Rilevamento di anomalie ed intrusion detection in architetture ICT di sistemi di controllo energetico

D. Cerotti[1], D. Codetta-Raiteri[1], G. Dondossola[2], L. Egidi[1,3], G. Franceschinis[1,3], L. Portinale[1,3], R. Terruggia[2]

[1]Istituto di Informatica, DiSIT, Univ. Piemonte Orientale
[2]Ricerca sul Sistema Energetico, RSE S.p.A.
[3]AI@UPO Research Center
{davide.cerotti,daniele.codetta,lavinia.egidi,giuliana.franceschinis,luigi.portinale}@uniupo.it
{roberta.terruggia,giovanna.dondossola}@rse-web.it

Abstract

Questo documento descrive alcune metodologie di Intelligenza Artificiale e Machine Learning sviluppate ed analizzate nell'ambito di un progetto di collaborazione tra Università del Piemonte Orientale e RSE - Ricerca sul Sistema Energetico relativo a tematiche di intrusion detection per infrastrutture energetiche. In particolare vengono illustrate esperienze di: analisi e sperimentazioni di funzioni di rilevamento di anomalie per la resilienza dei sistemi di controllo elettrico; metodi di rilevamento delle anomalie cyber in architetture Cloud/Fog e IoT.

1 Introduzione

Una delle priorità principali nel campo della cyber-security individuata a livello internazionale è sicuramente la protezione delle piattaforme digitali utilizzate da infrastrutture critiche come quelle relative ai sistemi energetici. Le metodiche di monitoraggio e rilevamento di anomalie giocano un ruolo fondamentale nell'intercettare tempestivamente azioni ostili, interrompere l'avanzamento dei processi di attacco ed implementare rimedi efficaci. In questo contesto l'uso di modelli di Intelligenza Artificiale e di Machine Learning in particolare risultano essere uno strumento particolarmente importante per portare le metodiche di anomaly detection ad un livello di maturità adeguato.

In questo contributo descriviamo alcuni approcci analizzati all'interno di un progetto congiunto tra Università del Piemonte Orientale e RSE - Ricerca sul Sistema Energetico, avente come focus il rilevamento di anomalie cyber nei sistemi energetici. In particolare vengono illustrate esperienze di:

- analisi e sperimentazioni di funzioni di rilevamento di anomalie per la resilienza dei sistemi di controllo elettrico;
- metodi di rilevamento di anomalie cyber in architetture emergenti quali Cloud/Fog e IoT.

2 Detection di attacchi a infrastrutture emergenti per mezzo di algoritmi di Machine Learning

Una particolare tipologia di analisi effettuata mira a confrontare l'efficacia di strumenti di Machine Learning (ML) per il rilevamento di anomalie cyber. Si è quindi proceduto ad analizzare alcuni modelli tramite la loro costruzione e valutazione su un dataset di letteratura disponibile in forma aperta. Il dataset è descritto in [Hindy *et al.*, 2020] ed è focalizzato sul protocollo MQTT-SN (Message Queuing Telemetry Transport Sensor Networks)[Stanford-Clark e Truong, 2013]. In particolare, abbiamo analizzato la versione relativa al flusso di traffico bidirezionale (biflow) in quanto le relative features sembrano essere particolarmente interessanti per la costruzione di un sistema di Intrusion Detection (IDS) basato su metodi ML.

Le situazioni considerate comprendono 4 potenziali metodi di attacco (2 di tipo scan e 2 di tipo forza bruta) oltre allo scenario di traffico normale (`(0) normal`), per un totale di 5 possibili scenari. L'attaccante può eseguire le seguenti tipologie di attacco alla rete IoT: aggressive scan (`(1) scan_A`), UDP scan (`(2) scan_sU`), Sparta SSH brute force (`(3) sparta`), MQTT brute force (`(4) mqqt_bf`). Il dataset simula una rete MQTT-SN realistica nello scenario di operatività normale. Le features finali sono ottenute tramite elaborazione dei dati raw ottenuti tramite `tcpdump` (.pcap files). Il totale è di 28 attributi descrittivi con l'aggiunta dell'attributo di classe (`is_attack`).

Il numero di istanze presenti nel dataset risulta essere di circa 260k con una distribuzione delle classi molto sbilanciata verso la classe di traffico normale.

