Anomaly detection

Leonardo Ganzaroli 1961846

Ottobre 2024

Indice

1 Descrizione generale

Obiettivo: Il sistema da sviluppare ha lo scopo di rilevare anomalie in stream di dati.

Esecuzione: Il rilevamento avviene basandosi su:

- Media di ogni stream
- Covarianze tra i diversi stream

Un'ulteriore richiesta è che i calcoli vengano effettuati su una finestra temporale.

Funzioni aggiuntive: Deve generare degli avvisi in caso di valori ritenuti anomali, ovvero valori che differiscono in modo considerevole dai precedenti.

Schema generale:

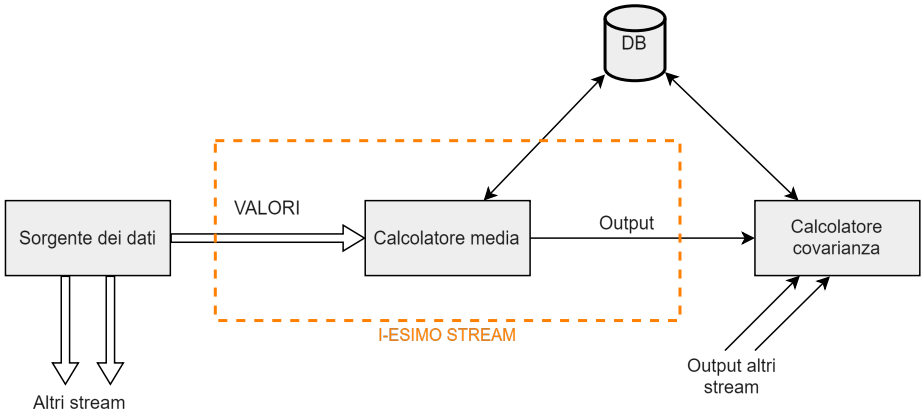

Figura 1: Schema generale del sistema

2 Requisiti Utente

Lista dei requisiti utente:

2.1 Funzionali

1. **Scelta dei valori di soglia**:
   1.1. L'utente deve poter configurare i valori di soglia per la rilevazione delle anomalie del valor medio.
   1.2. L'utente deve poter configurare i valori di soglia per la rilevazione delle anomalie della covarianza.
2. **Finestra temporale**:
   2.1. L'utente deve avere la possibilità di scegliere l'ampiezza della finestra usata per i calcoli.
3. **Avvisi anomalie**:
   3.1. L'utente deve poter leggere gli avvisi generati dal sistema riguardo le anomalie.

Si possono visualizzare questi requisiti sottoforma di Use Case nel diagramma di seguito.

Diagramma Use Case:

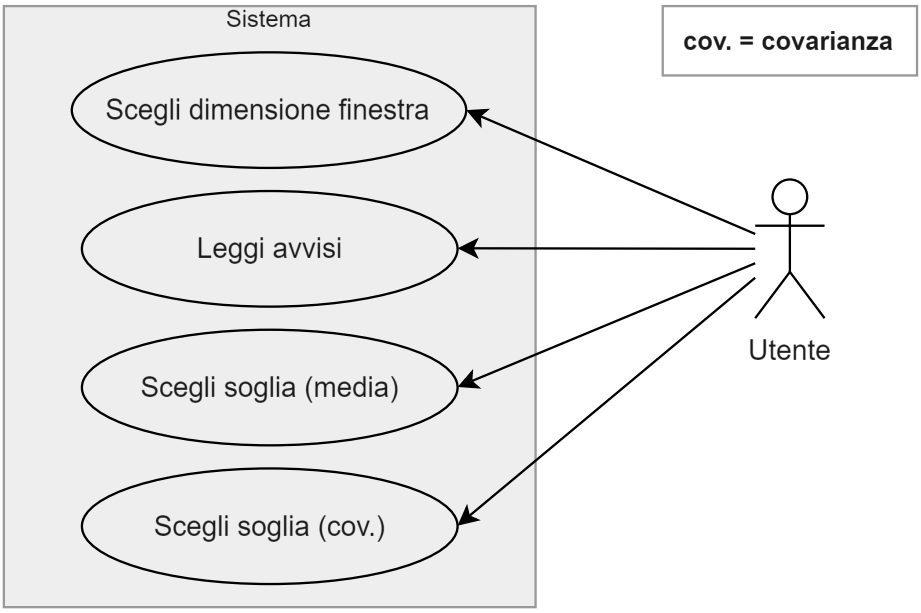

Figura 2: Use Case

3 Requisiti di Sistema

Lista dei requisiti di sistema:

3.1 Funzionali

1. **Lettura dati**:
   1.1. Il sistema deve essere in grado di gestire l'arrivo dei dati in tempo reale in base alla capacità della finestra.
2. **Calcoli statistici**:
   2.1. Il sistema deve calcolare il valor medio della finestra associata ad ogni singolo stream e salvarlo sul DB.
   2.2. Il sistema deve calcolare le covarianze tra le finestre associate ai diversi stream e salvarle sul DB.
3. **Determinazione anomalie**:
   3.1. Il sistema deve rilevare anomalie nei dati basandosi sulle soglie impostate dall'utente.
4. **Generazione avvisi**:
   4.1. Quando il sistema rileva un'anomalia deve generare un log, stamparlo a schermo e salvarlo sul DB.
5. **Archiviazione dati**:
   5.1. Il sistema deve salvare i valori su un database *PostgreSQL*.
6. **Input utente**:
   6.1. Il sistema deve ricevere e gestire gli input definiti nei requisiti utente.

3.2 Non Funzionali

1. **Tempo di invio degli avvisi**:
   1.1. L'invio dell'avviso riguardo il valor medio deve avvenire entro mezzo secondo dalla rilevazione dell'anomalia.
   1.2. L'invio dell'avviso riguardo la covarianza deve avvenire entro un secondo dalla rilevazione dell'anomalia.

4 Struttura

Si illustra adesso la struttura del sistema più nello specifico.

4.1 Componenti

I componenti del sistema sono 4:

- Test Generator
- Calcolatore media
- Calcolatore covarianza
- Monitor

Segue una breve descrizione.

4.1.1 Test Generator

Cosa fa: Invia i dati sugli stream.
Come lo fa: Legge una riga per volta da un file CSV ed invia i dati ai rispettivi stream.

Esempio:

Se il file ha questo formato:

Colonna 1	Colonna 2	Colonna 3	Colonna 4
12	2	0	1
32	3	88	111

Tabella 1: Esempio file

La prima riga letta sarà inviata sugli stream in questo modo:

- STREAM1 $\leftarrow$ 12
- STREAM2 $\leftarrow$ 2
- STREAM3 $\leftarrow$ 0
- STREAM4 $\leftarrow$ 1

4.1.2 Calcolatore media

Cosa fa: Calcola la media di ogni stream.

Come lo fa: Gestisce i valori in entrata tramite una coda (finestra = coda), ogni stream ha la sua coda.

Altre funzioni: Salva tutte le medie calcolate nel DB, invia gli elementi di ogni coda e la relativa media al Calcolatore della covarianza.

4.1.3 Calcolatore covarianza

Cosa fa: Calcola la covarianza tra le coppie di stream.

Come lo fa: Usa le finestre e le medie ricevute dal Calcolatore della media.

Altre funzioni: Salva tutte le covarianze calcolate nel DB.

4.1.4 Monitor

Cosa fanno: Effettuano dei controlli sul sistema, controllano che i requisiti richiesti vengano rispettati.

In questo caso ne sono stati implementati 5:

- Corretto inserimento della media
- Corretto inserimento degli avvisi riguardanti la media
- Corretto inserimento della covarianza
- Tempo trascorso dal riscontro dell'anomalia all'invio dell'avviso (media)
- Tempo trascorso dal riscontro dell'anomalia all'invio dell'avviso (covarianza)

4.2 Schemi

I singoli componenti comunicano e collaborano per svolgere i compiti richiesti, a seguire degli schemi che illustrano l'architettura ed il comportamento del sistema.

