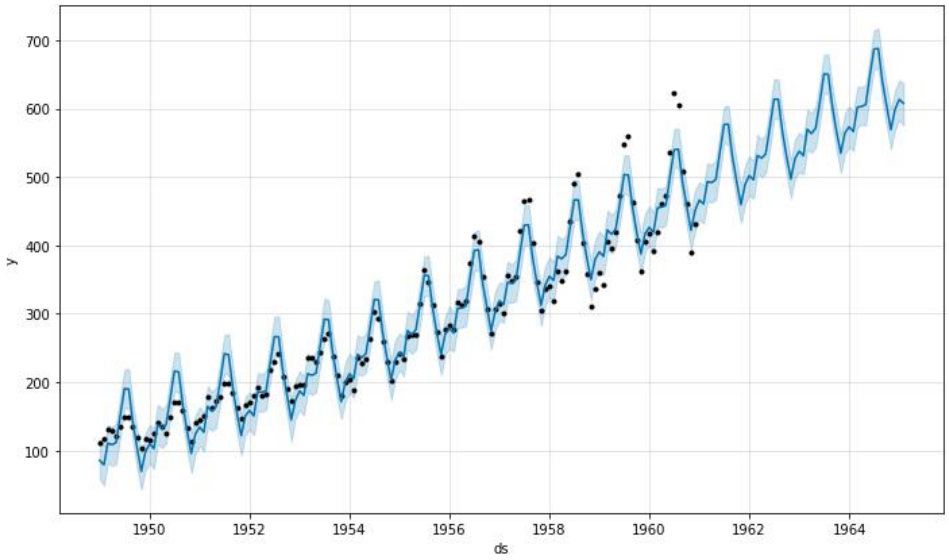


Figura 5: Modello additivo

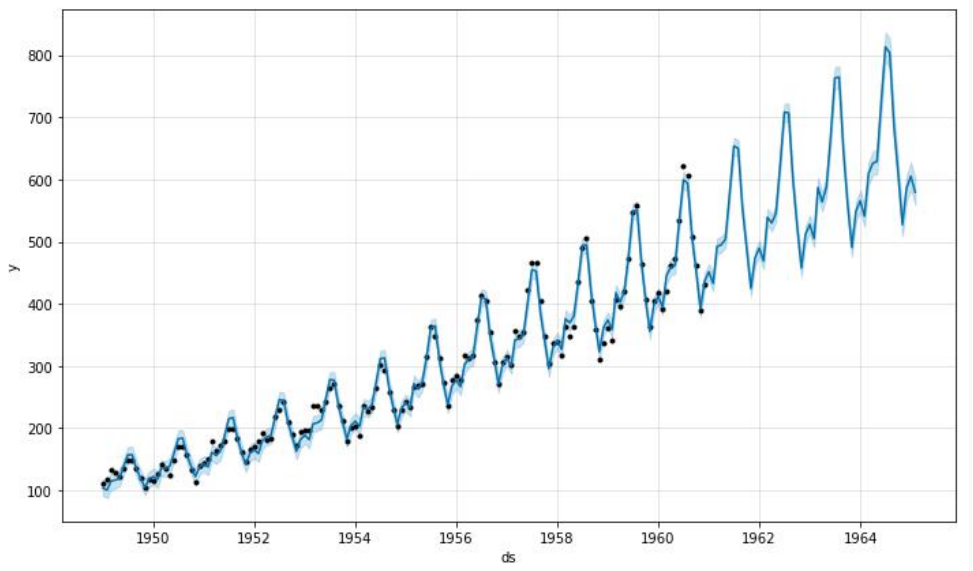


Figura 6: Modello moltiplicativo

Gli iper-parametri (ovvero i parametri che non vengono appresi dall'algoritmo di apprendimento) che vengono presi in considerazione per la costruzione del modello adatto alle analisi che si vogliono effettuare, sono:

- **changepoint_prior_scale**: definisce la quantità di punti, in corrispondenza dei quali si verifica un cambiamento di trend; un valore elevato di questo iper-parametro può portare il modello ad una situazione di overfitting, poiché il comportamento passato viene appresso troppo precisamente, senza riuscire a generalizzare le proprietà fondamentali della serie temporale;

- **seasonality_prior_scale**: tramite questo parametro è possibile controllare la capacità con cui la componente stagionale riesce a fittare i dati; aumentandone il valore, il modello può andare in overfitting;
- **ordine serie di Fourier:** definisce il numero di termini della serie di Fourier; ordini maggiori consentono una migliore approssimazione della funzione da apprendere, ma ciò può portare al fenomeno dell'overfitting.

I tre precedenti parametri vengono scelti, mediante una procedura di **model selection**, di cui si parlerà nella sezione riguardante la validazione.

Si sono individuati due tipi di stagionalità presenti (per ipotesi) nelle metriche che si vogliano analizzare con questo metodo: la periodicità settimanale e la periodicità giornaliera; in particolare la periodicità giornaliera è suddivisa in due sub-stagionalità, ovvero periodicità dei giorni feriali e periodicità del weekend (sabato e domenica).
Come accennato precedentemente, si ipotizza che le metriche analizzate abbiano le periodicità individuate, ma ovviamente ciò non sempre è vero; l'ipotesi fatta non è però così distante dalla realtà: si pensi, per esempio, agli host presenti in un ufficio pubblico; ci si può aspettare che sabato e domenica il traffico sia assente e nei giorni settimanali ci sia traffico solo durante l'orario di lavoro: in tale scenario si può notare la presenza delle stagionalità feriali e festive (sabato e domenica), e di conseguenza della stagionalità settimanale;
Quando le stagionalità individuate esistono realmente nelle metriche analizzate allora Prophet esprimerà la sua massima capacità di predizione; se invece le stagionalità non sono chiaramante visibili, allora il modello creato avrà una minore capacità di rilevazione.

L'analisi delle metriche di rete a medio-lungo termine, si divide in più fasi, ben distinte. Innanzitutto, è importante precisare che viene costruito un modello per ogni metrica di ogni host, poiché ogni metrica ha un comportamento unico, così come gli host.
La prima fase consiste nel recuperare i dati da analizzare dal database contente i punti delle serie temporali.
Successivamente si procede a verificare che ci siano abbastanza punti di una certa metrica, per poter eseguire il training e la conseguente predizione; lo scenario ideale sarebbe avere almeno tre settimane di dati completi (senza punti mancanti nella

serie temporale); poiché Prophet gestisce bene anche serie temporali incomplete, ci si accontenta di avere almeno il 70% dei punti di tre settimane di dati completi, per potere iniziare ad analizzare la metrica. Ovviamente, più punti si hanno a disposizione, migliore sarà la capacità di predizione.

Prima di eseguire il training viene effettuata la validazione del modello, in modo da scegliere i valori da assegnare agli iper-parametri: viene effettuata quindi la cosiddetta model selection. La fase di validazione verrà ampiamente discussa nell'apposita sezione; occorre però notare, che la fase di model selection viene effettuata una volta ogni due settimane per ogni metrica di ogni host, in modo da ammortizzare il costo dell'operazione considerata.

Dopo aver effettuato il training, utilizzando gli iper-parametri scelti durante la fase di model selection, si passa alla predizione di uno o più punti; le predizioni effettuate dovranno essere confrontate con i valori reali assunti dalla metrica sotto analisi.

Per far ciò, di default, Prophet genera degli intervalli di incertezza: il modello creato, oltre a predire il valore esatto della metrica in un certo istante temporale, fornisce una forchetta di possibili valori (che comprende anche il valore predetto) che possono essere assunti dalla metrica in esame; tale range di valori tiene conto dei cambiamenti di trend rilevati durante la fase di training e di eventuali osservazioni affette da rumore [28]. a tal proposito è stata incrementato il valore del parametro interval_width al 99% (il valore di default è 80%), il quale definisce la quantità di dati che andranno ad influire sulla determinazione degli intervalli di incertezza [29]; ciò consente di ottenere rilevazioni meno sensibili e quindi meno inclini all'individuazione di falsi positivi. Quindi, dato il valore reale di una metrica ad un certo istante temporale t, si controlla se esso si trova all'interno del range di valori dell'intervallo di incertezza individuato da Prophet al tempo t; in caso contrario viene generato un alert di comportamento anomalo.

Opzionalemente, è possibile effettuare un ulteriore controllo prima di sollevare la suddetta anomalia: è possibile controllare se nel periodo in cui si è verificato il comportamento sospetto della metrica sono comparse categorie di protocolli/eventi particolari: contatti con host catalogati come malware o collegati ad attività di mining, protocolli di accesso remoto e protocolli sconosciuti (come spiegato nella sezione 6.1.2)

Nel caso in cui all'anomalia rilevata da Prophet venga associata la presenza dei protocolli sopra definiti, la possibilità che si tratti di un vero positivo aumentano

notevolmente; d'altra parte questa tecnica può portare alla rilevazione di falsi negativi; queste ultime considerazioni verranno analizzate in seguito.

La figura 7 mostra il grafico raffigurante i risultati del fitting su circa tre settimane di dati e dalla predizione di dodici nuovi punti nel futuro, della metrica bytes ricevuti.