2.1 Feature Selection

Come prima operazione si è proceduto ad effettuare una selezione delle feature rilevanti sulla base delle classi presenti utlizzando un algoritmo piuttosto standard di filtering [Portinale e Saitta, 1998], ossia l'algoritmo CFS (Correlation-Based feature Selection)[Hall, 1998]. La selezione risulta ridurre drasticamente il numero di features da 28 (valore originario) a 5 ed in particolare: la porta sorgente (`prt_src`), la massima lunghezza di pacchetto forward

(`max_pkt_length`), il numero di bytes del flusso backward (`bwd_num_bytes`), il numero di flag push del flusso backward (`bwd_num_psh_flags`) ed il numero di flag reset sempre del flusso backward (`bwd_num_rst_flags`). La riduzione permette inoltre di poter testare anche approcci di tipo lazy learning come la classificazione k-NN di cui parleremo più avanti. Tale approccio infatti soffre della cosiddetta *curse of dimensionality* e non è consigliabile quando il numero di features è elevato.

2.2 Clustering degli scenari

Una prima metodica di analisi possibile avendo a disposizione diverse tipologie di attacco è quella di verificare se le descrizioni dei diversi pacchetti si raggruppano nei diversi tipi di attacco tramite metodi di clustering. Abbiamo considerato due diverse metodiche ed in particolare il *K-means clustering* basato su metrica di distanza ed un clustering probabilistico basato sul metodo di Expectation Maximization *EM clustering*. Si è proceduto ad una valutazione esterna (clusters vs classi) considerando un numero di clusters pari al numero delle potenziali classi, ossia 5. In entrambi i casi si evidenziano 3 clusters che contengono praticamente solo istanze di un determinato tipo di attacco, ed in particolare gli attacchi di tipo `scan` e lo `sparta`. I pacchetti normali e quelli di attacco forza bruta MQTT finiscono invece per essere inseriti in vari clusters. Possiamo quindi concludere che un'analisi di tipo non supervisionato non risulta essere particolarmente efficace.

2.3 Classificazione dei pacchetti

Abbiamo quindi rivolto la nostra attenzione ad alcune metodiche di classificazione, considerando quelle per così dire rappresentative di diverse categorie, ed in particolare: **multinomial logistic regression (LR)** con parametro ridge pari a $1e-08$ per evitare overfitting; **decision tree induction (C4.5)** con pruning basato su confidence factor; **naive Bayes classifier (NB)** assumendo quindi l'indipendenza degli attributi delle istanze, data la classe; **Bayes net classifeier (BNC)** che rimuove l'assunzione del NB ed utilizza l'algoritmo K2 [Cooper e Herskovits, 1992] per il learning della struttura e dei parametri; **multi-layer perceptron (MLP)** con 1 livello nascosto composto da 5 unità nascoste; infine **k nearest neighbour classifier (kNN)** utilizzando un numero di vicini $k = 1$.

Tutti i metodi di cui sopra sono stati validati utilizzando il tool `Weka` nella sua versione 3.8.4 ed effettuando una 10-fold cross validation, in modo che i risultati calcolati possano essere considerati significativi e non viziati da overfitting del modello.

I risultati ottenuti con i vari modelli testati sono molto simili; a titolo d'esempio riportiamo in Figura. 1 l'albero di decisione ottenuto con C4.5 e la relativa valutazione in 10fold CV.

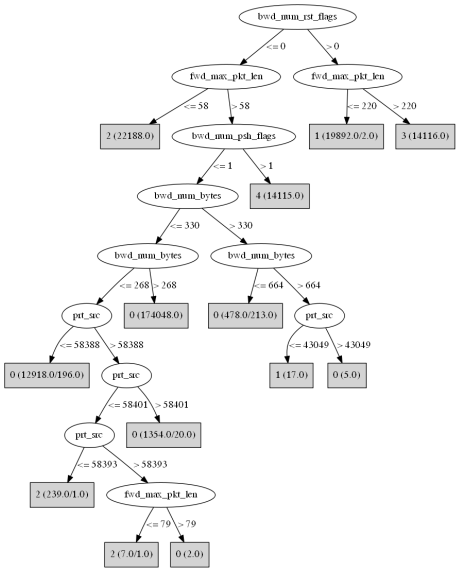
Notare come l'attacco n.3 (`sparta`) sia facilmente identificabile controllando 2 soli parametri: che vi sia almeno un flag di reset sul flusso backward, e che la lunghezza massima dei pacchetti nel flusso forward sia maggiore di 220 bytes. Il fatto che un classificatore ad albero abbia questo tipo di performance, unito all'uso di un numero limitato di attributi, rende questo approccio particolarmente interessante in un' ottica di spiegazione del risultato (cosa decisamente più complicata con altri modelli quali MLP o LR in particolare).

```
=== Summary ===

Correctly Classified Instances      258936               99.8292 %
Incorrectly Classified Instances       443                0.1708 %
Kappa statistic                          0.9962
Mean absolute error                      0.001
Root mean squared error                  0.0227
Relative absolute error                  0.5629 %
Root relative squared error              7.5462 %
Total Number of Instances           259379
```

