Sistema di monitoraggio INtelligente per la Sicurezza delle InfraStrutture urbane

Codice Progetto: ARS01_00913 (CUP E64E18000110005)

Durata del progetto: n. 42 mesi. Dal 01/05/2019 al 31/10/2022

DELIVERABLE

OR 2

Partners

- Università di Napoli Federico II (Soggetto Capofila)
- CNR ITC (socio di STRESS scarl)
- Consorzio Integra (socio di STRESS scarl)
- IOTSAFE
- MAPEI
- Politecnico di Torino
- STMicroelectronics
- STRESS scarl
- Tecno In (socio di STRESS scarl)
- Tecnosistem (socio di STRESS scarl)
- Università di Bergamo
- Università di Bologna
- Università di Palermo
- Università di Roma Tor Vergata

Informazioni del Deliverable		
Numero documento	*D.2*	
Diffusione documento	*Privato*	
Titolo del documento	Sistemi cognitivi per l'analisi dei dati	
Obiettivo Realizzativo	*OR2*	
Coordinatore OR	*STMicroelectronics*	
Stato del documento		
Verificato (Coordinatori OR)	*08/12/2022*	*Ing. Francesco Pappalardo* *Ing. Agatino Pennisi*
Approvato (Coordinatore Progetto)	*21/12/2022*	*Prof. Ing. Edoardo Cosenza*

PREMESSA

Il presente documento costituisce il prodotto atteso dell'OR2.

L'obiettivo realizzativo in esame si propone di ricercare e sviluppare nuovi algoritmi e sistemi intelligenti per estrarre in maniera efficace informazioni di rilievo dall'immensa quantità di dati raccolta dai sensori installati nelle strutture monitorate.

Lo scopo è facilitare il lavoro di ingegneri civili e strutturali, utilizzando tecniche avanzate in grado di processare in tempo reale quantità di dati estremamente elevate, portando alla luce informazioni e rilievi non facilmente individuabili, così che l'attenzione dell'ingegnere si possa focalizzare sugli aspetti più critici identificati dal sistema di monitoraggio.

Questo compito è svolto combinando raffinate tecniche di Big Data Analytics e Machine Learning con avanzati modelli matematici e simulazioni numeriche, il tutto combinato in un sistema cognitivo disegnato specificamente per il problema sotto analisi, ovvero quello del monitoraggio di grandi opere di ingegneria civile e industriale nel tempo.

I partner che hanno contributo alla realizzazione del deliverable sono stati: STMicroelectronics, IoTty e UniNA.

Tutte le informazioni e le analisi contenute nel documento hanno finalità di ricerca e la loro citazione può essere ricondotta al Progetto INSIST - Sistema di monitoraggio INtelligente per la Sicurezza delle InfraStrutture urbane (ARS01_00913).

Le opinioni e le conclusioni presentate non rispecchiano necessariamente quelle degli enti finanziatori.

SOMMARIO

1 DESCRIZIONE DELL'OBIETTIVO REALIZZATIVO E DELLE ATTIVITÀ

2 Progettazione, realizzazione e test di un sistema intelligente di analisi dei dati dei sensori capacitivi STMicroelectronics

2.1 Descrizione dell'attività: architettura di sistema monosensore

2.1.1 Structural Health Monitoring (SHM)

L'algoritmo di intelligenza artificiale sviluppato rientra nel quadro dello *Structural Health Monitoring* (SHM), una tecnica emergente che combina avanzate tecnologie di sensori con algoritmi intelligenti per inferire le condizioni di "salute" delle strutture, garantendo migliore affidabilità e sicurezza. Ciò richiede la disponibilità di sensori che possano essere inseriti nelle strutture e monitorati nel tempo, con l'obiettivo di consentire di valutare e monitorare in tempo reale lo stato di salute delle strutture in cui sono inseriti, riconoscendo automaticamente variazioni avverse nella loro integrità.

2.1.2 Predictive Maintenance e Anomaly Detection

La *predictive maintenance* è una tecnica nata con Industria 4.0, che mira a prevedere e anticipare possibili guasti ai macchinari, per avviare il processo di manutenzione prima che questi si verifichino. Nell'ambito del monitoraggio strutturale, l'applicazione della predictive maintenance ha l'obiettivo di consentire il monitoraggio delle condizioni delle strutture, per garantire che possano svolgere le loro funzioni e avvisare in caso di malfunzionamenti. La predictive maintenance dipende principalmente dall'*anomaly detection*, che è un insieme di tecniche che mira a identificare pattern anomali – che si discostano dal comportamento normale previsto – all'interno dei dati. Tipicamente, infatti, un sistema di monitoraggio strutturale include sempre un componente di diagnostica di basso livello, che fornisce informazioni che consentono di rilevare, localizzare e valutare eventuali danni, riducendo così tempi e costi di ispezione durante la manutenzione.

2.1.3 Architettura sistema monosensore

L'approccio comune del componente di diagnostica consiste nell'elaborare i dati del sensore utilizzando metodi di *pattern recognition*. Un sottoinsieme dei dati viene in genere utilizzato per allenare un modello che sia in grado di apprendere correttamente il comportamento della struttura in esame e prevederne la risposta. Tra i vari modelli descritti in letteratura, abbiamo scelto di utilizzare l'*autoencoder*, una rete neurale ampiamente utilizzata nella *computer vision*, ma recentemente anche nell'anomaly detection.

In figura 2.1.1 si riporta l'architettura del sistema di monitoraggio intelligente a livello di singolo nodo di acquisizione.

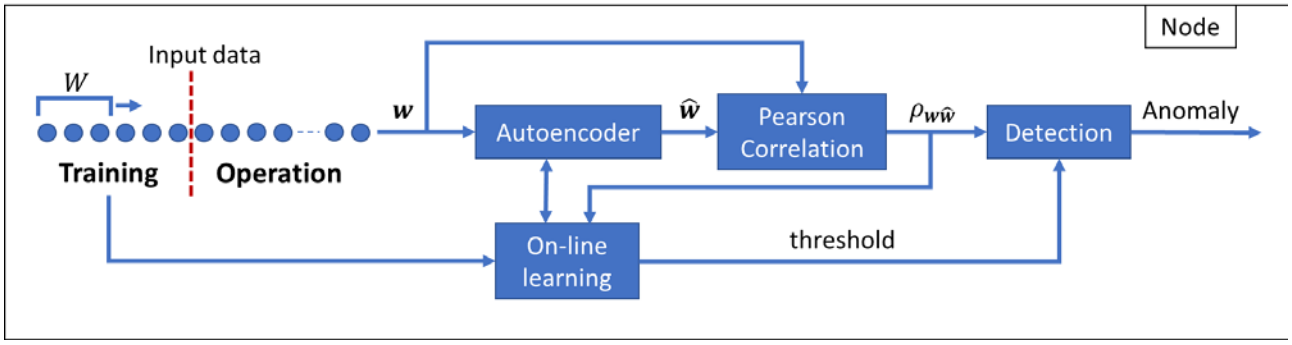


Figura 2.1.1 Architettura del sistema di monitoraggio intelligente per singolo nodo di acquisizione.

Gli elementi chiave dell'architettura, descritti in dettaglio di seguito, sono:

- Un modello di ricostruzione (*autoencoder*)
- Una fase di apprendimento (*training*), che ha lo scopo di individuare un modello che possa approssimare al meglio il comportamento del sistema considerato. L'apprendimento avviene "online" (*on-line learning*), cioè mentre i dati vengono acquisiti dal sensore.
- Una fase operativa (*operation*) di rilevazione delle anomalie, che si basa sul calcolo della correlazione tra ingresso e uscita dell'autoencoder, e permette di avere un'indicazione di quanto il comportamento del sistema si discosti da quello appreso durante la fase di apprendimento. In realtà, non solo viene indicato se è presente un'anomalia, ma, per ogni campione acquisito dal sensore e da valutare, viene generato un *anomaly score*, ovvero un valore percentuale che ci permette di sapere, istante per istante, qual è l'entità dell'anomalia rilevata nel sistema in esame.