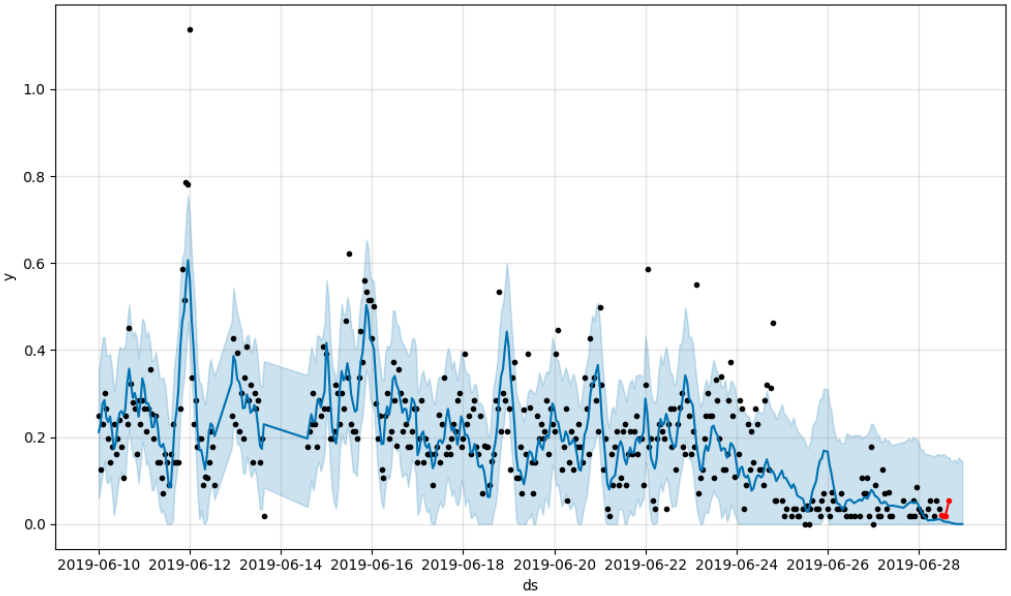


Figura 7: Fitting e predizione della metrica bytes ricevuti

I punti in nero rappresentano i valori della metrica sui quali il modello di Prophet viene allenato; la curva in blu approssima la serie temporale reale, ed è ottenuta inizialmente tramite il processo di fitting (dall'inizio della serie temporale, fino all'istante temporale in cui è presente l'ultimo punto in nero(incluso)), e successivamente grazie alla fase di predizione (dall'ultimo punto in nero (escluso), in poi); l'area in celeste corrisponde al range di valori assunti degli intervalli di incertezza. I punti in rosso sono osservazioni reali della metrica in esame e possono essere confrontati con la curva blu sottostante (valori predetti): in tal caso i valori osservati non sono anomali, poiché ricadono nel range di valori definiti dagli intervalli di incertezza. Si noti la stagionalità e il trend della metrica analizzata, che vengono appresi in modo abbastanza preciso dal modello di regressione creato da Prophet.

Prophet risulta molto efficiente in memoria, in quanto il modello creato utilizza circa 6 MB di RAM. Inoltre, il tempo impiegato per effettuare il fitting su circa tre

settimane di dati (quindi su circa cinquecento punti, poiché si ha un punto ogni ora), la predizione di dodici nuovi punti, ed il controllo delle anomalie è di circa 9 secondi per ogni metrica analizzata (validazione degli iper-parametri esclusa).

In estrema sintesi, l'approccio utilizzato da Prophet nella rilevazione di anomalie di rete presenta i seguenti principali vantaggi:

- Flessibilità: è possibile considerare stagionalità con diverse periodicità, facendo varie assunzioni sul trend della serie;
- Non occorre l'interpolazione di dati eventualmente mancanti;
- Processo di fitting molto efficiente in spazio e tempo;
- (iper-)Parametri umanamente interpretabili, che consentono di migliorare la capacità di predizione del modello creato, facendo assunzioni sulla natura della serie temporale da analizzare.
- Rilevazione semplice di anomalie, grazie alla presenza degli intervalli di incertezza.

6.3. Allarmi e Mitigazione

Generalmente, a seguito della rilevazione di un'anomalia viene generato un'alert e/o viene intrapresa un'azione correttiva automatica.
Nel primo caso, l'allarme generato deve servire all'amministratore di rete o ad altri sistemi automatici di terze parti per prendere atto di quanto avvenuto, analizzare la situazione ed applicare le dovute azioni che consentano di risolvere il problema che si è presentato; ovviamente, il formato dell'allarme deve dare informazioni più dettagliate possibili su ciò che si è verificato.
Se invece si dispone di un sistema di mitigazione, esso risponde in maniera del tutto automatica all'anomalia rilevata, consentendo quindi di risolvere il problema in maniera autonoma, senza il supporto umano.
Spesso i due approcci presentati vengono usati insieme, come viene fatto in questo lavoro di Tesi.

Rilevata un'anomalia, il sistema realizzato lancia un'alert con le seguenti informazioni:

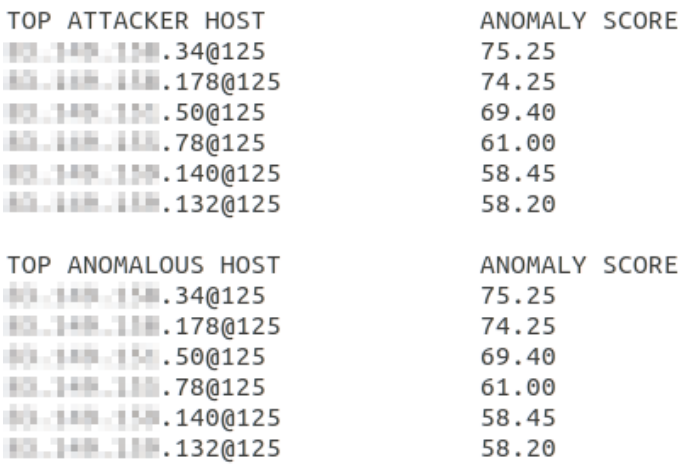
- **Tipo dell'alert**: si può avere sia un allarme che segnala l'inizio dell'anomalia, sia uno che ne sancisce la fine;
- **Tipo dell'anomalia**: fornisce il nome dell'anomalia rilevata;
- **Host/MAC**: indica l'indirizzo IP (Internet Protocol) o l'indirizzo MAC (Media Access Control) su cui è stata rilevata la metrica anomala;
- **Id dell'interfaccia**: identificatore dell'interfaccia di rete sulla quale è avvenuta la rilevazione;
- **Data**: mostra la data in cui è stata rilevata l'anomalia, nel formato YYYY-MM-DDThh:mm:ssZ;
- **Metodo della rilevazione**: indica quale tecnica ha evidenziato l'anomalia (Prophet, RSI, ecc.)
- **Valore anomalo rilevato**: mostra il valore anomalo rilevato.

Per sintetizzare il tutto, si considerino gli allarmi rilevati durante una sessione di analisi, in figura 8:

TYPE	ANOMALY	HOST/MAC	IF	DATE	METHOD	VAL
START	ping_packets	[illegible].145@125	0	2019-06-27T10:35:00Z	TRESHOLD	1.1
END	ping_packets	[illegible].145@125	0	2019-06-27T17:00:00Z	TRESHOLD	
START	ping_packets	[illegible].180@125	0	2019-06-27T13:40:00Z	TRESHOLD	1.2
END	ping_packets	[illegible].180@125	0	2019-06-27T13:45:00Z	TRESHOLD	
START	ping_packets	[illegible].241@125	0	2019-06-27T09:15:00Z	TRESHOLD	1.2
END	ping_packets	[illegible].241@125	0	2019-06-27T10:00:00Z	TRESHOLD	
START	ping_packets	[illegible].241@125	0	2019-06-27T10:20:00Z	TRESHOLD	1.8
END	ping_packets	[illegible].241@125	0	2019-06-27T10:25:00Z	TRESHOLD	
START	ping_packets	[illegible].241@125	0	2019-06-27T12:35:00Z	TRESHOLD	1.1
END	ping_packets	[illegible].241@125	0	2019-06-27T12:55:00Z	TRESHOLD	
START	ping_packets	[illegible].78@125	0	2019-06-27T08:15:00Z	TRESHOLD	1.8
START	ping_packets	[illegible].234@125	0	2019-06-27T09:25:00Z	TRESHOLD	4.0
END	ping_packets	[illegible].234@125	0	2019-06-27T21:25:00Z	TRESHOLD	
START	ping_packets	[illegible].234@125	0	2019-06-27T21:40:00Z	TRESHOLD	1.1

Figura 8: Rilevazione problemi relativi all'ICMP ECHO

Inoltre, ad ogni host viene assegnato uno score, che indica il suo grado di anomalia; lo score viene calcolato considerando le varie problematiche presenti sull'host analizzato: ad ogni anomalia è associato un grado di rilevanza (un float), il quale viene moltiplicato per il numero di volte che l'anomalia è stata rilevata; eseguendo la somma dei valori calcolati, per ogni anomalia rilevata, si ottiene *l'anomaly score*. Al termine delle analisi, è possibile controllare quali sono i x host più anomali, dove x è un parametro configurabile, come mostra la figura seguente.



TOP ATTACKER HOST	ANOMALY SCORE
[redacted].34@125	75.25
[redacted].178@125	74.25
[redacted].50@125	69.40
[redacted].78@125	61.00
[redacted].140@125	58.45
[redacted].132@125	58.20

TOP ANOMALOUS HOST	ANOMALY SCORE
[redacted].34@125	75.25
[redacted].178@125	74.25
[redacted].50@125	69.40
[redacted].78@125	61.00
[redacted].140@125	58.45
[redacted].132@125	58.20

Figura 9: Top host anomali

Oltre alla funzionalità di generare allarmi, il sistema realizzato si propone di mitigare autonomamente alcune anomalie rilevate.

La funzionalità di mitigazione è disabilitata di default, ma può essere attivata opzionalmente. Essa viene applicata ad anomalie che hanno un alto grado di pericolosità, e che vengono rilevate con tecniche che presentano una quantità minima di falsi positivi (valori soglia e Prophet + NDPI). Inoltre, l'attività di mitigazione riguarda gli host locali rilevati anomali per sospetta attività d'attacco: se un host sembra essere sotto attacco DNS flooding, tale problema non verrà mitigato dal sistema; invece se un host è collegato ad attività di DNS exfiltration, il suo IP (IPv4 o IPv6) verrà inserito in una blacklist in modo da non far transitare il traffico da esso generato. Per far ciò si utilizza la tecnologia eBPF (extended Berkeley Packet Filter), che consente l'iniezione di codice (ovvero bytecode, generato dal compilatore BCC (BPF Compiler Collection)) eseguibile nel kernel di Linux, con particolari restrizioni [30]. In particolare, è stato scritto un programma XDP (eXpress Data Path), che sfrutta la tecnologia eBPF per analizzare in modo veloce i pacchetti che arrivano sull'interfaccia di rete.