Singoli componenti e loro collegamenti:

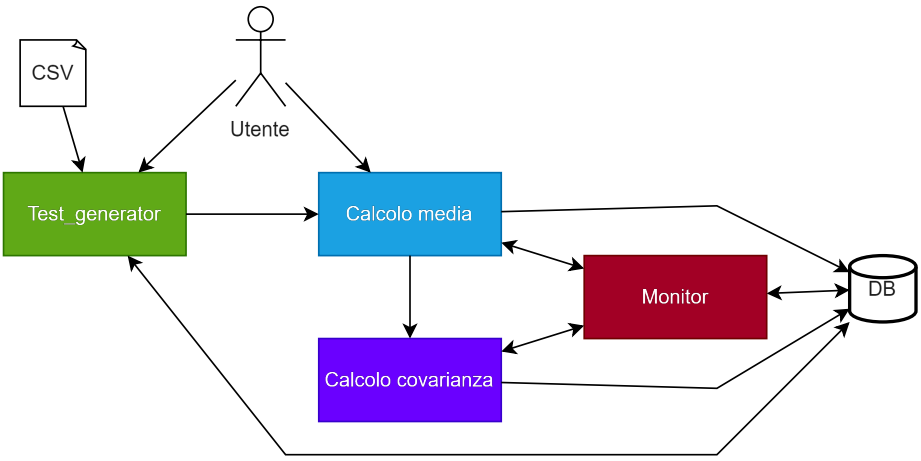

Figura 3: Componenti del sistema

L'utente fornisce (come visto negli Use Case) le soglie per gli avvisi e la grandezza della finestra, inoltre ha la possibilità di scegliere il file CSV utilizzato.

Anche se i componenti sono indipendenti, è presente una certa linearità nelle operazioni, come si può notare dal seguente diagramma delle attività:

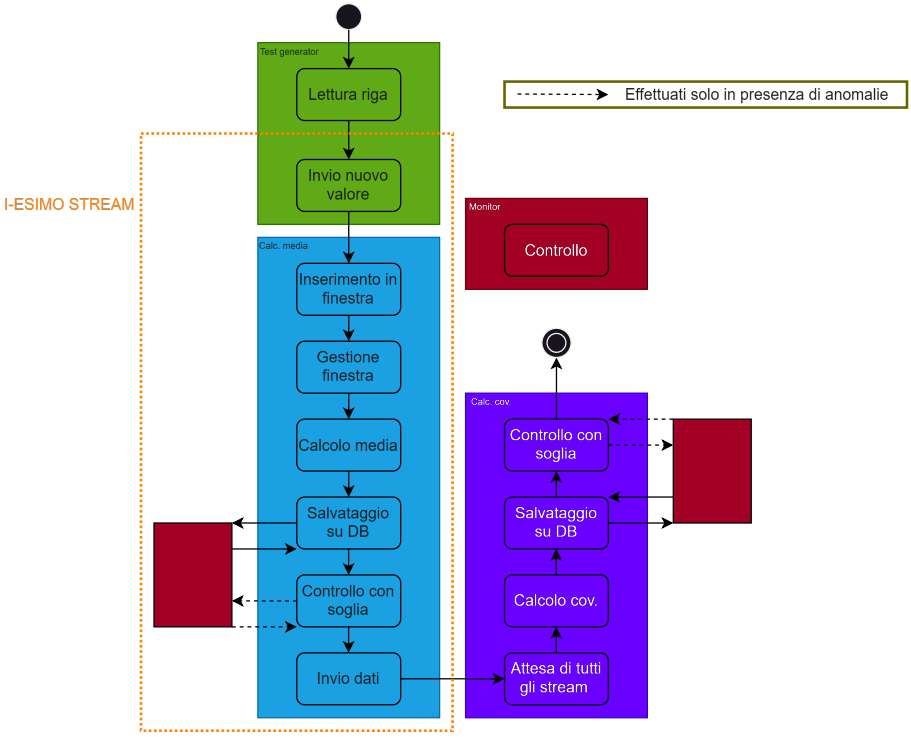

Figura 4: Activity diagram

Il funzionamento dei monitor si può modellare come una macchina a stati, per esempio il monitor che controlla il corretto inserimento della media si può rappresentare come segue:

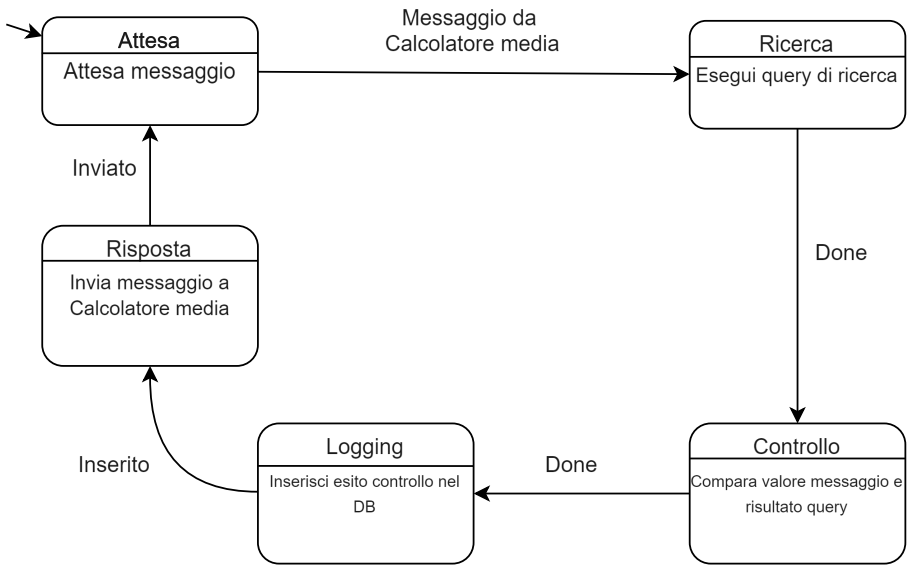

Figura 5: State Diagram Monitor

I diagrammi degli altri monitor risultano pressoché identici.

Ad operazioni già avviate (finestre riempite e tutti gli input dell'utente ricevuti ed elaborati) lo scambio di messaggi tra i vari componenti si può riassumere in questo modo:

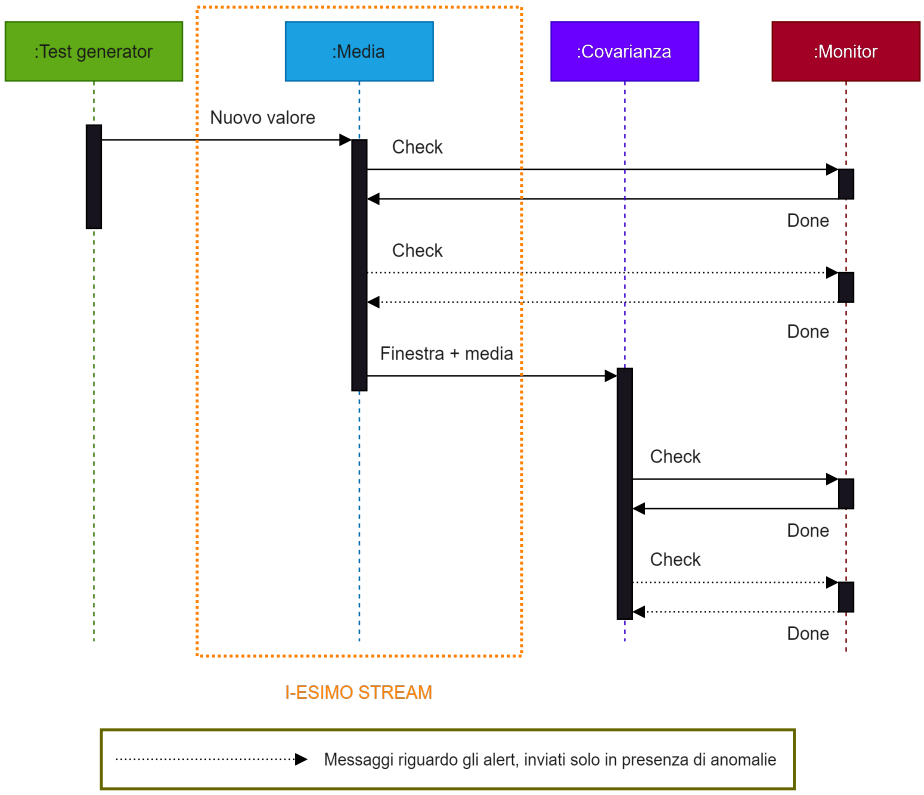

Figura 6: Message Sequence Chart

5 Implementazione

Questa sezione contiene le implementazioni dei componenti in pseudocodice e lo schema del database utilizzato.