2.1.4 L'autoencoder

Un autoencoder è un tipo di rete neurale utilizzata per apprendere efficacemente le rappresentazioni/codifiche di un insieme di dati in modo non supervisionato, ovvero senza la necessità di un'etichettatura manuale dei dati. Nel nostro caso, questo significa che non abbiamo

bisogno – ma semplicemente potremmo non essere in grado – di etichettare ogni campione temporale come normale o anomalo.

Un autoencoder è composto da due parti:

- un *encoder*
- un *decoder*

L'encoder mappa i dati che riceve in ingresso in uno spazio delle *feature* più piccolo (spazio delle feature di input), essenzialmente comprimendo i dati che riceve in ingresso. Il decoder, invece, cerca di ricostruire i dati originari partendo dallo spazio appreso (detto anche "spazio latente"), che deve quindi conservare tutte e solo le informazioni necessarie alla ricostruzione dei dati. I parametri di queste due sottoreti vengono appresi attraverso una *loss function* di ricostruzione, che obbliga il modello a conservare le informazioni essenziali per ricostruire i dati. Infatti, per ridurre al minimo gli errori di ricostruzione, il modello deve apprendere regolarità importanti, schemi ricorrenti nei dati su cui viene allenato. Il presupposto su cui si basa la sua scelta come modello di anomaly detection è che, se addestrato su dati "normali", cioè dati che contengono schemi di funzionamento normali, l'autoencoder imparerà a riprodurre fedelmente le caratteristiche di input più frequentemente osservate, mentre le sue prestazioni di ricostruzione peggioreranno quando incontrerà anomalie. Queste, infatti, sono difficilmente ricostruibili dalle rappresentazioni risultanti, e quindi generano grossi errori di ricostruzione, a differenza delle istanze normali, che possono essere meglio ricostruite dallo spazio compresso appreso.

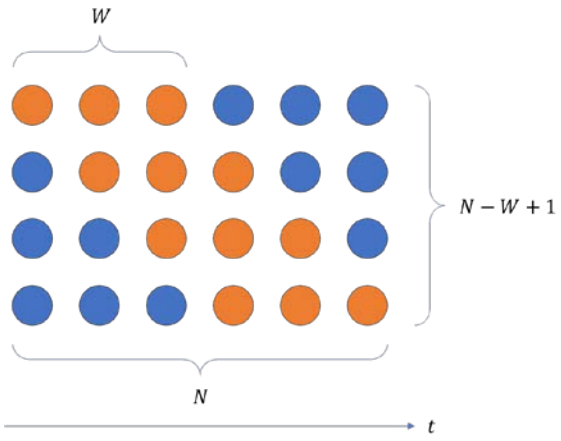


Figura 2.1.2 *Sliding window* utilizzate per l'autoencoder.

In realtà, l'autoencoder non è un modello sequenziale, cioè non è nato per modellare serie temporali, che sono i dati acquisiti dai nostri sensori. Per questo motivo si è scelto di utilizzare il meccanismo delle *sliding window* (figura 2.1.2). Si tratta di una tecnica che prevede la suddivisione degli N campioni acquisiti nel tempo da un sensore in $N - W + 1$ gruppi/finestre di W elementi, con la

finestra di osservazione che scorre un passo alla volta sui dati, come in un buffer circolare. Le finestre ottenute attraverso questo meccanismo sono mostrate in arancione nella rappresentazione sottostante.

In particolare, definito x_t il valore acquisito dal sensore al tempo t, e W la dimensione della finestra scorrevole da applicare sui dati, l'ingresso dell'autoencoder al tempo t sarà costituito dal vettore $\boldsymbol{w} = [x_{t-W+1} \quad \dots \quad x_t]$, mentre la sua uscita sarà il vettore $\hat{\boldsymbol{w}} = [\hat{x}_{t-W+1} \quad \dots \quad \hat{x}_t]$.

Di seguito è mostrato un esempio di un'architettura di autoencoder *fully connected*.

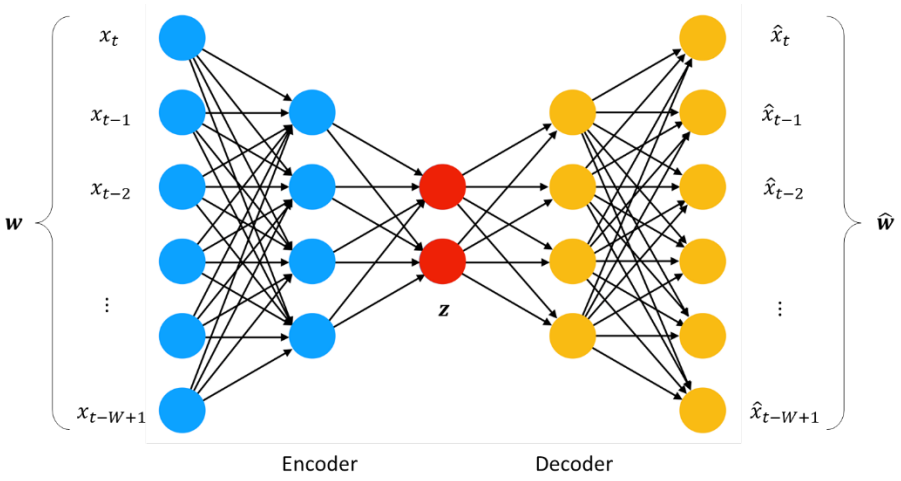


Figura 2.1.3 Schema di un autoencoder *fully connected*.

2.1.5 La fase di apprendimento

L'obiettivo della fase di apprendimento è riconoscere il modello della struttura in questione, che sarà rappresentato dai valori dei parametri della rete neurale, che di fatto definiranno la relazione input-output del sistema. Questa fase di apprendimento presuppone la disponibilità di una sequenza iniziale di N campioni acquisiti dal sensore, in cui non è presente alcuna anomalia. È grazie a questa sequenza iniziale che la rete neurale sarà in grado di riconoscere i normali schemi di funzionamento della struttura e, successivamente, rilevare anomalie. Come detto prima, questa sequenza di N campioni viene suddivisa in finestre di dimensione pari a W, in modo da ottenere $N - W + 1$ finestre di training, che verranno date in input all'autoencoder, che imparerà a ricostruirle. Oltre ad un *training set*, viene creato anche un *validation set*, costituito dai successivi M campioni acquisiti dai sensori. Questo ulteriore insieme viene utilizzato per eseguire il *tuning* degli iperparametri della rete neurale, ovvero per scegliere quei parametri che influenzano l'allenamento e il comportamento della rete neurale, e per effettuare la model selection, ovvero selezionare alla fine di ogni epoca di apprendimento il modello che è stato in grado di generalizzare meglio il comportamento della struttura in esame. Al termine della fase di apprendimento viene inoltre definita una soglia, basata sulla correlazione tra le finestre di allenamento e le rispettive ricostruzioni.