6.3.1. XDP

La tecnologia XDP fornisce degli strumenti per analizzare e filtrare i pacchetti di rete direttamente all'interno del kernel. Il codice XDP viene eseguito nella parte bassa dello stack protocollare, consentendo quindi di operare sui pacchetti alla massima velocità possibile [31]. Il codice XDP per essere iniettato all'interno del Kernel viene controllato da un validatore; quindi il codice presenta delle restrizioni: per esempio, non sono ammessi cicli ed ogni volta che si accede ai dati raw dei pacchetti è necessario controllare esplicitamente che non si esca dall'area di memoria in cui viene memorizzato il pacchetto stesso [32].

Per questo lavoro di Tesi è stato scritto in XDP un firewall di rete capace di filtrare i pacchetti di rete in arrivo, in base a delle blacklist di indirizzi IPv4, IPv6 e MAC. In particolare, il codice Python eseguito in spazio utente comunica con il codice XDP eseguito in spazio Kernel tramite delle strutture chiave-valore (Tabelle hash); quando viene rilevata un'anomalia che può essere mitigata, si procede a recuperare l'indirizzo dell'host anomalo, il quale viene inserito in una delle mappe sopra citate (a seconda del tipo dell'indirizzo); l'indirizzo verrà eliminato dalla blacklist al momento della chiusura del sistema realizzato. Ogni qualvolta arriva un pacchetto sull'interfaccia di rete il processo XDP si preoccupa di controllare se uno degli indirizzi (MAC o IP) del pacchetto si trova in qualche tabella hash; in caso affermativo si procede a bloccare il pacchetto ricevuto, senza che esso possa risalire lo stack protocollare. È chiaro che le tabelle hash utilizzate fungono da blacklists per il firewall di rete realizzato.

Una blacklist per indirizzi IPv4 in XDP è stata così definita:

```
BPF_TABLE("percpu_hash", uint32_t, uint64_t, ipv4Blacklist, 10000);
```

dove:

- **"percpu_hash"** determina il tipo della tabella, ovvero una tabella hash definita per ogni CPU (per migliorare performance);
- **uint32_t** è il tipo della chiave della tabella; siccome è un indirizzo IPv4 esso occupa 32 bit;

- **uint64_t** è il tipo del campo non chiave della tabella; esso è il contatore dei pacchetti filtrati per il relativo campo chiave, che verrà letto periodicamente dallo spazio utente;
- **ipv4Blacklist** è il nome della tabella;
- **10000** è la capienza della tabella hash.

Tramite codice XDP è stato necessario effettuare il parsing dei pacchetti, in modo da poter estrapolare gli indirizzi MAC, IPv4 e IPv6.
Il pacchetto che arriva sull'interfaccia può essere letto tramite la struttura *xdp_md*, che tra i vari campi contiene i puntatori all'inizio e alla fine del pacchetto.

```
struct xdp_md {
        __u32 data;
        __u32 data_end;
        __u32 data_meta;
        /* Below access go through struct xdp_rxq_info */
        __u32 ingress_ifindex; /* rxq->dev->ifindex */
        __u32 rx_queue_index;  /* rxq->queue_index  */
};
```

Per decodificare il pacchetto si possono utilizzare le strutture presenti nel Kernel di Linux; per esempio per parsare l'header Ethernet si può far uso della struttura *struct ethhdr* nel seguente modo:

```
void* data_end = (void*)(long)ctx->data_end;
void* data = (void*)(long)ctx->data;

struct ethhdr *eth = data; //struct header ethernet
uint64_t nh_off = sizeof(*eth);
if (data + nh_off > data_end) return XDP_DROP; //check bounds
uint64_t macIn = mac2u64(eth->h_source);
uint64_t macEg = mac2u64(eth->h_dest);
if(checkMac(&macIn) || checkMac(&macEg)) return XDP_DROP;
uint16_t h_proto = eth->h_proto;
```

In giallo, si può notare il controllo sui limiti del pacchetto sotto analisi. Le funzioni *mac2u64* e *checkMac*, servono rispettivamente a trasformare un MAC codificato come array di caratteri in un intero a 64 bit e a controllare che il MAC del pacchetto non compaia nella corrispettiva blacklist (in tal caso il pacchetto non risale lo stack protocollare, grazie all'azione *XDP_DROP*). Per completezza si riportano le due funzioni sopra descritte;

```
static inline uint64_t mac2u64(unsigned char* mac) {
        uint64_t macVal;
        macVal = (uint64_t)mac[0] << 40  |
                 (uint64_t)mac[1] << 32  |
                 (uint64_t)mac[2] << 24  |
                 (uint64_t)mac[3] << 16  |
                 (uint64_t)mac[4] << 8   |
                 (uint64_t)mac[5];

        return macVal;
}

static inline int checkMac(uint64_t* mac){
        uint64_t* value = macBlacklist.lookup(mac);
        if(value){
                *value += 1;
                return 1;
        }
        return 0;
}
```

La decodifica degli header IPv4 e IPv6 viene eseguita in modo analogo a quanto illustrato per l'header di livello 2.

In estrema sintesi, XDP consente di implementare un semplice firewall di rete capace di processare i pacchetti in arrivo a velocità molto elevate e nel lavoro di Tesi esso è stato usato per bloccare gli indirizzi di host locali considerati malevoli.

6.4. Architettura Software

In questa sezione viene illustrata l'architettura del software realizzato, necessario per validare i concetti teorici espressi. Si è deciso di costruire un programma multi-processo, in modo da poter eseguire in parallelo l'analisi delle metriche a breve termine e a medio-lungo termine; non si è usata una soluzione multi-thread poiché Python, il linguaggio usato, presenta limitazioni nella gestione dei thread, infatti essi non possono eseguiti in parallelo su processori diversi.

In ogni istante si hanno al massimo tre processi comunicanti. Il processo principale svolge i seguenti compiti:

- Parsing e inizializzazione dei parametri passati tramite linea di comando;

- Connessione al database Influx, contenente i dati delle serie temporali da analizzare;
- Iniezione codice XDP nel kernel, se viene impostata la mitigazione automatica;
- Se viene attivata l'analisi in tempo reale, viene inizializzato lo scheduler in modo da eseguire la rilevazione di anomalie ogni cinque minuti per le metriche a breve termine e ogni ora per le metriche a medio-lungo termine;
- Creazione sotto processi responsabili dell'esecuzione di Prophet;
- Analisi su metriche a breve termine tramite il controllo delle soglie e l'esecuzione della variante dell'RSI;
- Raccolta e stampa delle statistiche delle metriche analizzate;
- Mitigazione host anomali, inserendo nella mappa condivisa con il codice XDP, l'indirizzo da bloccare.

I processi creati per le analisi sulle metriche a medio-lungo termine, comunicano i risultati sulla rilevazione delle anomalie al processo principale tramite delle pipe; il processo principale integra i dati ricevuti, con quelli già in suo possesso in modo da avere una visione globale sulle anomalie rilevate e per poter stampare statistiche cumulative, sia a livello generale che a livello di host.
La scelta di eseguire Prophet su processi diversi è dovuta al fatto che l'analisi delle metriche a medio-lungo termine impiega generalmente un tempo maggiore di cinque minuti, quindi nel periodo in cui è in esecuzione Prophet non sarebbe possibile effettuare alcuna analisi a breve termine.
Ogni volta che viene rilevata un'anomalia essa viene immediatamente stampata su file o schermo in modo da generare un allarme.
Ovviamente tutti i processi creati eseguono delle query sul database, in modo da ottenere i dati su cui effettuare le varie rilevazioni; tale database viene popolato (o è stato popolato) periodicamente dal software di monitoraggio di rete ntop.
Il software può essere usato in due diverse modalità:

- **Modalità real-time**: vengono effettuare query al database periodicamente (ogni cinque minuti per le metriche a breve termine, ogni ora per le metriche a medio-lungo termine) utilizzando l'ora attuale del sistema;

- **Modalità Test**: viene eseguita immediatamente l'analisi sul database già popolato con le serie temporali da analizzare, considerando l'orario di fine analisi definibile dall'utente.

Schematicamente l'architettura del software è rappresenta nella figura sottostante.

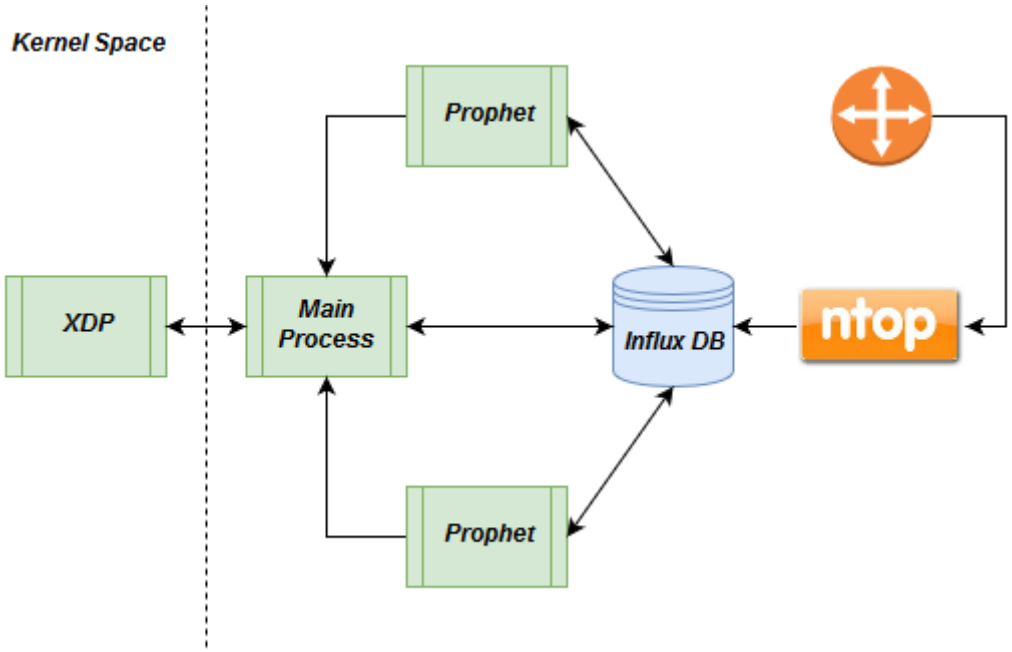


Figura 10: Architettura software

7. Validazione

In questa sezione viene spiegata la scelta dei parametri dei modelli/algoritmi utilizzati (validazione dei modelli), e viene quantificata la capacità di rilevazione di anomalie del software prodotto (validazione della performance di rilevazione);