(a) j48 Statistiche.

```
=== Confusion Matrix ===

      a      b      c      d      e    <-- classified as
 188343      0      4      0     31 |      a = 0
      0  19907      0      0      0 |      b = 1
      0      2  22432      0      0 |      c = 2
      0      0      0  14116      0 |      d = 3
    405      1      0      0  14138 |      e = 4
```

(b) j48 Matrice confusione.



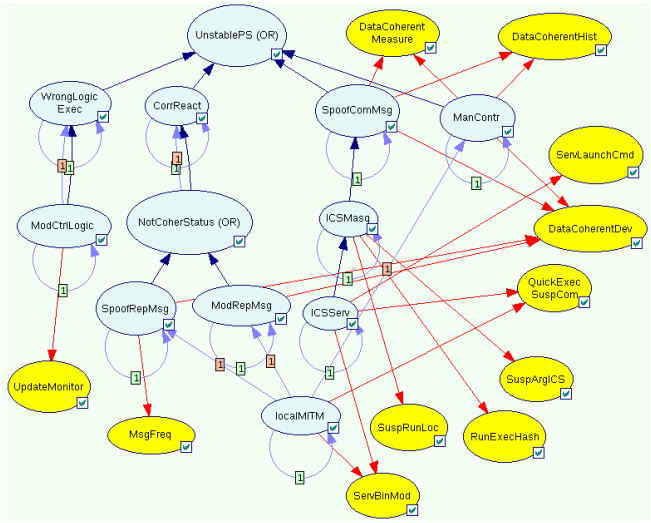
(c) j48 modello.

Figura 1: Decision tree: risultati

3 Analisi e rilevamento di attacchi alle smart grid tramite Reti Bayesiane Dinamiche

L'analisi basata su Machine Learning presuppone la disponibilità di dataset per l'apprendimento dei modelli. Un approccio alternativo, si basa sulla costruzione del modello basandosi sulla conoscenza di possibili attività avversariali derivata dal progetto MITRE ATT&CK e sue estensioni [The MITRE Corporation, 2015], che mira alla costruzione di una base condivisa di conoscenze delle tecniche di compromissione e intrusione, a partire da report di casi reali di attacchi informatici. Per analizzare e rilevare gli attacchi si è applicata

una metodologia basata sulla costruzione di una Rete Bayesiana Dinamica [Portinale e Codetta, 2015] (DBN). La metodologia prevede di costruire un modello iniziale dell'attacco rappresentandolo con un *grafo d'attacco*, i cui nodi sono stati, mentre gli archi rappresentano dei passi elementari d'attacco. La definizione di tali passi è parzialmente derivata dal progetto citato [The MITRE Corporation, 2015].



Tecniche, eventi, stati		
Abbreviazioni	Nomi completi	Tempi medi
ModCtrlLogic	Modified Control Logic	50
localMITM	Local Man In The Middle	2
SpoofRepMsg	Spoof Report Message	20
ModRepMsg	Modified Report Message	10
ICSServ	ICS Service	2
ICSMasq	ICS service masquerading	2
SpoofComMsg	Spoof Command Message	50
ManContr	Manipulated Control	30
WrongLogicExec	WrongLogicExecution	0
CorrReact	Correct Reaction	0
NotCoherStatus	Not Coherent Status	0
UnstablePS	Unstable Power System	0

Figura 2: Modello DBN dell'attacco all'ICS.