La correlazione tra le finestre viene calcolata attraverso l'indice di correlazione di Pearson, definito come:

$$-1 \leq \rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y} = \frac{\sum_{i=1}^{n}(x_i - \mu_x)\left(y_i - \mu_y\right)}{\sqrt{\sum_{i=1}^{n}(x_i - \mu_x\ )^2}\sqrt{\sum_{i=1}^{n}(y_i - \mu_y)^2}} \leq 1$$

dove σ_{xy} è la covarianza tra $\boldsymbol{x}$ e $\boldsymbol{y}$, mentre σ_x e σ_y sono le deviazioni standard di $\boldsymbol{x}$ e $\boldsymbol{y}$, rispettivamente.

2.1.6 La fase operativa

L'obiettivo della fase operativa è rilevare l'eventuale presenza di anomalie nei campioni acquisiti dai sensori. Per fare questo ci affidiamo alle correlazioni tra gli ingressi e relative ricostruzioni, che ci danno un'indicazione di quanto bene l'autoencoder riesca a ricostruire gli ingressi, cioè quanto i dati acquisiti dal sensore si "adattano" al modello della struttura appresa dall'autoencoder. Data la soglia definita al termine della fase di apprendimento, per definire il grado di anomalia associato ad un dato campione (ovvero il anomaly score), vengono prese in considerazione tutte le finestre in cui tale campione ricade. Chiamata W la dimensione delle finestre, ogni campione può rientrare in W finestre successive. Nella figura 2.1.4 sono evidenziate in arancione le finestre che concorrono a determinare l'anomalia di un campione acquisito all'istante di tempo t_i.

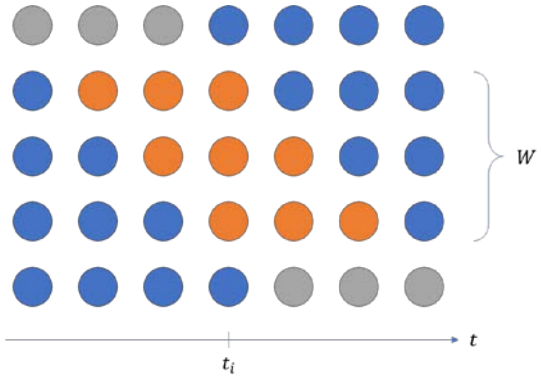


Figura 2.1.4 Sliding window utilizzate nella fase operativa.

2.1.7 Analisi sperimentale

Il lavoro fatto da STMicroelectronics è quello di verificare la validità del sistema descritto in precedenza sui dati dei sensori capacitivi acquisiti durante le prove di carico eseguite presso UniPA (durante i test sui provini in muratura) e durante le simulazioni del terremoto dell'Aquila effettuate presso UniNA (sul telaio in calcestruzzo con tamponatura in mattoni forati).

In dettaglio, si è deciso di adottare finestre scorrevoli di dimensioni W pari a 64 per il dataset di Palermo e a 128 per il dataset di Napoli, poiché la frequenza di campionamento nel dataset di Napoli è maggiore di quella nel dataset di Palermo. L'architettura dell'autoencoder, con i dettagli sulle dimensioni (nel formato $C \times L$, dove C indica il numero di canali e L la dimensione temporale) degli input e degli output di ogni *layer* della rete neurale, assumendo $W = 64$, è mostrata nella figura 2.1.5.

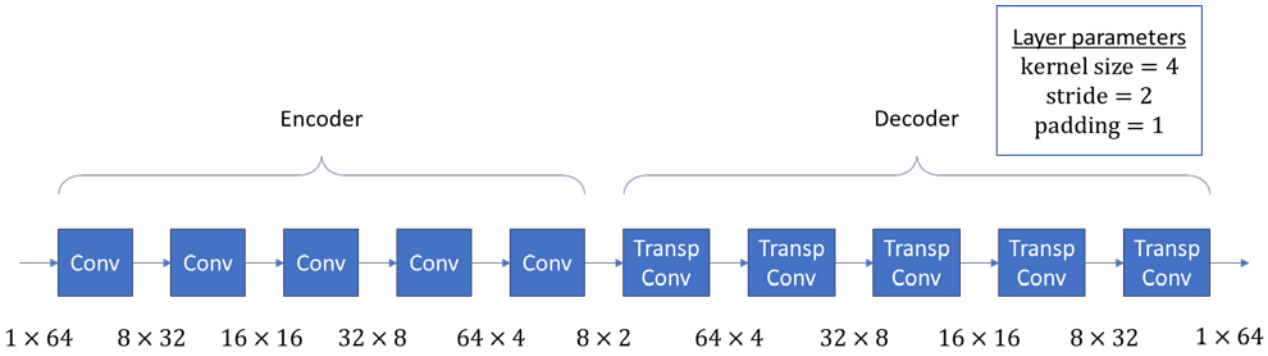


Figura 2.1.5 Analisi sperimentale: progettazione dell'autoencoder.

L'architettura è quella classica di un autoencoder convoluzionale, in cui encoder e decoder hanno una struttura speculare: mentre ogni *layer* convoluzionale dell'encoder riduce la dimensione spaziale dell'ingresso e aumenta il numero di canali (che corrispondono ai filtri convoluzionali applicati all'input), ogni layer convoluzionale trasposto del decoder esegue il procedimento inverso, cioè aumenta la dimensione spaziale dell'ingresso e riduce il numero di canali, restituendo in uscita la ricostruzione della finestra che l'autoencoder ha ricevuto in ingresso.

Ad eccezione dell'ultimo layer della rete, ogni layer convoluzionale e convoluzionale trasposto è seguito dalla funzione di attivazione non lineare relu, definita come:

$$\mathrm{relu}(x) = \max(0, x) = \begin{cases} x & x > 0 \\ 0 & \text{altrimenti} \end{cases}$$

2.2 Risultati attività

2.2.1 Risultati sul dataset di Palermo

L'analisi dei dati acquisiti presso l'Università degli Studi di Palermo (UniPA) ha riguardato le prove effettuate su provini con sensori capacitivi preinstallati nei laboratori del Dipartimento di Ingegneria Civile, Ambientale, Aerospaziale e dei Materiali (DICAM). Nella figura 2.2.1 sono riportati gli schemi dei provini con i sensori installati.

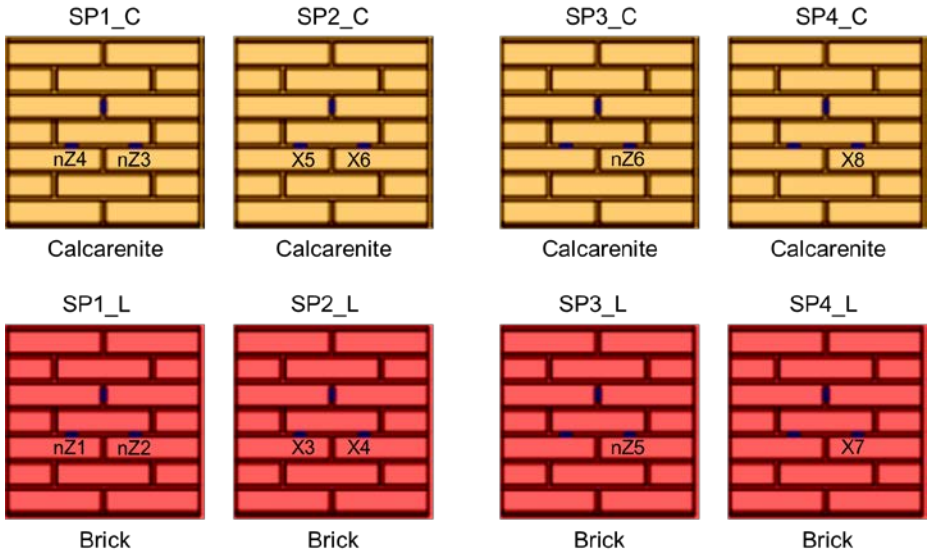


Figura 2.2.1 Schema dei sensori STMicroelectronics installati nei provini in muratura presso UniPA.