7.1. Validazione dei Modelli

I parametri che sono stati considerati per il corretto funzionamento delle tecniche di rilevazione proposte sono i seguenti:

- Per la tecnica a soglie (treshold) si hanno i tre valori soglia, discussi in precedenza;
- Nell'RSI va impostata la lunghezza del periodo temporale e la soglia superiore al valore dell'RSI calcolato;
- In Prophet è necessario definire la quantità di dati necessari per poter effettuare il fitting, i tre iper-parametri da cui dipende l'apprendimento della stagionalità e del trend, e la quantità di tempo che intercorre tra due validazioni del modello (per la scelta automatica dei tre iper-parametri); inoltre viene giustificata la scelta del modello moltiplicativo.

Di seguito vengono ricordati i valori assegnati alle soglie fisse, già discussi nella sezione 6.2.1.
La soglia per rilevare problemi nel rapporto tra pacchetti DNS di risposta e richiesta DNS, tra pacchetti ICMP echo di risposta e richiesta e tra pacchetti ARP di richiesta inviati e di risposta ricevuti, è stata fissata al valore di 1, poiché in uno scenario normale il rapporto tra numeratore e denumeratore risulta essere minore o uguale a 1 (non vale il minore per il protocollo ARP), e valori maggiori indicano la presenza un problema.
Per quanto riguarda la soglia per la rilevazione della fuga o dell'infiltrazione di dati tramite DNS, ipotizzando che il protocollo utilizzi UDP e sia dunque presente il

problema della frammentazione, la soglia è stata fissata al valore di 576 bytes per i ragionamenti sulla MTU effettuati precedentemente.

Il valore soglia per la rilevazione di anomalie in tutte le altre metriche a breve termine è stato fissato a 0.50, poiché esso è un valore che sicuramente indica un problema nelle metriche che vengono controllate; valori minori di 0.50 hanno riportato un numero considerevole di falsi positivi, soprattutto quando il volume del traffico non risulta elevato.

Per eseguire il calcolo dell'RSI, è stato necessario scegliere la lunghezza del periodo temporale, ovvero il parametro N descritto nella sezione 6.2.2.
Come per tutti gli oscillatori, più breve è il periodo, più l'oscillatore è sensibile e ha un'ampiezza maggiore, con la possibilità quindi di generare falsi positivi.
Il periodo è stato fissato al valore di 50; ciò significa che ci occorrono 51 punti per poter iniziare l'analisi con l'RSI: considerando che ogni punto viene registrato ogni cinque minuti, occorrono poco più di quattro ore di dati per poter iniziare a calcolare l'RSI.
Il valore è stato scelto con l'obiettivo di minimizzare il numero di falsi positivi; per verificare ciò si sono effettuate due sessioni di rilevazione di anomalie, con due differenti valori assegnati al periodo dell'RSI: quando si è impostato il periodo di lunghezza pari a 25 si sono rilevate anomalie che non sono comparse nella rilevazione dell'RSI con periodo pari a 50: sono stati generati i seguenti report di rilevazione di anomalie (eseguiti sugli stessi dati), il primo eseguito su un periodo lungo 25 (figura 11), il secondo su periodo lungo 50 (figura 12):

```
START   dns_errors      [redacted].126@125      0   2019-06-27T19:20:00Z   RSI
END     dns_errors      [redacted].126@125      0   2019-06-27T19:25:00Z   RSI
START   dns_errors      [redacted].162@125      0   2019-06-27T18:55:00Z   RSI
END     dns_errors      [redacted].162@125      0   2019-06-27T19:00:00Z   RSI
START   dns_errors      [redacted].163@125      0   2019-06-27T20:05:00Z   RSI
END     dns_errors      [redacted].163@125      0   2019-06-27T20:10:00Z   RSI
START   dns_errors      [redacted].46@125       0   2019-06-27T18:50:00Z   RSI
END     dns_errors      [redacted].46@125       0   2019-06-27T18:55:00Z   RSI
START   dns_errors      [redacted].46@125       0   2019-06-27T20:50:00Z   RSI
END     dns_errors      [redacted].46@125       0   2019-06-27T20:55:00Z   RSI
START   dns_errors      [redacted].46@125       0   2019-06-27T21:00:00Z   RSI
END     dns_errors      [redacted].46@125       0   2019-06-27T21:05:00Z   RSI
START   dns_errors      [redacted].183@125      0   2019-06-27T23:20:00Z   RSI
END     dns_errors      [redacted].183@125      0   2019-06-27T23:45:00Z   RSI
START   dns_errors      [redacted].44@125       0   2019-06-27T19:15:00Z   RSI
END     dns_errors      [redacted].44@125       0   2019-06-27T19:20:00Z   RSI
START   dns_errors      [redacted].44@125       0   2019-06-27T20:15:00Z   RSI
END     dns_errors      [redacted].44@125       0   2019-06-27T20:20:00Z   RSI
START   dns_errors      [redacted].65@125       0   2019-06-27T19:05:00Z   RSI
END     dns_errors      [redacted].65@125       0   2019-06-27T19:20:00Z   RSI
```

Figura 11: Report allarmi con periodo RSI pari a 25

START	dns_errors	[redacted].162@125	0	2019-06-27T18:55:00Z	RSI
END	dns_errors	[redacted].162@125	0	2019-06-27T19:00:00Z	RSI
START	dns_errors	[redacted].163@125	0	2019-06-27T20:05:00Z	RSI
END	dns_errors	[redacted].163@125	0	2019-06-27T20:10:00Z	RSI
START	dns_errors	[redacted].46@125	0	2019-06-27T18:50:00Z	RSI
END	dns_errors	[redacted].46@125	0	2019-06-27T18:55:00Z	RSI
START	dns_errors	[redacted].46@125	0	2019-06-27T20:50:00Z	RSI
END	dns_errors	[redacted].46@125	0	2019-06-27T20:55:00Z	RSI
START	dns_errors	[redacted].46@125	0	2019-06-27T21:00:00Z	RSI
END	dns_errors	[redacted].46@125	0	2019-06-27T21:05:00Z	RSI
START	dns_errors	[redacted].65@125	0	2019-06-27T19:05:00Z	RSI
END	dns_errors	[redacted].65@125	0	2019-06-27T19:20:00Z	RSI

Figura 12: Report allarmi con periodo RSI pari a 50

Si noti, che la rilevazione evidenziata in giallo in figura 11, non compare in figura 12. Il tipo di anomalia *"dns_errors"*, si riferisce al rapporto tra i pacchetti DNS di risposta ricevuti di tipo errore e i pacchetti DNS ricevuti totali. La rilevazione risulta però essere un falso positivo come mostra il grafico seguente, in figura 13.

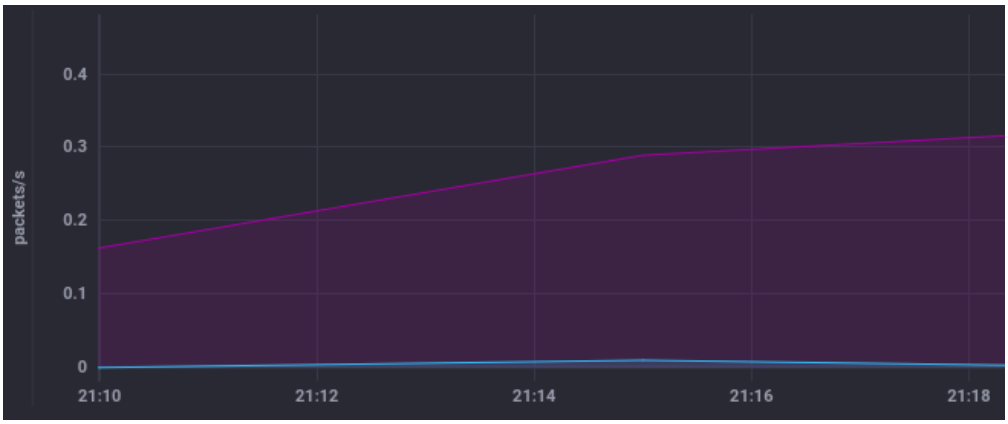


Figura 13: Serie temporale con falso positivo

Si può notare che nell'orario in cui si è rilevata l'anomalia (+2 ore, a causa del fuso orario), si ha un leggerissimo picco (curva blu, che assume un valore di 0.01 pacchetti al secondo) che non indica la presenza un'anomalia; la curva blu rappresenta i valori assunti dalla metrica dei pacchetti DNS di risposta ricevuti di tipo errore, la curva in viola modella il comportamento della metrica dei pacchetti DNS di risposta ricevuti in totale.
La scelta di considerare un periodo più lungo risulta quindi essere corretta.

L'altro parametro da scegliere è il valore massimo dell'RSI per cui un punto della metrica non risulti anomalo; tale parametro è stato fissato al valore di 80. Nell'algoritmo originale si consiglia di usare il valore 70 che genera però un numero maggiore di falsi positivi rispetto alla soglia scelta (80): vengono rilevati anche picchi poco ripidi, che non sono considerati anomali. In figura 14, viene mostrato

un report di analisi (generato considerando gli stessi dati dei report di figura 11 e 12), in cui viene fissata la soglia dell'RSI al valore di 70.

