SAPIENZA
UNIVERSITÀ DI ROMA

# Using Sensor and Process Noise Fingerprint to Detect Cyber Attacks in CPS


Facoltà di Ingegneria dell'Informazione, Informatica e Statistica

Corso di Laurea in Informatica

Candidato

Andrei Laurentiu Lepadat
Matricola 1677093

Relatore

Prof. Enrico Tronci



Anno Accademico 2020/2021

Tesi non ancora discussa

---

**Using Sensor and Process Noise Fingerprint to Detect Cyber Attacks in CPS**
Tesi di Laurea. Sapienza – Università di Roma



Questa tesi è stata composta con LaTeX e la classe Sapthesis.

Versione: 22 novembre 2021

Email dell'autore: lepadat.1677093@studenti.uniroma1.it

*Decidere se inserire. Ne vale la pena?*

# Indice

# Sommario

# Capitolo 1

# Introduzione

## 1.1 Contesto

## 1.2 Motivazioni

## 1.3 Contributi

## 1.4 Lavori correlati

## 1.5 Struttura

# Capitolo 2

# Background

Ogni sistema cyber-fisico che si rispetti è dotato di almeno un sensore che ha il compito di misurare una determinata "qualità" fisica di interesse per il sistema stesso. I dati che vengono rilevati dai sensori spesso vengono memorizzati localmente e/o in modo remoto e possono essere impiegati, come nel lavoro qui presentato, per fini paralleli o trasversali a quelli per cui sono stati installati. Una sequenza di dati estratti da sensori ordinata temporalmente viene chiamata *serie temporale* (*time-series* in inglese).

Comunemente i sensori sono imperfetti per costruzione e trasportano intrinsecamente un'incertezza (rumore) che influenza le misurazioni da essi compiute. Sia

$$\bar{y}_k = y_k + \delta_k \tag{2.1}$$

il valore misurato da un determinato sensore nell'istante di tempo $k$, composto da $y_k$, il valore effettivo in quell'istante della grandezza misurata, più $\delta_k$, il rumore aggiunto.

In un determinato istante di tempo, il valore di ogni sensore del sistema costituisce lo *stato* del sistema. La sfida di estrarre il fingerprint dai sensori è data dal fatto che questi stati sono dinamici. Prendendo in considerazione, per esempio, un termometro, se la temperatura dell'ambiente che misura rimane costante nel tempo è facile estrarre il fingerprint del rumore e costruirne il profilo, ma in processi reali non è così semplice, gli stati cambiano continuamente, per esempio l'aumento di velocità di una macchina per via della pressione sul pedale dell'acceleratore. È importante catturare queste variazioni affinché le misurazioni dinamiche dei sensori possano essere stimate. In [1] questo problema viene affrontato definendo un modello analitico del sistema interessato, rappresentato tramite il modello *State-Space*. Vengono implementate le tecniche definite in [2], definendo così il modello lineare tempo inviariante (LTI) del sistema, rappresentato dal sistema di equazioni

$$\begin{cases} x_{k+1} = Ax_k + Bu_k + \vartheta_k, \\ y_k = Cx_k + \eta_k : \end{cases} \tag{2.2}$$

in cui $x_k \in \mathbb{R}^n$ rappresenta lo stato del sistema, $u_k \in \mathbb{R}^p$ l'input di controllo e $\vartheta_k$ il rumore al tempo $k$. $y_k \in \mathbb{R}_m$ e $\eta_k \in \mathbb{R}_m$ rappresentano, rispettivamente, la misurazione e il rumore del sensore al tempo $k$. $A$, $B$, $C$ sono le matrici dello spazio di stato di dimensioni adeguate che rappresentato la dinamica del sistema.

Definito il precedente sistema, ci sono molti punti che un attaccante mal intenzionato potrebbe bersagliare. Nel lavoro presentato, così come in [1], vengono presi in considerazione *spoofing attack* ai sensori che potrebbero essere portati a termine tramite uno schema *Man-in-The-Middle*. L'equazione lineare che rappresenta questa tipologia di attacchi è data da

$$\bar{y}_k = y_k + \delta_k = Cx_k + \eta_k + \delta_k,^1 \tag{2.3}$$

in cui $\delta_k \in \mathbb{R}_m$ rappresenta un attacco ai sensori.

In [1], dato l'output $\bar{y}_k$, viene adoperato il *filtro di Kalman* per stimare lo stato del sistema e il vettore dei *residui*, definito, in questo contesto, come la differenza tra la reale misurazine effettuata dal sensore e la stima della misurazione calcolata dal filtro nell'istante $k$:

$$r_k := \bar{y}_k - \hat{y}_k, \tag{2.4}$$

dove $\hat{y}_k$ è l'output del filtro di Kalman.