In ogni fase di test, le letture dei sensori acquisite dai nostri sistemi e da UniPA sono state memorizzate come file di testo e/o CSV. Questi file, opportunamente denominati per tenere traccia dei provini e dei sensori, sono stati quindi pre-elaborati, in modo tale che tutte le letture rappresentative dell'andamento nel tempo delle risposte dei sensori capacitivi (espressi in pF) e del carico applicato dalla pressa (espresso in kN) fossero allineate alla stessa base dei tempi. Il risultato è stato quindi utilizzato per eseguire l'analisi descritta nella sezione seguente.

In ogni prova di compressione si possono distinguere tre fasi:

- due fasi di precarico
- una fase di carico fino a rottura

Data questa configurazione, abbiamo deciso di allenare il nostro autoencoder sulla prima fase di precarico, che rappresenta per noi una normale fase operativa, validarlo sui pochi campioni successivi, per poi testarlo sulla seconda fase di precarico e sulla fase di carico fino a rottura.

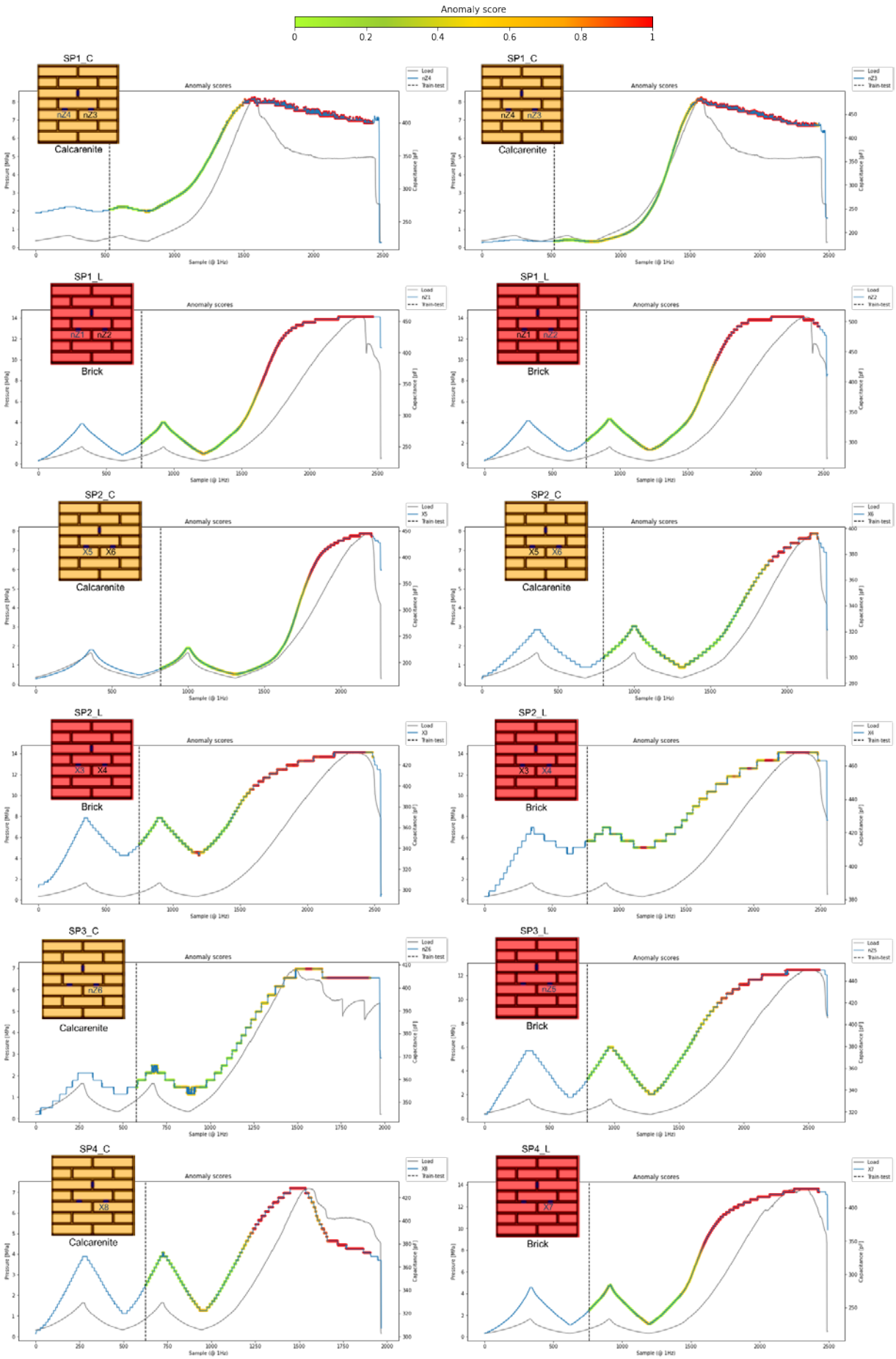


Figura 2.2.2 Risultati delle analisi sul dataset dei test eseguiti presso UniPA.

Ci aspettavamo che l'autoencoder riconoscesse come normale la seconda fase di precarico, assegnandole quindi un anomaly score basso, mentre l'anomaly score aumentasse nella fase di carico, fino a quando non fosse stata rilevata la rottura, e da quel momento in poi assegnasse un anomaly score molto alto (presumibilmente pari a 1). Le figure seguenti evidenziano gli anomaly score assegnati dall'autoencoder a ciascun campione acquisito. Si noti che gli ultimi $W - 1$ campioni non vengono esaminati, poiché non possono rientrare in W finestre successive.

Come si può notare, i risultati sembrano abbastanza buoni (figura 2.2.2), in quanto il modello è in grado di assegnare correttamente anomaly score più elevati ai campioni prima e dopo il punto di rottura, mentre, salvo alcuni campioni discutibili, assegna anomaly score molto bassi ai campioni nella seconda fase di precarico.

2.2.2 Risultati sul dataset di Napoli

È stata testata la validità dell'algoritmo su un altro dataset acquisito durante le simulazioni del terremoto dell'Aquila effettuate su un telaio presso UniNA. La sequenza temporale delle acquisizioni è dettagliata nella tabella 2.2.1.

Data di acquisizione	Stress % (rispetto al terremoto dell'Aquila)		
26 gennaio 2021	10%		
28 gennaio 2021	25%	50%	75%
29 gennaio 2021	100%		
2 febbraio 2021	125%	150%	

Tabella 2.2.1 Sequenza temporale delle acquisizioni eseguite presso UniNA.

Ancora una volta, abbiamo scelto con cura come eseguire l'allenamento dell'autoencoder. A tale scopo, abbiamo deciso di suddividere le acquisizioni nel modo seguente:

- *Training*: dati al 10%, 25%, 50%
- *Test*: dati al 75%, 100%, 125%, 150%

Abbiamo scelto questa configurazione perché abbiamo ipotizzato che con ampiezze basse il telaio lavorasse ancora in campo elastico, e questo rappresenta per noi una normale fase operativa, mentre con ampiezze elevate lavorasse in campo anelastico. Ci aspettavamo che l'autoencoder fosse in grado di riconoscere i picchi discutibili come anomali.

Le figure seguenti evidenziano gli anomaly score assegnati dall'autoencoder a ciascun campione acquisito. La prima figura mostra il segnale completo ottenuto concatenando ogni segnale acquisito ad ogni percentuale di stress, mentre le altre mostrano ogni segnale in dettaglio. Ancora una volta, si noti che gli ultimi $W - 1$ campioni non vengono esaminati, poiché non possono rientrare in W finestre successive.