START	dns_errors	[redacted].126@125	0	2019-06-27T19:30:00Z	RSI
END	dns_errors	[redacted].126@125	0	2019-06-27T19:40:00Z	RSI
START	dns_errors	[redacted].162@125	0	2019-06-27T18:55:00Z	RSI
END	dns_errors	[redacted].162@125	0	2019-06-27T19:00:00Z	RSI
START	dns_errors	[redacted].162@125	0	2019-06-27T21:45:00Z	RSI
END	dns_errors	[redacted].162@125	0	2019-06-27T21:50:00Z	RSI
START	dns_errors	[redacted].163@125	0	2019-06-27T20:05:00Z	RSI
END	dns_errors	[redacted].163@125	0	2019-06-27T20:10:00Z	RSI
START	dns_errors	[redacted].46@125	0	2019-06-27T18:50:00Z	RSI
END	dns_errors	[redacted].46@125	0	2019-06-27T18:55:00Z	RSI
START	dns_errors	[redacted].46@125	0	2019-06-27T20:50:00Z	RSI
END	dns_errors	[redacted].46@125	0	2019-06-27T20:55:00Z	RSI
START	dns_errors	[redacted].46@125	0	2019-06-27T21:00:00Z	RSI
END	dns_errors	[redacted].46@125	0	2019-06-27T21:05:00Z	RSI
START	dns_errors	[redacted].44@125	0	2019-06-27T20:15:00Z	RSI
END	dns_errors	[redacted].44@125	0	2019-06-27T20:20:00Z	RSI
START	dns_errors	[redacted].65@125	0	2019-06-27T19:05:00Z	RSI
END	dns_errors	[redacted].65@125	0	2019-06-27T19:20:00Z	RSI

Figura 14: Report con soglia fissata a 70

Confrontando il report di figura 14 con quello di figura 12, generato considerando una soglia fissata a 80, si può vedere che l'allarme evidenziato non compare nella rilevazione con soglia maggiorata, poiché presenta un valore RSI inferiore a 80.

L'allarme evidenzia una situazione in cui è presente un picco, come si può vedere dal grafico in figura 15, ma esso è troppo poco ripido per poter essere considerato anomalo: la curva in blu poco prima del picco assume valore pari a zero, e nel periodo in cui è stata rilevata l'anomalia, assume un valore pari a 0.02 (pacchetti al secondo), quindi il rapporto con la metrica rappresentata dalla curva in viola (pacchetti DNS di risposta ricevuti in totale) presenta un incremento; l'allarme può essere considerato un falso positivo, poiché i valori registrati sono del tutto fisiologici.

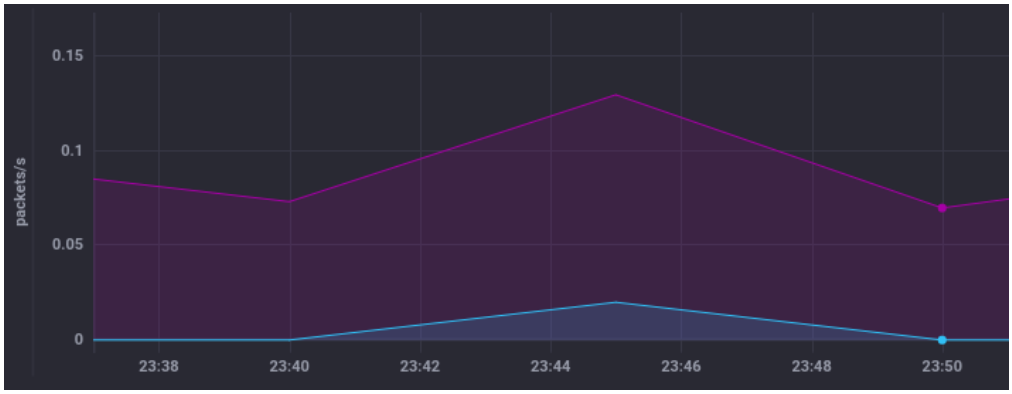


Figura 15: Rilevazione falso positivo, con soglia fissata a 70

La scelta di fissare la soglia a 80 appare quindi sensata. Si ritiene sconveniente aumentare tale valore in quanto ci sarebbe un numero consistente dei falsi negativi.

Per quanto riguarda Prophet, la necessità di circa tre settimane di dati per effettuare il training è di seguito spiegata; si è detto che le metriche analizzate presentano stagionalità settimanale e giornaliera, ed il modello sfrutta tali proprietà per predire nuovi punti; quindi, Prophet deve apprendere come si comporta la metrica settimanalmente e giornalmente per poter fare predizioni includendo le stagionalità considerate; per poter far ciò si ha quindi bisogno di almeno una settimana di dati di training. Eseguire il training su una sola settimana di dati può portare il modello all'overfitting, in quanto esso non apprende i cambiamenti che si possono avere nelle varie settimane e quindi tende a replicare la stagionalità appressa nell'unica settimana di dati di training.
Bisogna quindi considerare almeno due o più settimane per poter effettuare un buon training.
Inoltre, siccome vogliamo scegliere i "migliori" iper-parametri di Prophet tramite un processo di model selection, si è ritenuto opportuno avere circa tre settimane di dati per effettuare una buona validazione del modello. In particolare, durante la fase di model selection i dati a disposizione vengono partizionati come segue: le prime due settimane vengono utilizzate per effettuare il training e l'ultima settimana viene usata per controllare quanto è capace di generalizzare (predire valori futuri) il modello creato. Il training viene quindi eseguito più volte, tante quante sono le possibili configurazioni degli iper-parametri: ogni iper-parametro può assumere due valori diversi, quindi in totale si esegue otto volte il processo di training; alla terminazione di ogni processo di training si verifica quanto si discostano i valori di test (quelli dell'ultima settimana di dati) dai valori predetti, calcolando il RMSE (Root Mean Square Error), ovvero:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(predicted_i - \ actual_i)^2}{n}}$$

dove n è la dimensione del validation set (numero di punti nell'ultima settimana di dati).
La configurazione di iper-parametri che presenta il minimo RMSE, sarà usata per il training completo del modello, che utilizza per intero le tre settimane di dati.

Essendo la validazione un processo costoso la si esegue una volta ogni due settimane per ogni metrica (a medio-lungo termine) di ogni host: è molto difficile che in un periodo di tempo inferiore a due settimane l'host cambi così radicalmente comportamento da dover riscegliere gli iper-parametri.

Nella sezione 6.2.3, si è discusso sulla scelta del tipo di modello da utilizzare: si è deciso di considerare il modello moltiplicativo, piuttosto che quello additivo; la scelta è stata fatta considerando la capacità di apprendere correttamente il comportamento passato della serie temporale: In generale, si è visto che usando un modello additivo, il processo di fitting risulta leggermente meno preciso, come viene mostrato dalle due figure seguenti.

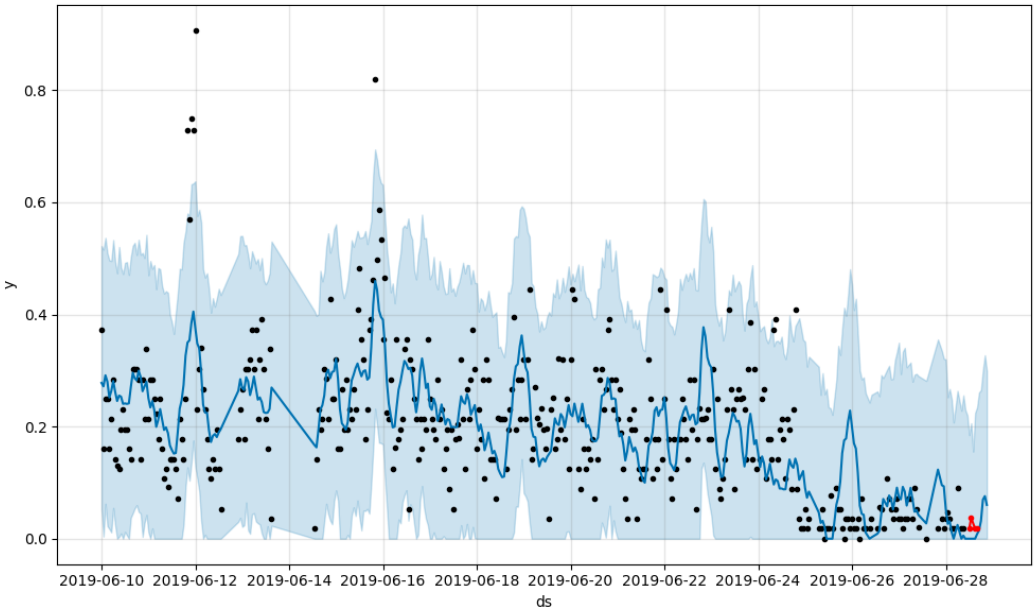


Figura 16: modello additivo

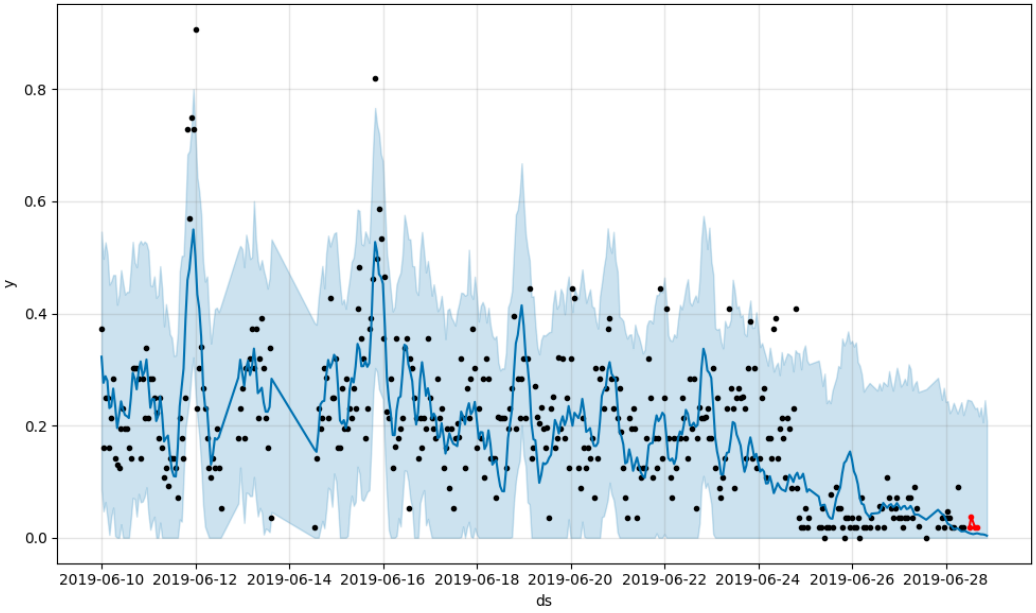


Figura 17: modello moltiplicativo

La figura 16 mostra il modello additivo, la figura 17 quello moltiplicativo; si noti come il modello additivo sia incapace di raggiungere i punti (in nero) posti più in alto nel grafico.