5.1 Componenti

5.1.1 Test generator

$nomefile \leftarrow$INPUT UTENTE
$file \leftarrow apri$(nome file)
Genera stream pari a $file.colonne$
Invia messaggio con numero di stream
while file ha nuova riga da leggere **do**
    Leggi nuova riga
    Invia ogni valore della riga al proprio stream
end while

5.1.2 Calcolatore media

$dimfinestra \leftarrow$ INPUT UTENTE
$soglia \leftarrow$ INPUT UTENTE
Leggi messaggio
$finestre \leftarrow$ Array di code[valore messaggio]
while Messaggi da leggere disponibili **do**
    Riempi ogni coda fino a $dimfinestra$ da rispettivo stream
    Calcola media di ogni stream
    for media calcolata in questa iterazione **do**
        Salva media su DB
        Chiama monitor
        Aspetta risposta
        if media diversa da precedente almeno di $soglia$ **then**
            Invia avviso
            Chiama monitor normale e temporale
            Aspetta entrambe le risposte
        end if
    end for
    Invia ogni finestra + sua media a Calcolatore covarianza
    Rimuovi da ogni coda elemento più vecchio
end while

5.1.3 Calcolatore covarianza

soglia ← INPUT UTENTE
Leggi messaggio
while Messaggi da leggere disponibili **do**
    Prendi primo messaggio non letto per ogni stream
    Calcola covarianza tra tutti i valori dei messaggi (a coppie)
    for covarianza calcolata in questa iterazione **do**
        Salva covarianza su DB
        Chiama monitor
        Aspetta risposta
        if covarianza diversa da precedente almeno di *soglia* **then**
            Invia avviso
            Chiama monitor normale e temporale
            Aspetta entrambe le risposte
        end if
    end for
end while

5.1.4 Monitor

while 1 **do**
    Aspetta messaggio
    Leggi messaggio
    Prendi da DB ultimo valore per quello stream
    Confronta risultato e messaggio
    Salva su DB l'esito del controllo
    Invia messaggio
end while

La struttura è identica per i monitor dello stesso tipo.
La struttura dei monitor temporali è invece:

while 1 **do**
    Aspetta messaggio
    Leggi messaggio
    Prendi da DB tempo d'inserimento valore per quello stream
    Prendi da DB tempo d'inserimento avviso per quello stream
    Calcola tempo trascorso e controlla se è nei limiti
    Salva su DB tempo + esito del controllo
    Invia messaggio
end while

5.2 Database

Il database contiene 5 tabelle:

- **Session_info:** mantiene informazioni sulle singole sessioni
- **Media e Covarianza:** contengono i rispettivi valori
- **Alerts:** contiene le anomalie riscontrate
- **Log_monitor:** contiene gli esiti dei controlli effettuati dai monitor

Le chiavi delle tabelle sono rappresentate in rosso.

ID	Data_Inizio	Nome_file	Numero_colonne

Tabella 2: Session_info

Tutti i campi S_id delle prossime tabelle si riferiscono a ID di Session_info.

S_id	Data_ora	Nome_stream	Valore

Tabella 3: Media

S_id	Data_ora	Nome_stream1	Nome_stream2	Valore

Tabella 4: Covarianza

ID	S_id	Data_evento	Nome_stream	Tipo_anomalia	Valore

Tabella 5: Alerts

ID	S_id	Data_controllo	Nome_stream	Tipo_controllo	Esito	Tempo_trascorso

Tabella 6: Log_monitor

5.3 Stream Utilizzati

Per la comunicazione sono stati usati diversi stream di *Redis*.

Una lista esaustiva:

- "INFOSTREAM" usato per comunicare informazioni di servizio
- "STREAM*" con $* = [0, 1, ..., numero\_colonne\_file - 1]$ per inviare i valori dal Test generator al Calcolatore della media
- "STREAM_*" con $* = [0, 1, ..., numero\_colonne\_file - 1]$ per inviare i valori dal Calcolatore della media al Calcolatore della covarianza
- "TMonitor", "AMonitor", "CMonitor", "CTMonitor", "MMonitor" usati per inviare messaggi ai rispettivi monitor
- "M*" con $* = [1, 2, 3, 4, 5]$ usati dai monitor per inviare risposte

6 Risultati Sperimentali

Il sistema è stato testato con diversi file CSV, per ogni sessione i risultati ottenuti sono stati in linea con le aspettative meno qualche approssimazione. Inoltre il numero di elementi inseriti nel DB è stato esattamente quello atteso sia come numero di valori che come numero di log generati.