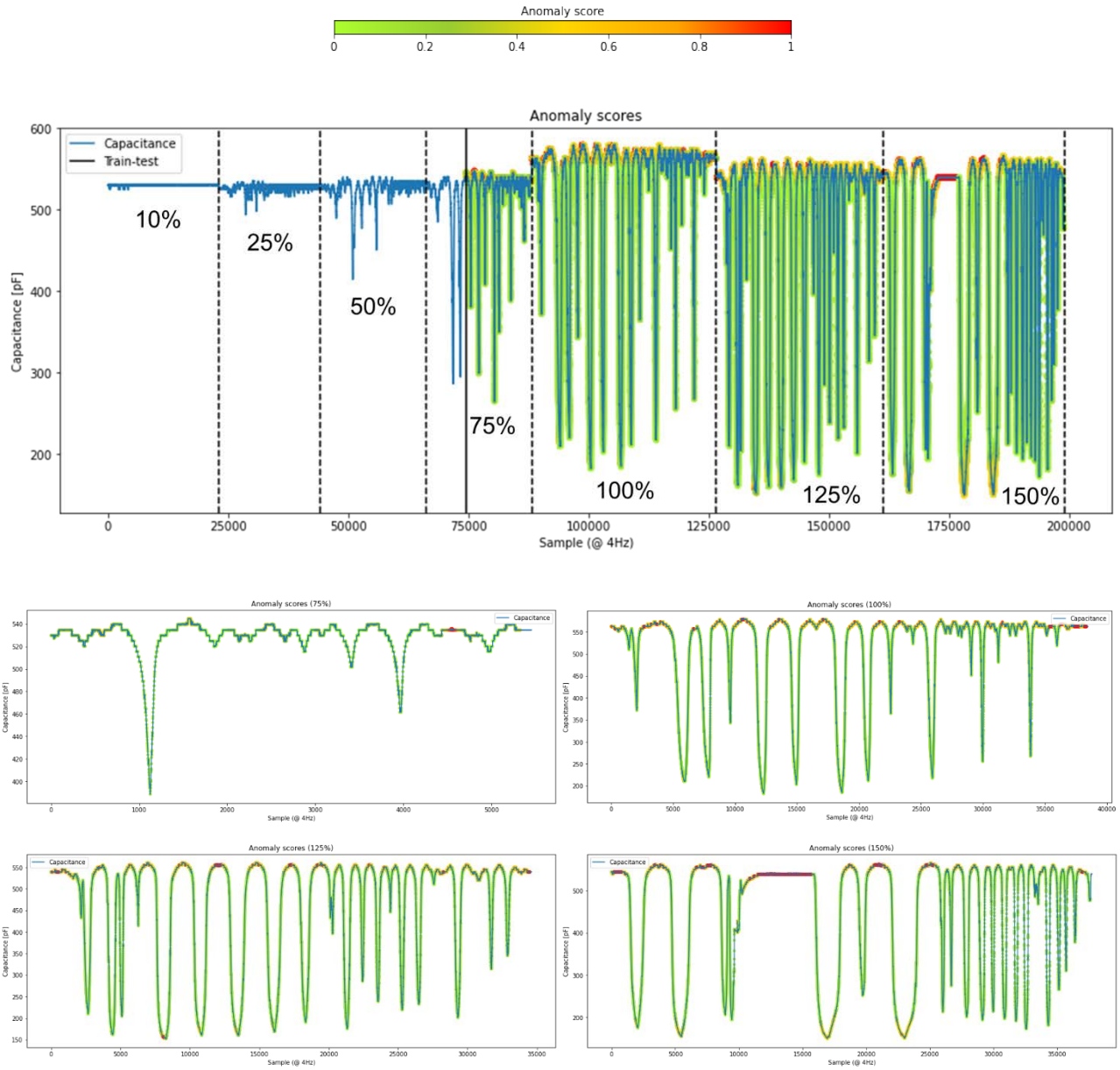


Figura 2.2.3 Risultati delle analisi sul dataset dei test eseguiti presso UniNA.

Anche in questo caso i risultati sembrano abbastanza buoni (figura 2.2.3), in quanto il modello sembra assegnare correttamente anomaly score più elevati a campioni vicini ai picchi discutibili o appartenenti a trend "insoliti", come quello costante che possiamo notare nei campioni da 10000 a 15000 al 150%, dove il segnale si mantiene costante a causa di un problema alla pressa utilizzata durante le prove.

2.3 Descrizione dell'attività: architettura di sistema multisensore

L'architettura di sistema del sistema di monitoraggio intelligente è stata successivamente rivista e generalizzata. La differenza principale rispetto all'architettura precedente è che l'elaborazione non viene più fatta semplicemente a livello del singolo nodo di acquisizione, ma si è deciso di estenderla a un gruppo di sensori che condividono alcune caratteristiche o sono relativi allo stesso elemento strutturale. Questa rappresenta fondamentalmente una versione multisensoriale dell'algoritmo precedente, e consente al sistema di catturare caratteristiche comuni dai dati acquisiti da diversi sensori, il che fornisce un anomaly score che tiene conto del contributo di più sensori, presumibilmente rafforzandone la consistenza e riducendo i falsi positivi.

La nuova architettura del sistema ha un impatto sull'algoritmo di Anomaly Detection, poiché ora il sistema ha più ingressi, uno per ciascun sensore, mentre emette, come prima, un singolo anomaly score, che condensa il contributo di ciascun sensore nel gruppo.

La nuova architettura del sistema è mostrata in figura 2.3.1.

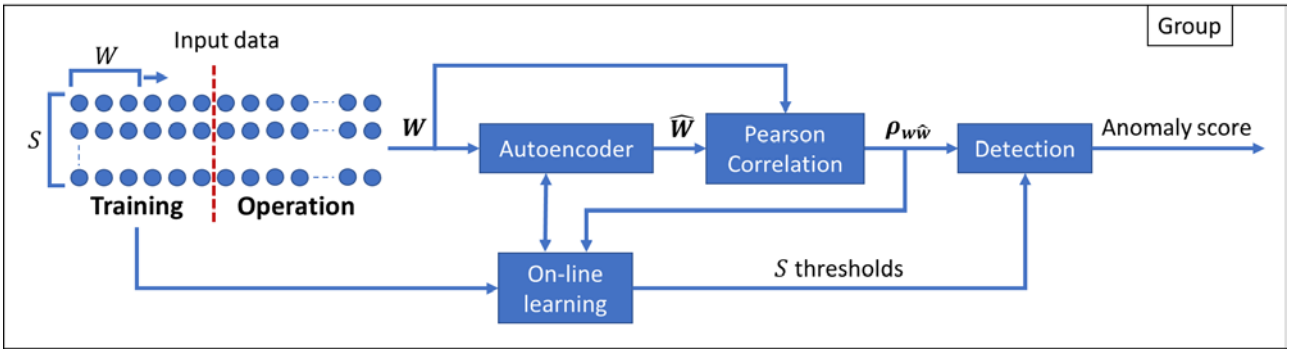


Figura 2.3.1 Architettura del sistema di monitoraggio intelligente multisensore.