7.2. Validazione della Performance di Rilevazione

In questa sezione viene validato il sistema di rilevazione realizzato nel suo complesso, ovvero viene quantificata la sua capacità di rilevazione di anomalie di rete.

I test di rilevazione di anomalie sono stati effettuati utilizzando i dati di host appartenenti ad un ISP (Internet Service Provider) italiano che gestisce **decina di migliaia di host**. I dati sono stati ottenuti tramite il software di monitoraggio di rete ntop (installato all'interno dell'ISP), responsabile del salvataggio delle serie temporali, rappresentanti il comportamento delle varie metriche degli host, su Influx, un database specializzato nella memorizzazione di dati relativi a serie temporali. È stato effettuato poi il backup del database installato sulla stessa macchina su cui è stato lanciato ntop, in modo da poter eseguire la validazione localmente.

Prima di analizzare l'enorme quantità di traffico circolante all'interno dell'ISP, si è proceduto incrementalmente, analizzando il traffico di piccole reti locali, in modo da affinare le varie tecniche di rilevazione; si è visto che senza un opportuno filtraggio dei dati degli host, le rilevazioni risultavano essere colme di falsi positivi. Inoltre, inizialmente non tutte le metriche considerate erano state implementate e ci si è accorti, quindi, che alcuni comportamenti degli host rimanevano sconosciuti (per esempio non si aveva visibilità sul traffico ARP); si è quindi provveduto ad estendere l'insieme di serie temporali già disponibili in ntop, in modo da poter reperire i nuovi dati da analizzare.

La validazione viene effettuata sulla capacità di rilevazione di anomale delle tecniche e gli algoritmi utilizzati, opportunamente impostati come spiegato nella sezione precedente.
La figura 18 mostra le statistiche cumulative di una sessione di rilevazione di anomalie.

TYPE	TOTAL_CHECK	ANOMALIES	METHOD
ping_packets	2602	1393	TRESHOLD
dns_packets	35406	32408	TRESHOLD
dns_errors	34687	5287	TRESHOLD
port_unreach_srv	4237	25	TRESHOLD
port_unreach_clt	9355	6	TRESHOLD
host_unreach_clt	9054	296	TRESHOLD
host_unreach_srv	1891	0	TRESHOLD
TCP_client_iss	1133	96	TRESHOLD
TCP_server_iss	1119	9	TRESHOLD
dns_size_srv	490	3	TRESHOLD
dns_size_clt	27769	3	TRESHOLD
anmls_flows_srv	29644	517	TRESHOLD
anmls_flows_clt	43539	566	TRESHOLD
dns_errors	13731	14	RSI
port_unreach_srv	2563	9	RSI
port_unreach_clt	5287	13	RSI
host_unreach_clt	4382	11	RSI
host_unreach_srv	1191	4	RSI
TCP_client_iss	357	0	RSI
TCP_server_iss	581	0	RSI
dns_size_srv	308	0	RSI
dns_size_clt	18587	0	RSI
anmls_flows_srv	13695	0	RSI
anmls_flows_clt	23350	2	RSI
flows_as_client	28	2	PROPHET
flows_as_server	199	0	PROPHET
bytes_sent	28	0	PROPHET
bytes_rcvd	199	1	PROPHET

Figura 18: Statistiche cumulative sessione di analisi

Oltre alle statistiche globali, vengono considerate anche le statistiche per host, mostrate in figura 19: ciò consente di avere una visione delle anomalie a livello di

host, in modo da poter capire quali problematiche sono presenti nei singoli sistemi analizzati.

HOST	TYPE	TOTAL_CHECK	ANOMALIES	METHOD
[redacted].130@125				
	dns_errors	97	0	RSI
	anmls_flows_srv	59	0	RSI
	anmls_flows_clt	100	0	RSI
	dns_packets	148	59	TRESHOLD
	dns_errors	148	1	TRESHOLD
	port_unreach_clt	1	0	TRESHOLD
	TCP_client_iss	25	4	TRESHOLD
	TCP_server_iss	27	0	TRESHOLD
	anmls_flows_srv	109	0	TRESHOLD
	anmls_flows_clt	150	0	TRESHOLD

Figura 19: Statistiche per host

Tali informazioni, insieme ai singoli allarmi come già mostrato in figura 8, consentono di capire se un host è affetto o meno da un problema, in modo da poter effettuare la validazione del software proposto.

Per effettuare la validazione, si è proceduto nel modo seguente: per ogni tecnica di rilevazione (Prophet, RSI, Treshold) si sono presi a campione alcuni host rilevati anomali e alcuni host rilevati non anomali, e si è controllato manualmente (mediante l'ausilio di strumenti grafici) che la classificazione risulti corretta.

Per la validazione del treshold, sono stati considerati quattro host che presentano problemi con i messaggi di tipo echo del protocollo ICMP (si hanno più pacchetti echo di risposta rispetto ai pacchetti echo di richiesta), e quattro host che presentano problemi con i pacchetti DNS (si hanno più pacchetti di risposta che di richiesta). Per il controllo dei falsi negativi sono stati analizzati otto host, che non superano la soglia fissata al valore di 1.
Il primo host ad essere analizzato, ha effettivamente problemi con i messaggi ICMP di tipo echo, come mostra la figura seguente.

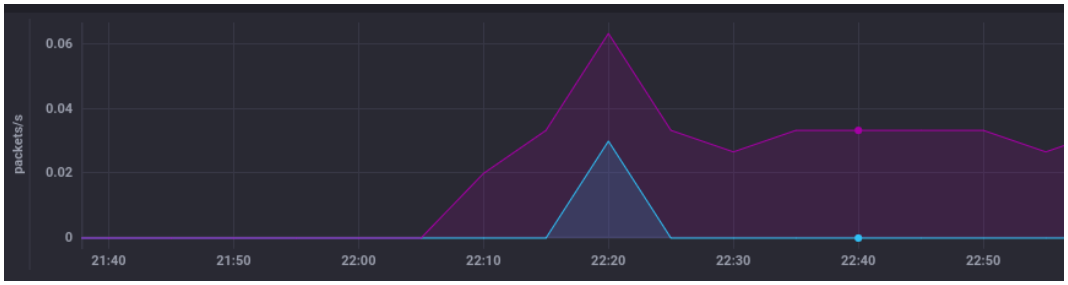


Figura 20: Rilevazione anomalia ICMP echo

In viola si ha la curva dei pacchetti ICMP ricevuti di tipo risposta echo, mentre la curva blu rappresenta i pacchetti ICMP inviati di tipo richiesta echo; si può notare che l'host ha ricevuto delle risposte senza che esso abbia inviato delle richieste: ciò è indubbiamente anomalo.
Gli altri cinque host analizzati presentano anomalie simili, ed i sei host rilevati normali (valori delle metriche sotto la soglia) non presentano effettivamente problemi;
La tecnica del treshold sul campione considerato, ha riportato il 100% in ognuno dei test statistici considerati, ovvero precisione, sensibilità e specificità, definiti nella sezione 3.

Per validare l'RSI sono stati considerati tre host con problemi relativi al numero di pacchetti DNS di risposta con errore, e tre host anomali rispetto al numero di flussi ICMP port unreachable in entrata; inoltre sono stati analizzati analizzati sei host che secondo l'RSI non riportano i problemi sopra citati.
L'analisi dei pacchetti DNS con l'RSI riporta la presenza di un falso positivo (anomalia non significativa) e due veri positivi;
Il grafico della figura sottostante mostra la rilevazione di uno dei due veri positivi.

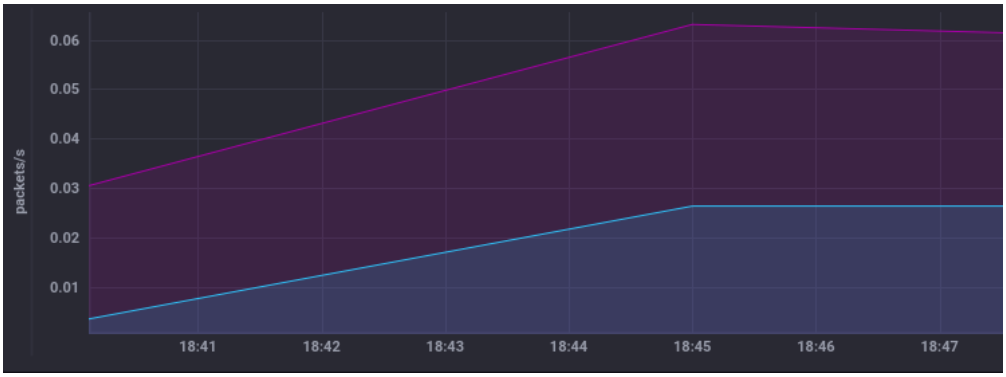


Figura 21: rilevazione di un'anomalia nelle risposte DNS

La curva in blu rappresenta il numero di riposte DNS con errori al secondo, la curva viola rappresenta il numero di risposte DNS corrette; si può notare che il numero di pacchetti con errori è molto vicino al numero dei pacchetti senza errori ed inoltre è presente un picco importante del rapporto tra i due tipi di risposte DNS. Si tratta certamente di un problema.