La rete rappresenta un attacco che ha l'obiettivo di disturbare i sistemi di controllo presenti nelle smart grid che regolano la gestione di risorse energetiche distributite (DER), quali ad esempio impianti di generazione fotovoltaica. Nell'esempio si assume che l'attaccante abbia già compromesso un sistema di controllo, supervisione e acquisizione dati (SCADA) all'interno della rete di controllo industriale (ICS) dell'infrastruttura e la DBN descrive i possibili restanti passi per causare l'instabilità energetica del sistema. I nodi azzurri rappresentano i passi d'attacco, ad esempio l'uso della tecnica *Local Man In The Middle*, o nel caso di *WrongLogicExecution* l'esecuzione automatica di comandi dannosi, mitigata da un sistema di controllo imperfetto, a seguito di una modifica della logica di controllo (*Modified Control Logic*) perpetrata dall'attaccante, o stati del sistema come l'instabilità della rete elettrica (*UnstablePS(OR)*). Maggiori dettagli dell'attacco sono descritti in [Cerotti *et al.*, 2020].

I nodi gialli indicano delle analitiche, definite dal MITRE come evidenze dell'uso di deteminate tecniche. L'implementazione all'interno del sistema di monitoraggio di un sensore che rileva tali evidenze permette di attivare un allarme qualora la tecnica venga usata durante un attacco. Ad esempio il rilevamento di un incremento nella frequenza di ricezione dei periodici messaggi di report (rappresentato nella DBN dal nodo *MsgFreq*) potrebbe indicare un tentativo da parte dell'attaccante di spoofing di tali messaggi (*SpoofRepMsg*); le evidenze pertanto costituiscono le variabili osservabili della DBN. Ai nodi sono associati valori binari che indicano l'occorrenza della corrispondente tecnica, evento o stato, inoltre in una DBN tali valori possono evolvere in tempi discreti. Un arco che connette due nodi indica che il valore di un nodo influenza quello di un altro. Ad esempio, l'arco da *localMITM* a *ManContr* indica che l'esecuzione della tecnica *Manipulated Control* è possibile solo dopo aver completato con successo *Local Man In The Middle*. Inoltre il valore ad un determinato tempo può essere influenzato dal suo valore all'istante precedente. Ad esempio l'arco che connette *localMITM* a se stesso (self-loop) rappresenta l'evoluzione temporale della tecnica: se ad un certo tempo *Local Man In The Middle* non è attiva, al passo successivo avrà una determinata probabilità di attivarsi. I nodi *WrongLogicExec* e *CorrReact* hanno un comportamento particolare che serve a modellare dei meccanismi di difesa reattivi che intervengono all'occorrenza di un tentativo di attacco. Ad esempio, si è assunto che gli effetti della tecnica *Modified Control Logic* possano essere annullati dall'intervento di un meccanismo automatico di verifica della consistenza della logica di controllo, ma che tale sistema di protezione abbia successo solo il 20% delle volte. Similmente abbiamo assunto un meccanismo di difesa che annulla con una probabilità del 30% il tentativo di un attaccante di innescare un'azione correttiva errata indotta dalla fasulla segnalazione di uno stato incoerente (*Not Coherent Status*) della rete elettrica.

Stabilite tutte le dipendenze fra i vari nodi, per completare la definizione del modello è necessario determinare il valore dei suoi parametri, ossia per ogni nodo n le probabilità condizionate che in un certo istante, in base alla configurazione attuale dei valori del nodo n e dei suoi genitori, si transiti ad un altro valore, o si mantenga lo stesso, al passo successivo. L'insieme di tali probabilità condizionate e dei valori iniziali (al tempo 0) delle variabili determina l'evoluzione stocastica dell'intero sistema nel tempo.

3.1 Approssimazione del tempo continuo

La rete DBN integra un modello di tempo discreto e quindi è in grado di rappresentare e valutare la probabilità che una determinata tecnica occorra dopo un'altra che la abilita, oppure calcolare la probabilità di una determinata sequenza di step di attacco e quindi determinare le probabilità dei vari percorsi di attacco. Tuttavia non permette di calcolare le distribuzioni di probabilità dei tempi di completamento delle diverse tecniche o di interi percorsi d'attacco. Questo tipo di indicatori però sarebbero molto utili per un analista di sicurezza, in quanto potrebbero supportarlo nel decidere come reagire durante un tentativo di attacco in corso.

