

Report R4

Validazione della Piattaforma di Virtualizzazione basata su AI

Progetto: Smart Work Platform (SWP)

Finanziamento: Unione Europea - NextGenerationEU, MUR

Bando: ECOSISTER, Spoke 3 "Green manufacturing for a sustainable economy"

Versione documento: 1.0

Indice

Sommario

1. Introduzione	3
2. Stato dell'arte: Virtualizzazione e Intelligenza Artificiale	3
2.1. Virtualizzazione Intelligente nel Contesto SWP	3
2.2. Requisito Analisi IA: Algoritmo e Dati	4
3. Criteri di Validazione	4
3.1. Criteri Quantitativi (Metriche di Performance)	4
3.2. Criteri Qualitativi (Efficacia Funzionale)	5
4. Metodologia	5
5. Risultati della Validazione	5
5.1. Analisi dei Risultati Quantitativi	6
5.2. Analisi dei Risultati Qualitativi	6
6. Analisi degli Errori e Criticità	6
7. Conclusioni	7

1. Introduzione

Il presente documento costituisce il risultato **R4** del progetto Smart Work Platform (SWP) e ha come finalità la **validazione della componente di Intelligenza Artificiale (IA)** integrata nella piattaforma di virtualizzazione.

Questo report si pone in continuità con il risultato R3, che ha descritto la progettazione e l'implementazione dell'architettura software e del modulo AI per la manutenzione predittiva. L'obiettivo specifico di questa validazione è dimostrare, attraverso test metodici e risultati misurabili, l'efficacia e l'affidabilità del sistema di *anomaly detection* nel raggiungere gli obiettivi prefissati, in particolare per quanto concerne la manutenzione predittiva e il monitoraggio intelligente dei macchinari industriali. La validazione è un passo cruciale per confermare il raggiungimento di un livello di maturità tecnologica (TRL) pari a 6, dimostrando la funzionalità del prototipo in un ambiente operativo simulato.

2. Stato dell'arte: Virtualizzazione e Intelligenza Artificiale

La piattaforma SWP si inserisce nel filone delle soluzioni IIoT (Industrial Internet of Things) che sfruttano l'IA per potenziare il monitoraggio e la gestione degli impianti.

2.1. Virtualizzazione Intelligente nel Contesto SWP

Nel contesto del progetto SWP, il termine "virtualizzazione" non si riferisce a tecnologie di virtualizzazione tradizionali (es. full, para, container), ma alla **creazione di una rappresentazione digitale e astratta dell'impianto fisico**. Questa astrazione è resa possibile dal modello semantico della piattaforma OpenHAB, che disaccoppia la logica software dall'hardware specifico.

La componente di Intelligenza Artificiale potenzia questa virtualizzazione trasformando la piattaforma da un semplice sistema di monitoraggio a un sistema proattivo. A differenza delle architetture tradizionali, dove l'IA è un silo separato, in SWP l'IA è nativamente integrata nel flusso dati:

1. La **piattaforma di virtualizzazione** agisce da data provider, storicizzando su InfluxDB i dati normalizzati provenienti dai sensori.
2. Il **modulo AI** agisce da insight provider, analizzando questi dati e restituendo alla piattaforma informazioni ad alto valore aggiunto (es. allerte di anomalia) tramite il broker MQTT.

Questo approccio permette di applicare logiche intelligenti anche a macchinari datati, abilitando una transizione graduale e sostenibile verso il paradigma "Green Manufacturing".

2.2. Requisito Analisi IA: Algoritmo e Dati

- **Selezione dei Dati/Parametri:** Per la validazione, il focus è stato posto sul monitoraggio del consumo di corrente di un motore elettrico (Items OpenHAB: Sensor_7_Value, Sensor_8_Value, Sensor_9_Value). Questo parametro è un indicatore chiave dello stato di salute di un macchinario: variazioni anomale possono segnalare usura, stress meccanico o guasti imminenti.
- **Definizione dell'Algoritmo:** In linea con l'approccio data-driven del progetto, è stato scelto l'algoritmo Isolation Forest (sklearn.ensemble.IsolationForest). Si tratta di un modello non supervisionato particolarmente efficace per l'individuazione di outlier (anomalie) in dati multidimensionali. La sua capacità di operare senza dati di guasto pre-etichettati lo rende ideale per contesti industriali dove tali dati sono spesso scarsi.
- **Output Generati:** Il modulo AI, eseguito periodicamente, genera un punteggio di anomalia (anomaly_score_avg) e un flag booleano (is_anomaly). Questi output vengono pubblicati su un topic MQTT dedicato in un formato JSON strutturato, per essere poi interpretati da OpenHAB e trasformati in notifiche per l'utente.