Il falso positivo viene mostrato in figura 22.

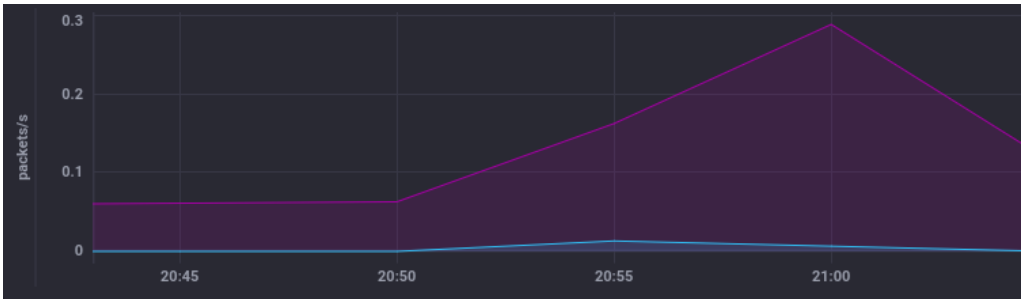


Figura 22: rilevazione falso positivo

Nel grafico di figura 22, alle 20:55, si può notare un leggerissimo picco dei pacchetti DNS con errori; esso non rappresenta una chiara anomalia ed è stato quindi catalogato come falso positivo.

L'analisi dell'RSI sui pacchetti ICMP port unreachable ha dato invece i seguenti risultati: due veri positivi ed un falso positivo.

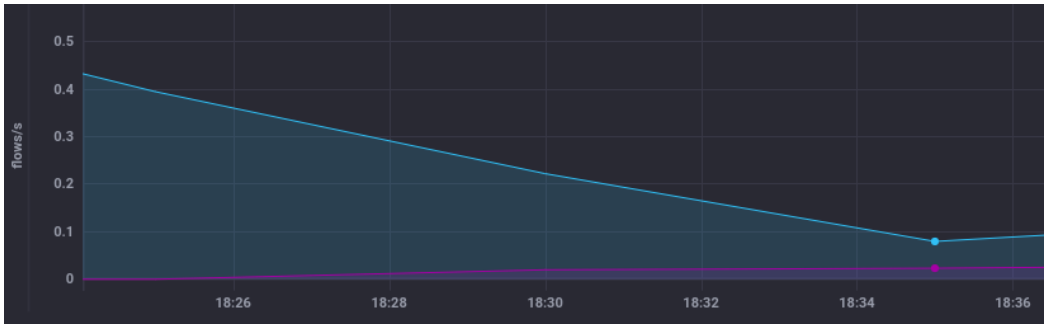


Figura 23: rilevazione vero positivo

Uno dei veri positivi viene mostrato in figura 23; Alle 18:35, la curva blu (flussi port unreachable come server) si inclina leggermente verso l'alto, mentre la curva viola (flussi totali in uscita) si abbassa verso la curva sottostante; si ha quindi un aumento considerevole del rapporto tra flussi port unreachable e flussi totali (rilevazione del picco), ed il valore del rapporto stesso appare sospetto (i flussi port unreachable in entrata sono il 25% dei flussi totali in uscita). Possiamo quindi considerare tale rilevazione un vero positivo

Per quanto riguarda la presenza di eventuali falsi negativi con l'RSI, si sono considerati sei host che non sono stati rilevati anomali; è stato rilevato un solo falso negativo come relativo agli errori DNS, come mostrato in figura 24.

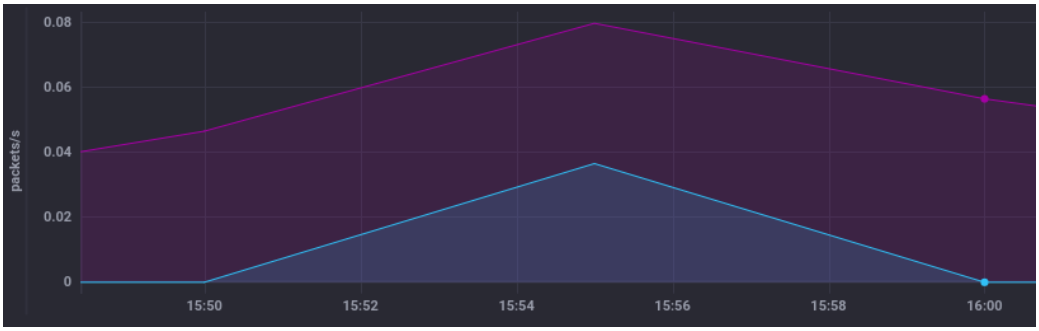


Figura 24: anomalia non rilevata

Tale anomalia non è stata rilevata, poiché i valori anomali sono stati usati per inizializzare il valore dell'RSI; durante l'inizializzazione, se si ha un picco positivo seguito da un picco negativo della stessa dimensione, l'anomalia non viene rilevata, poiché il trend generale della serie temporale risulta costante. Questa è una debolezza presente fisiologicamente nell'RSI.

I risultati sulla validazione dell'RSI sono riassunti nella tabella seguente.

	host Anomali	host non Anomali
Rilevati	4	2
Non rilevati	1	5

Dalla tabella è possibile calcolare la capacità dell'RSI in termini di precisione, specificità e sensitività:

$$PRECISIONE = \frac{TP}{TP + FP} = \frac{4}{4 + 2} = 67\%$$

$$SPECIFICITA' = \frac{TN}{TN + FP} = \frac{5}{5 + 2} = 71\%$$

$$SENSIBILITA' = \frac{TP}{TP + FN} = \frac{4}{4 + 1} = 80\%$$

Da questi risultati, anche se con campione poco numeroso, possiamo concludere che l'RSI riesce ad individuare molto precisamente i picchi presenti nelle serie temporali analizzate, ma a volte essi sono troppo piccoli per essere considerati chiare anomalie. Inoltre, il problema espresso precedentemente circa il periodo di inizializzazione dell'RSI, fa si che possa sfuggire qualche anomalia (presenza di falsi negativi).

Prophet, durante la sessione di analisi relativa al report di figura 18, ha rilevato tre anomalie, che però sono state considerate falsi positivi.
Uno dei falsi positivi viene mostrato in figura 25.

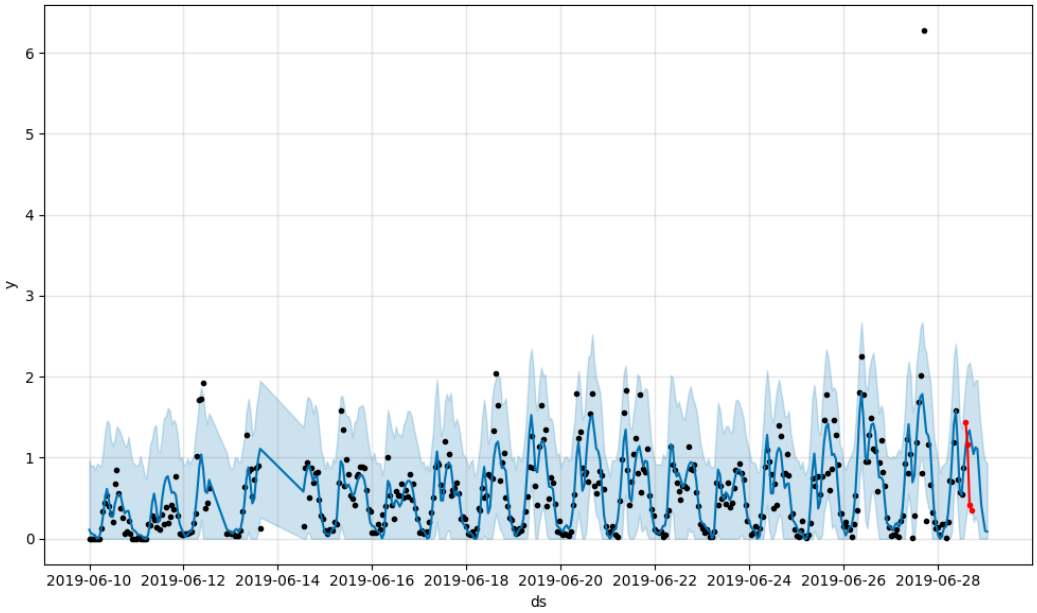


Figura 25: rilevazione falso positivo

Come si può notare, la curva in rosso (valori osservati) si discosta dalla predizione effettuata (curva in blu, sovrapposta temporalmente alla curva in rosso), poiché si ha una variazione del tutto normale, ma imprevedibile, del trend e della stagionalità della serie temporale analizzata.

D'altra parte, sono state effettuate in tutto oltre 400 controlli e non risultano falsi negativi. Prophet presenta quindi una specificità molto vicina al 100%. Vista l'assenza di veri positivi, i valori della precisione e della sensibilità non appaiono significativi.

Inoltre, se viene attivato il controllo delle categorie NDPI, i falsi positivi precedentemente rilevati con Prophet non vengono più individuati, poiché nel periodo di tempo in cui si sono rilevate le anomalie, non è presente traffico delle categorie NDPI sospette discusse nella sezione 6.1.2.

8. Lavori Futuri

Come lavoro futuro, sarebbe interessante implementare una tecnica di correlazione tra le varie serie temporali che si vogliono analizzare, in modo da verificare differenze e somiglianze tra di esse: per esempio, quando si verifica un'anomalia in un host, si potrebbe controllare se tale problema è anche presente in maniera generalizzata in tutti gli host della rete che si sta analizzando, in modo da avere informazioni più dettagliate sull'anomalia rilevata. Attualmente le serie temporali vengono analizzate singolarmente, e non è quindi possibile in nessun modo confrontare il comportamento tra diversi host e diverse metriche;

Inoltre, sarebbe opportuno dare gli allarmi generati dalle attuali tecniche di rilevazione, in input ad un livello di analisi superiore, in modo da filtrare ulteriormente le anomalie e migliorare quindi la capacità di rilevazione: si potrebbe usare per esempio un autoencoder [33], ovvero un tipo di rete neurale artificiale che può essere usata in maniera non supervisionata, per replicare l'input sull'output (apprende la funzione identità), in modo da capire quali allarmi sono ricorrenti (quindi normali) e quali non lo sono (quindi anomali).