Si è quindi approssimato il sistema a tempo continuo con una DBN a tempo discreto. In sintesi, dati in input delle stime dei tempi medi di completamento di ciascuna tecnica, la metodologia prevede di: i) determinare la durata dell'intervallo di tempo ΔT fra un passo ed il successivo di un modello a tempo discreto; ii) per ciascuna variabile della DBN originaria calcolare le probabilità condizionate di cambiamento, in un singolo passo, del valore della variabile. In questo modo si deriva un modello DBN a tempo discreto in cui tutte le probabilità condizionate sono riscalate in base al ΔT calcolato. L'approssimazione di un tempo continuo si ottiene assumendo che tra un step della DBN con parametri riscalati ed il successivo trascorra un tempo ΔT con tale valore sufficientemente piccolo (rispetto ai valori dei parametri in input). Tale approssimazione permette di calcolare le distribuzioni di probabilità dei *time to compromise* (TTC) delle diverse tecniche e di interi path di attacco.

3.2 Risultati analisi

Dalla DBN si sono derivati i valori che approssimano il tempo continuo usando in input i tempi medi elencati nella tabella sottostante la Figura 2. Le stime dei tempi medi di ciascuna tecnica sono state ottenute in collaborazione con gli esperti di RSE per tenere in considerazione le specificità dei sistemi di controllo dei DER.

In Figura 3 sono illustrate le distribuzioni cumulative di probabilità di completamento delle diverse tecniche e di instabilità del sistema. In questo caso le distribuzioni sono calcolate in assenza di evidenze, ad esempio a causa della mancanza del sistema di monitoraggio. Le probabilità di completamento ad un determinato istante di tempo rispecchiano i possibili percorsi dell'attacco. In particolare le curve di *localMITM* e *ICSServ* coincidono in quanto entrambe non dipendono da tecniche precedenti nella DBN ed hanno lo stesso tempo medio di completamento. Al contrario *ManContr* ha tempi di completamento più lunghi perché il suo tempo medio è pari a a 30 ed inoltre dipende dal precedente completamento di *localMITM*.

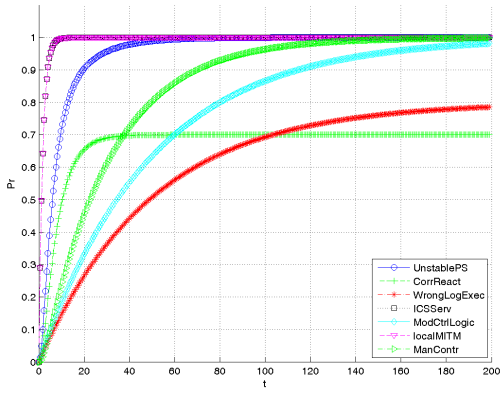


Figura 3: Distribuzione dei TTC delle tecniche e dell'intero sistema.

Riferimenti bibliografici

[Cerotti *et al.*, 2020] D. Cerotti, D. Codetta, G. Dondossola, L. Egidi, G. Franceschinis, L. Portinale, e R. Terruggia. Evidence-based analysis of cyber attacks to security monitored distributed energy resources. *Applied Sciences*, 10, 2020.

[Cooper e Herskovits, 1992] G. Cooper e E. Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, 1992.

[Hall, 1998] M. A. Hall. *Correlation-based Feature Subset Selection for Machine Learning*. PhD thesis, University of Waikato, Hamilton, New Zealand, 1998.

[Hindy *et al.*, 2020] H. Hindy, E. Bayne, M. Bures, R. Atkinson, C. Tachtatzis, e X. Bellekens. Machine learning based iot intrusion detection system: An mqtt case study (mqtt-iot-ids2020 dataset). In *Lecture Notes in Networks and Systems*, volume 180, pages 73–84. Springer, 2020.

[Portinale e Codetta, 2015] L. Portinale e D. Codetta. *Modeling and analysis of dependable systems: a probabilistic graphical model perspective*. World Sc., 2015.

[Portinale e Saitta, 1998] L. Portinale e L. Saitta. Feature selection. Technical report, University of Dortmund, 1998.

[Stanford-Clark e Truong, 2013] A. Stanford-Clark e H.L. Truong. MQTT for sensor networks (mqtt-sn) protocol specification, 2013. IBM Corp. version 1,2.

[The MITRE Corporation, 2015] The MITRE Corporation. Adversarial tactics, techniques and common knowledge (ATT&CK), 2015. https://attack.mitre.org/.