Detto ciò, per quantificare la bontà del modello del sistema, viene utilizzato l'*Errore Quadratico Medio (RMSE)*, definito come

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{n}}. \tag{2.5}$$

Questa metrica rappresenta la distanza tra il valore stimato e quello misurato, ovvero quanto il primo è lontano dal secondo. Nella letteratura della teoria del controllo, modelli con un'accuratezza superiore al 70% sono considerati accettabili approssimazioni della dinamica di sistemi reali.

Per ogni momento statistico (media, deviazione standard, ...) di una serie storica (ma non solo) si può definire un *intervallo di confidenza* che esprime la probabilità che il valore calcolato sugli $N$ campioni della serie approssimi il valore effettivo del momento statistico. Questo intevallo dipende, nel caso del valore medio, si definisce come

$$Pr\{\bar{x} - \epsilon \leq \mu \leq \bar{x} + \epsilon\} = 1 - \delta, \tag{2.6}$$

in cui $\mu$ e $\bar{x}$ sono, rispettivamente, la media effettiva e quella calcolata. $\epsilon$ e $\delta$ sono valori che dipendono da $N$, e mantenendo $\delta$ costante e incrementando $N$, anche $\epsilon$ cresce, allargando l'intervallo di confidenza. Tale intervallo di condifenza può essere definito anche per momenti di ordine superiore.

Nel contesto del presente lavoro, come si vedrà, volendo giudicare la legittimità delle misurazioni di un determinato sensore, determinate proprietà statistiche delle nuove misurazioni (nuove nel contesto di normale funzionamento del sistema aperto ad attacchi) verranno confrontate con le stesse proprietà di misurazioni effettuate in condizioni *sicure* (questi valori sono chiamati valori di *reference*). Per le nuove misurazioni, prendendo ancora in esempio il valore medio e volendo avere un intervallo di confidenza il più piccolo possibile (quindi un $\epsilon$ il più piccolo possibile), bisogna essere attenti per via di valori di $N$ non molto grandi, caratteristica preferibile in quanto non si vogliiono campionare troppi valori in situazioni real-time.

[1] Notare l'ugualianza con l'equazione 2.1: un attacco è considerato come un'introduzione di rumore nella misurazione fatta da un sensore.

| Feature | Descrizione |
|---|---|
| Media | $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ |
| Varianza | $\sigma = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2$ |
| Dev. Med. Ass. | $D_{\bar{x}} = \frac{1}{N}\sum_{i=1}^{N} \lvert x_i - \bar{x}\rvert$ |
| Asimmetria | $\gamma = \frac{1}{N}\sum_{i=1}^{N}(\frac{x_i-\bar{x}}{\sigma})^3$ |
| Curtosi | $\beta = \frac{1}{N}\sum_{i=1}^{N}(\frac{x_i-\bar{x}}{\sigma})^4 - 3$ |

**Tabella 2.1.** Lista delle feature utilizzate; $x$ è la serie temporale di dimensione $N$ proveniente dal sensore.

... bisognerà affrontare il problema avvalendosi dell'aiuto di un determintao modello di *Machine Learning*, che dà buone approssimazioni

Il vettore dei residui è quindi parte fondamentale per la definizione dei fingerprint dei sensori. A tale scopo viene definito un problema di M.L. che ha come *feature* alcuni valori statistici estratti dai vettori residui. Queste feature sono mostrate nella Tabella 2.1.

Un problema di M.L. può essere definito come una funzione $f : X \to Y$, dato un insieme $D$ (dataset) contenente informazioni riguardanti $f$. Fare il *learning* della funzione $f$ significa trovare un'altra funzione $\hat{f}$ che approssima e ritorna valori più vicini possibile ad $f$, specialmente per elementi non presenti in $D$. Nel presente lavoro il problema viene definito come un problema *supervised*[2] di *classificazione*, cioè in cui $f$ è definita tale che

$$\begin{aligned} X &:= \mathbb{R}^m, \\ Y &:= \{C_1, C_2, \ldots, C_k\}. \end{aligned} \tag{2.7}$$

Quindi $f$ associa ad ogni elemento di $X$, cioè un vettore di $m$ reali, un elemento di $Y$, cioè la classe $C_i$ di appartenenza.

[2] $D = \{(x_i, y_i)_{i=1}^N\}$

# Capitolo 3

# Metodi

# Capitolo 4

# Implementazione

# Capitolo 5

# Risultati sperimentali

## 5.1 Obiettivi

## 5.2 Configurazione (Setting) (?)

## 5.3 Casi di studio

## 5.4 Correttezza

## 5.5 Valutazione computazionale

## 5.6 Valutazione tecnica

# Capitolo 6

# Conclusioni