Come si è visto, la tecnica di mitigazione appare abbastanza aggressiva e non consente di protegge gli host soggetti ad un possibile attacco; quindi, si vorrebbe sviluppare una tecnica di mitigazione più intelligente, in grado di risolvere meno drasticamente i problemi rilevati, e che sia in grado di arginare anche le anomalie associate ad un attacco subito dagli host analizzati.

9. Conclusioni

Attualmente, il problema della rilevazione di anomalie di rete non sembra avere una soluzione semplice e universale; le varie soluzioni proposte funzionano meglio nel rilevare una certa anomalia rispetto ad un'altra, e hanno diversi gradi di complessità ed efficienza.

In particolare, In questo lavoro di Tesi si è voluto realizzare un sistema intelligente, capace di rilevare con buona precisione ed efficienza le anomalie relative ad un sottoinsieme delle possibili metriche di rete.
Si sono scelte con attenzione le metriche da analizzare, considerando le loro proprietà; solo dopo avere individuato le metriche d'interesse si è proceduto a scegliere la tecnica più consona per analizzarle.

In generale risulta impossibile essere capaci di rilevare anomalie relative ad ogni metrica di rete senza commettere errori, e l'applicativo realizzato non fa ovviamente eccezione, come dimostrano i risultati sulla validazione;

Si è appreso che uno dei punti fondamentali nella rilevazione di problematiche di rete, è il filtraggio dei dati da analizzare: molte volte i dati vengono processati senza nessuna analisi preliminare, la quale risulta fondamentale per una corretta rilevazione di anomalie. Inoltre, è altresì importante definire, implicitamente o esplicitamente, il concetto di normalità di ciò che si vuole analizzare: senza di esso è impensabile catturare l'anomalia.

Infine, si è visto che la tecnica di mitigazione proposta consente di filtrare in modo veloce in pacchetti appaertenenti a host considerati anomali, e va applicata solo ad anomalie rilevate con tecniche che presentano una percentuale quasi nulla di falsi positvi.

10. Referenze

1. J. Armin, B. Thompson, D. Ariu, G. Giacinto, F. Roli and P. Kijewski, "2020 Cybercrime Economic Costs: No Measure No Solution," 2015 10th International Conference on Availability, Reliability and Security, Toulouse, 2015, pp. 701-710; *http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7299982&isnumber=7299862.*
2. D. A. Effendy, K. Kusrini and S. Sudarmawan, "Classification of intrusion detection system (IDS) based on computer network," 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE), Yogyakarta, 2017, pp. 90-94; *http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8285566&isnumber=8285462.*
3. Kumar, Manish, M. Hanumanthappa, and TV Suresh Kumar. "Intrusion detection system-false positive alert reduction technique." ACEEE Int. J. on Network Security 2.03 (2011); *https://pdfs.semanticscholar.org/b11c/a573f506c66aea0205cedb30162b97a5f74c.pdf.*
4. Martin Roesch. Writing Snort Rules. How to write Snort rules and keep your sanity. Version 1.7; *https://paginas.fe.up.pt/~mgi98020/pgr/writing_snort_rules.htm.*
5. Introduction to Zeek; *https://docs.zeek.org/en/stable/intro/index.html.*
6. P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, E. Vázquez, Anomaly-based network intrusion detection: Techniques, systems and challenges, Computers & Security, Volume 28, Issues 1–2, 2009, Pages 18-28; *https://www.sciencedirect.com/science/article/pii/S0167404808000692.*
7. Callegari, C., & Cyprus, N. (2009). Statistical approaches for network anomaly detection. Proc. ICIMP. *https://www.researchgate.net/profile/Christian_Callegari/publication/242*

607190_Advanced_Statistical_Approaches_for_Network_Anomaly_Detection_3_hours_tutorial/links/56bb5e3308ae090818681091.pdf.

8. Mohiuddin Ahmed, Abdun Naser Mahmood, Jiankun Hu, A survey of network anomaly detection techniques, Journal of Network and Computer Applications, Volume 60, 2016, Pages 19-31; *http://www.sciencedirect.com/science/article/pii/S1084804515002891*.
9. Mirsky, Yisroel, et al. "Kitsune: an ensemble of autoencoders for online network intrusion detection." arXiv preprint arXiv:1802.09089 (2018). https://arxiv.org/abs/1802.09089.
10. M. Gupta, J. Gao, C. C. Aggarwal and J. Han, "Outlier Detection for Temporal Data: A Survey," in IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 9, pp. 2250-2267, Sept. 2014. *http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6684530&isnumber=6871455*.
11. Chatfield, C. (1978). The Holt-winters forecasting procedure. Journal of the Royal Statistical Society: Series C (Applied Statistics), 27(3), 264-279. *https://www.researchgate.net/profile/Fabrice_Clerot/post/Can_anyone_help_with_extracted_patterns_of_attributed_graph_as_a_time_series/attachment/59d62cffc49f478072e9e4a5/AS%3A273553243475969%401442231583503/download/Kalekar+-+Exponential+smoothing.pdf.*
12. Malhotra, P., Vig, L., Shroff, G., & Agarwal, P. (2015, April). Long short term memory networks for anomaly detection in time series. In Proceedings (p. 89). Presses universitaires de Louvain. *https://link.springer.com/chapter/10.1007/978-1-4471-0219-9_20.*
13. Datadog anomaly monitor; *https://docs.datadoghq.com/monitors/monitor_types/anomaly/*.
14. SignalFx detectors and alerts; *https://docs.signalfx.com/en/latest/detect-alert/index.html.*
15. Wikipedia contributors. (2019, May 7). Sensitivity and specificity. In Wikipedia, The Free Encyclopedia. Retrieved 16:25, June 4, 2019; *https://en.wikipedia.org/w/index.php?title=Sensitivity_and_specificity&oldid=895891646.*

16. Mahjabin, T., Xiao, Y., Sun, G., & Jiang, W. (2017). A survey of distributed denial-of-service attack, prevention, and mitigation techniques. International Journal of Distributed Sensor Networks; *https://doi.org/10.1177/1550147717741463.*

17. Nadler, A., Aminov, A., & Shabtai, A. (2019). Detection of malicious and low throughput data exfiltration over the DNS protocol. Computers & Security, 80, 36-53; *https://www.sciencedirect.com/science/article/pii/S0167404818304000.*

18. Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2011). Surveying port scans and their detection methodologies. The Computer Journal, 54(10), 1565-1581; *https://academic.oup.com/comjnl/article/54/10/1565/634732.*

19. Chen, Shigang & Ranka, Sanjay. (2004). An Internet-Worm Early Warning System. GLOBECOM - IEEE Global Telecommunications Conference. 4. 2261 - 2265 Vol.4. 10.1109/GLOCOM.2004.1378411; *https://www.researchgate.net/publication/4116108_An_Internet-Worm_Early_Warning_System.*

20. Relative Strength Index. (7 ottobre 2018). Wikipedia, L'enciclopedia libera; *it.wikipedia.org/w/index.php?title=Relative_Strength_Index&oldid=100177438.*

21. Relative Strength Index (RSI): Calculation; *http://cns.bu.edu/~gsc/CN710/fincast/Technical%20_indicators/Relative%20Strength%20Index%20(RSI).htm*

22. Wikipedia contributors. (2019, July 1). Autoregressive integrated moving average. In Wikipedia, The Free Encyclopedia. *https://en.wikipedia.org/w/index.php?title=Autoregressive_integrated_moving_average&oldid=904398305*

23. Fried R., George A.C. (2011) Exponential and Holt-Winters Smoothing. In: Lovric M. (eds) International Encyclopedia of Statistical Science. Springer, Berlin, Heidelberg. *https://doi.org/10.1007/978-3-642-04898-2_244*

24. Wikipedia contributors. (2019, July 2). Long short-term memory. In Wikipedia, The Free Encyclopedia. *https://en.wikipedia.org/w/index.php?title=Long_short-term_memory&oldid=904463056*

25. Taylor SJ, Letham B. 2017. Forecasting at scale. PeerJ Preprints 5:e3190v2; *https://doi.org/10.7287/peerj.preprints.3190v2.*

26. Wikipedia contributors. (2019, May 31). Limited-memory BFGS. In Wikipedia, The Free Encyclopedia. Retrieved 17:35, June 29, 2019; *https://en.wikipedia.org/w/index.php?title=Limited-memory_BFGS&oldid=899609391*

27. Prophet, Multiplicative Seasonality; *https://facebook.github.io/prophet/docs/multiplicative_seasonality.html.*

28. Prophet, Uncertainty Intervals; *https://facebook.github.io/prophet/docs/uncertainty_intervals.html.*

29. Bartosz Mikulski. Understanding uncertainty intervals generated by Prophet; *https://www.mikulskibartosz.name/understanding-uncertainty-intervals-generated-by-prophet/.*

30. BPF and XDP Reference Guide; *https://cilium.readthedocs.io/en/latest/bpf/.*

31. Introduction to XDP; *https://www.iovisor.org/technology/xdp*

32. XDP programs with eBPF; *https://prototype-kernel.readthedocs.io/en/latest/networking/XDP/end-user/coding.html#special-xdp-ebpf-cases*

33. Autoencoders; *http://ufldl.stanford.edu/tutorial/unsupervised/Autoencoders/*

11. Appendice

11.1. Link al Progetto Realizzato

Il software realizzato è stato caricato online, sulla piattaforma Github, al seguente indirizzo: *https://github.com/SalvatoreCostantino/Network-Time-Series-Analyzer*