2.3.1 Analisi sperimentale

È stata verificata la validità del sistema descritto in precedenza sui dati dei sensori capacitivi acquisiti durante le prove di carico sui sensori pre-installati e post-installati eseguite presso UniPA e durante le simulazioni di terremoto dell'Aquila effettuate presso UniNA. Poiché quando è disponibile una sola acquisizione di sensori il comportamento del nuovo sistema è equivalente a quello del vecchio sistema, abbiamo testato la validità del sistema solo quando erano disponibili più acquisizioni dei sensori. Come per il precedente algoritmo, si è deciso di adottare sliding window di dimensione pari a 64 per il dataset di Palermo e a 128 per il dataset di Napoli, poiché la frequenza di campionamento nel dataset di Napoli è maggiore di quella nel dataset di Palermo.

2.4 Risultati attività

2.4.1 Risultati sul primo dataset di Palermo

L'analisi dei primi dati acquisiti presso l'Università degli Studi di Palermo (UniPA) ha riguardato le prove effettuate su provini con sensori capacitivi preinstallati nei laboratori del Dipartimento di Ingegneria Civile, Ambientale, Aerospaziale e dei Materiali (DICAM). Nella figura 2.4.1 sono riportati gli schemi dei provini analizzati con i sensori installati.

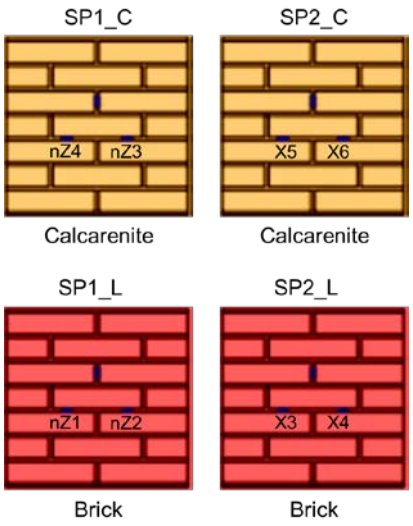


Figura 2.4.1 Schema dei sensori STMicroelectronics installati nei provini in muratura presso UniPA, analizzati nel caso di sistema multisensore.

In ogni fase di test, le letture dei sensori acquisite dai nostri sistemi e da UniPA sono state memorizzate come file di testo e/o CSV. Questi file, opportunamente denominati per tenere traccia dei provini e dei sensori, sono stati quindi preelaborati, in modo tale che tutte le letture rappresentative dell'andamento nel tempo delle risposte dei sensori capacitivi (espressi in pF) e del carico applicato dalla pressa (espresso in kN) fossero allineate alla stessa base dei tempi. Il risultato è stato quindi utilizzato per eseguire l'analisi descritta nella sezione seguente.

In ogni prova di compressione si possono distinguere tre fasi:

- due fasi di precarico
- una fase di carico fino a rottura

Data questa configurazione, abbiamo deciso, come prima, di allenare il nostro autoencoder sulla prima fase di precarico, che rappresenta per noi una normale fase di funzionamento, validarlo sui pochi campioni seguenti, e poi testarlo sulla seconda fase di precarico e sulla fase di carico fino a rottura.

Le figure seguenti evidenziano gli anomaly score assegnati dall'autoencoder a ciascun campione acquisito. Si noti che gli ultimi $W - 1$ punti non vengono esaminati, poiché non possono rientrare in W finestre successive.

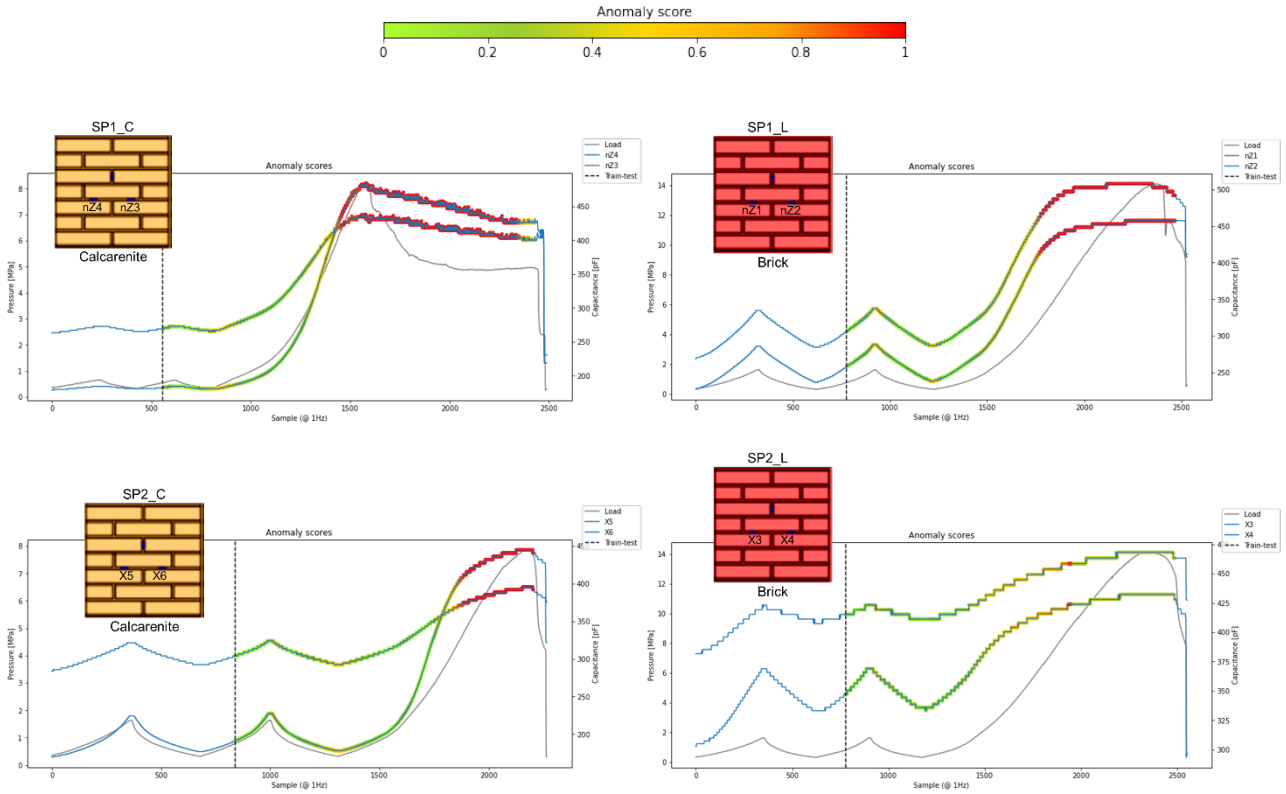


Figura 2.4.2 Risultati delle analisi sul primo dataset dei test eseguiti presso UniPA, analizzati nel caso di sistema multisensore.

Come possiamo notare (figura 2.4.2), i risultati sembrano abbastanza buoni, in alcuni casi anche migliori che con il precedente sistema, poiché il modello è in grado di sfruttare le informazioni congiunte fornite da più acquisizioni di sensori per assegnare correttamente anomaly score più elevati ai campioni prima e dopo la rottura, mentre, ad eccezione di alcuni campioni discutibili, assegna anomaly score molto bassi ai campioni nella seconda fase di precarico. L'unica eccezione è rappresentata dal provino SP2_L, dove le acquisizioni dal sensore X3 (in basso) e ancor più dal sensore X4 (in alto) non sono molto fluide, ma con "step" molto ampi, e questo rende l'algoritmo incapace di sfruttare le informazioni congiunte fornite dai sensori.

2.4.2 Risultati sul secondo dataset di Palermo

L'analisi del secondo dataset acquisito presso l'Università degli Studi di Palermo (UniPA) ha riguardato le prove effettuate su provini con sensori capacitivi post-installati nei laboratori del Dipartimento di Ingegneria Civile, Ambientale, Aerospaziale e dei Materiali (DICAM). Nella figura 2.4.3) sono riportati gli schemi dei provini analizzati con i sensori installati.