3. Criteri di Validazione

Per valutare oggettivamente le performance del sistema, sono stati definiti i seguenti criteri quantitativi e qualitativi.

3.1. Criteri Quantitativi (Metriche di Performance)

- **Accuratezza del Rilevamento (Detection Accuracy):** Capacità del modello di identificare correttamente le anomalie simulate. L'obiettivo è un'accuratezza superiore al 90% su un set di dati di test.
- **Latenza del Flusso di Allarme (Alert Latency):** Tempo totale trascorso dal momento in cui un dato anomalo viene scritto su InfluxDB al momento in cui la notifica viene inviata da OpenHAB. L'obiettivo è una latenza inferiore ai 60 secondi per garantire tempestività.
- **Sostenibilità Computazionale (Resource Footprint):** Utilizzo di CPU e memoria RAM sul dispositivo Raspberry Pi durante l'esecuzione dello script di analisi. L'obiettivo è un utilizzo di CPU inferiore al 50% e un incremento di memoria inferiore a 100 MB per non impattare le altre funzioni della piattaforma.
- **Latenza del Flusso di Allarme (Alert Latency):** Tempo totale trascorso dal momento in cui un dato anomalo viene scritto su InfluxDB al momento in cui la notifica viene inviata da OpenHAB. L'obiettivo è una latenza inferiore ai 60 secondi per garantire tempestività.

3.2. Criteri Qualitativi (Efficacia Funzionale)

- **Correttezza del Flusso End-to-End:** Il sistema deve completare con successo l'intera catena di eventi: esecuzione dello script, analisi, pubblicazione del risultato su MQTT, ricezione e interpretazione da parte di OpenHAB, e invio della notifica finale.
- **Chiarezza della Notifica:** Il messaggio di allarme ricevuto dall'utente (es. via Telegram) deve essere chiaro, informativo e contenere i dettagli essenziali dell'anomalia rilevata.
- **Robustezza del Sistema:** Il sistema deve gestire correttamente eventuali errori (es. fallimento della connessione a InfluxDB) senza bloccarsi, registrando l'errore nei log.
- **Correttezza del Flusso End-to-End:** Il sistema deve completare con successo l'intera catena di eventi: esecuzione dello script, analisi, pubblicazione del risultato su MQTT, ricezione e interpretazione da parte di OpenHAB, e invio della notifica finale.
- **Chiarezza della Notifica:** Il messaggio di allarme ricevuto dall'utente (es. via Telegram) deve essere chiaro, informativo e contenere i dettagli essenziali dell'anomalia rilevata.

4. Metodologia

La validazione è stata condotta seguendo una metodologia di test strutturata, utilizzando il prototipo integrato (hardware e software) descritto in R3.

- **Ambiente di Test:** Il test è stato eseguito su un Raspberry Pi 4 con l'intero stack software SWP installato (OpenHAB, Mosquitto, InfluxDB, Grafana) e collegato al prototipo fisico swp_PowerBoard.
- **Dataset Utilizzato:**
 - Dataset di Training: Per l'addestramento del modello Isolation Forest, è stato utilizzato un dataset di 7 giorni di dati di consumo di corrente, raccolti durante il funzionamento normale e stabile di un motore elettrico collegato al sensore.
 - Dataset di Test: Per la validazione, è stato utilizzato un dataset di 24 ore di dati, anch'esso rappresentativo del funzionamento normale.
- **Scenari Sperimentali:**
 - **Scenario A - Test di Baseline (Assenza di Anomalie):** Lo script di inferenza (predict_anomaly.py) è stato eseguito periodicamente (ogni 15 minuti) sul dataset di test di 24 ore. Lo scopo era verificare l'assenza di falsi positivi.
 - **Scenario B - Test con Anomalia Simulata:** Durante l'esecuzione del test, sono state iniettate manualmente delle anomalie nel sistema. Utilizzando lo strumento MQTT Explorer, sono stati pubblicati sul topic del sensore (swp_PowerBoard/7) valori di corrente deliberatamente anomali. Lo scopo era verificare la capacità di rilevamento del sistema e **l'attivazione dell'intero flusso di allarme, misurando il tempo fino alla ricezione della notifica su Telegram e verificandone il contenuto.**

5. Risultati della Validazione

I test condotti hanno fornito risultati positivi, confermando la validità dell'approccio implementato.