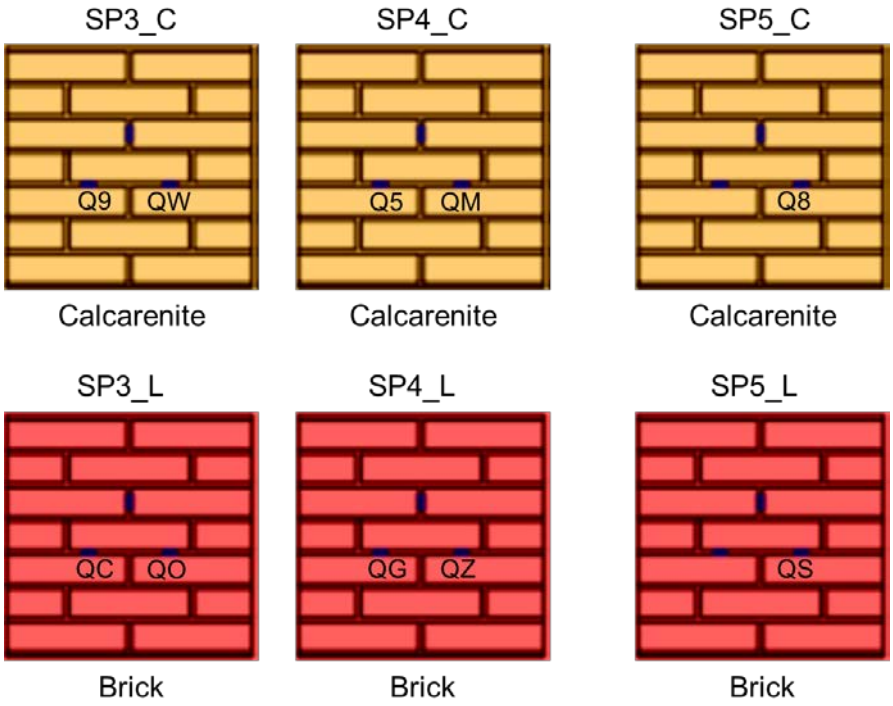


Figura 2.4.3 Schema dei sensori STMicroelectronics installati nei provini in muratura presso UniPA, analizzati nel caso di sistema multisensore.

In ogni prova di compressione si possono distinguere tre fasi:

- due fasi di precarico
- una fase di carico fino a rottura

Data questa configurazione, come nel primo dataset, abbiamo deciso di allenare il nostro autoencoder sulla prima fase di precarico, che rappresenta per noi una normale fase operativa, validarlo sui pochi campioni successivi, e poi testarlo sulla seconda fase di precarico e sulla fase di carico fino a rottura.

Le figure seguenti evidenziano gli anomaly score assegnati dall'autoencoder a ciascun campione acquisito. Si noti che gli ultimi $W - 1$ campioni non vengono esaminati, poiché non possono rientrare in W finestre successive.

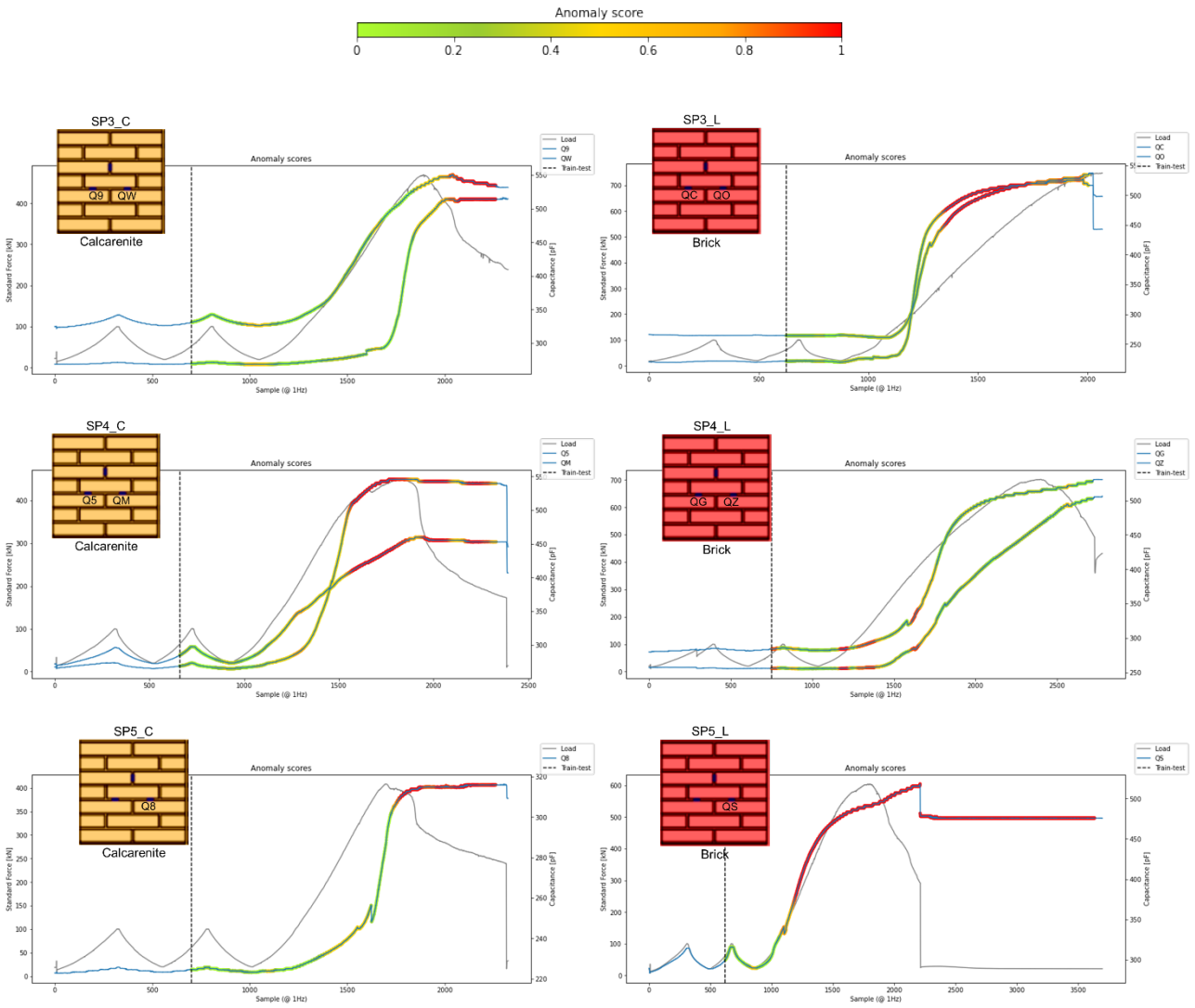


Figura 2.4.4 Risultati delle analisi sul secondo dataset dei test eseguiti presso UniPA, analizzati nel caso di sistema multisensore.

Come nel primo dataset, i risultati sembrano abbastanza buoni (figura 2.4.4), poiché il modello è in grado di sfruttare le informazioni congiunte fornite da più acquisizioni di sensori per assegnare correttamente anomaly score più elevati ai campioni prima e dopo il punto di rottura, mentre, ad eccezione di alcuni campioni discutibili, assegna anomaly score molto bassi ai campioni nella seconda fase di precarico. L'unica eccezione è relativa a un campione di mattone (come nel primo dataset). In particolare, è rappresentato dal provino SP4_L, dove le acquisizioni dei sensori QG e QZ hanno una dinamica troppo "morbida" nella prima fase di precarico (train), e questo rende il modello impreciso nella fase di carico (test).