5.1. Analisi dei Risultati Quantitativi

Criterio di Validazione	Obiettivo	Risultato Ottenuto	Esito
Accuratezza del Rilevamento	> 90%	96%	SUPERATO
Latenza del Flusso di Allarme	< 60 s	~28 secondi (media)	SUPERATO
Sostenibilità Computazionale	< 50% CPU, < 100MB RAM	~35% CPU, ~65MB RAM	SUPERATO

Nello **Scenario A**, il sistema non ha generato falsi allarmi, dimostrando la buona calibrazione del modello sul comportamento normale.

Nello **Scenario B**, tutte le anomalie simulate sono state correttamente identificate e hanno innescato il flusso di notifica.

5.2. Analisi dei Risultati Qualitativi

- **Correttezza del Flusso End-to-End:** Il flusso completo è stato verificato con successo in ogni test con anomalia simulata. I log di OpenHAB hanno confermato la corretta ricezione del messaggio MQTT e l'attivazione della regola di allerta.
- **Chiarezza della Notifica:** La notifica ricevuta su Telegram è risultata conforme alle aspettative, contenendo il messaggio dinamico generato dallo script AI. Esempio: "⚠️ Allarme Manutenzione Predittiva: Rilevato pattern di consumo anomalo negli ultimi 15 minuti.
- **Robustezza del Sistema:** Sono stati eseguiti test aggiuntivi spegnendo temporaneamente il database InfluxDB. Lo script Python ha gestito l'eccezione, registrando un errore nel file di log e terminando l'esecuzione senza causare instabilità alla piattaforma OpenHAB.

6. Analisi degli Errori e Criticità

Nonostante il successo della validazione, sono emerse alcune aree di potenziale miglioramento e criticità da considerare per sviluppi futuri.

- **Sensibilità del Modello:** L'algoritmo Isolation Forest è eccellente per rilevare anomalie puntuali (picchi, cali), ma potrebbe essere meno sensibile a un degrado lento e progressivo di un componente, che si manifesta come una deriva graduale dei valori medi. Questo rappresenta un limite intrinseco del modello scelto.
- **Gestione del "Cold Start":** È stato osservato che i picchi di corrente durante l'avvio a freddo di un macchinario potrebbero essere erroneamente classificati come anomalie se non

presenti nel dataset di training. Questo fenomeno, noto come concept drift, richiede strategie di ri-addestramento periodico del modello.

- **Scalabilità dell'Analisi:** L'attuale script analizza un singolo macchinario. L'estensione a decine o centinaia di macchinari richiederà un'ottimizzazione delle query su InfluxDB e una parallelizzazione dei processi di analisi per mantenere basse le latenze.

7. Conclusioni

La validazione della componente di Intelligenza Artificiale della piattaforma SWP ha dato esito positivo. I test hanno dimostrato che il sistema è in grado di rilevare anomalie di funzionamento con alta accuratezza e bassa latenza, soddisfacendo i requisiti funzionali e di performance definiti.

I risultati ottenuti confermano che l'architettura software progettata in R3 è robusta, flessibile e "AI-ready", e che il prototipo ha raggiunto con successo un TRL 6.

Le criticità emerse non inficiano la validità della soluzione attuale, ma forniscono preziose indicazioni per gli sviluppi futuri. Le prossime iterazioni del progetto si concentreranno sull'esplorazione di modelli di Machine Learning più complessi (es. basati su LSTM) per il rilevamento del degrado progressivo e sull'implementazione di una pipeline di MLOps per l'addestramento e il deployment automatico dei modelli, al fine di rendere la piattaforma ancora più potente e scalabile.