2.4.3 Risultati sul secondo dataset di Napoli

Abbiamo anche testato la validità dell'algoritmo su un altro dataset acquisito durante le simulazioni del terremoto dell'Aquila effettuate su un telaio presso UniNA. La sequenza temporale delle acquisizioni è dettagliata nella tabella 2.2.1.

Per provare l'efficacia del nuovo sistema, abbiamo deciso di utilizzare le acquisizioni provenienti da due sensori:

- Z10, che è il miglior sensore delle prove
- Z12, che più o meno ha saputo seguire le dinamiche del telaio in esame

Ancora una volta, si è scelto con cura come eseguire l'allenamento dell'autoencoder. A tale scopo, si è deciso di suddividere le acquisizioni nel modo seguente:

- *Training*: dati al 10%, 25%, 50%
- *Test*: dati al 75%, 100%, 125%, 150%

Questo setup è stato scelto perché è possibile ipotizzare che con ampiezze basse il telaio lavorasse ancora in campo elastico, e questo rappresenta per noi una normale fase operativa, mentre con ampiezze elevate lavorasse in campo anelastico. Ci si aspettava, quindi, che l'autoencoder riconoscesse i picchi discutibili come anomali.

Le figure 2.4.5 evidenziano gli anomaly score assegnati dall'autoencoder a ciascun campione acquisito. La prima figura mostra il segnale completo ottenuto concatenando ogni segnale acquisito ad ogni percentuale di stress, prima quello acquisito da Z10, poi quello acquisito da Z12, mentre le altre mostrano ogni segnale in dettaglio. Ancora una volta, si noti che gli ultimi $W - 1$ campioni non vengono esaminati, poiché non possono rientrare in W finestre successive.

Anche in questo caso i risultati sembrano abbastanza buoni e molto simili a quelli ottenuti nel precedente sistema, poiché anche in questo caso il modello sembra assegnare correttamente anomaly score più elevati a campioni vicini ai picchi discutibili o appartenenti a trend "insoliti", come quello costante possiamo vedere nei campioni da 10000 a 15000 al 150%, dove il segnale si mantiene costante a causa di un problema alla pressa utilizzata durante le prove.

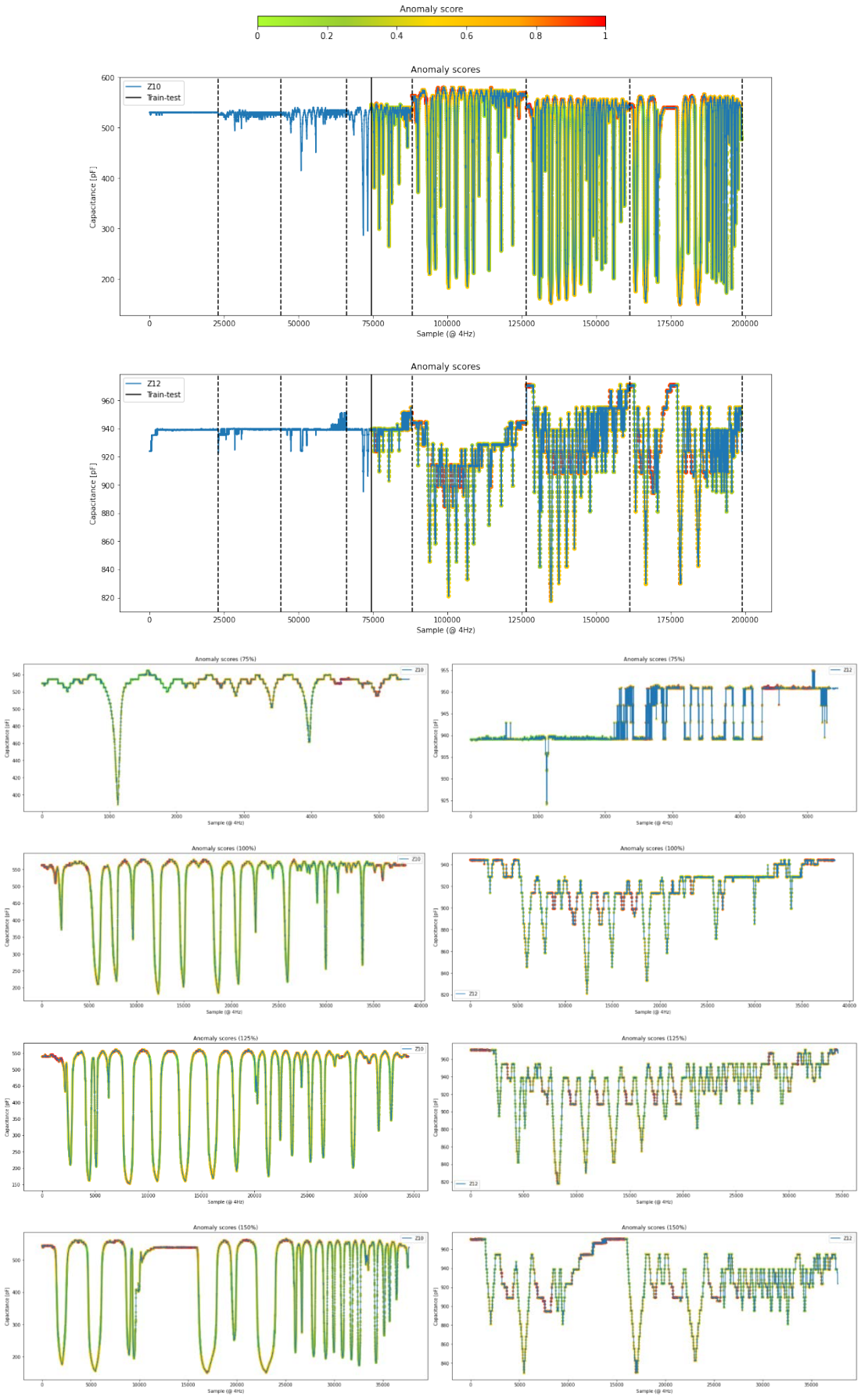


Figura 2.4.5 Risultati delle analisi sul secondo dataset dei test eseguiti presso UniNA, analizzati nel caso di sistema multisensore.

2.5 Descrizione dell'attività: Piattaforma di edge computing

L'*edge computing* è una forma di elaborazione che avviene presso o vicino alla posizione fisica dell'utente o delle fonti dei dati, come dispositivi IoT o server periferici locali, riducendo al minimo la necessità di elaborazione dei dati in un data center remoto.

L'invio di tutti i dati generati dal dispositivo a un data center centralizzato o al cloud, infatti, causa problemi di larghezza di banda e latenza. L'edge computing offre un'alternativa più efficiente: i dati vengono elaborati e analizzati più vicino al punto in cui vengono creati. Poiché i dati non attraversano una rete verso un cloud o un data center per essere elaborati, questa vicinanza dei dati alla fonte riduce significativamente la latenza. Pertanto, l'edge computing consente un'analisi dei dati più rapida e completa, creando l'opportunità di intuizioni più "profonde", tempi di risposta più rapidi e una migliore disponibilità di larghezza di banda.

2.5.1 Architettura sistema

Il sistema è composto da due parti:

- Un sistema di acquisizione, responsabile della gestione delle acquisizioni dei sensori
- Un sistema di anomaly detection, che interroga il sistema di acquisizione per ottenere le acquisizioni dei sensori ed eseguire l'elaborazione AI relativa all'algoritmo di anomaly detection

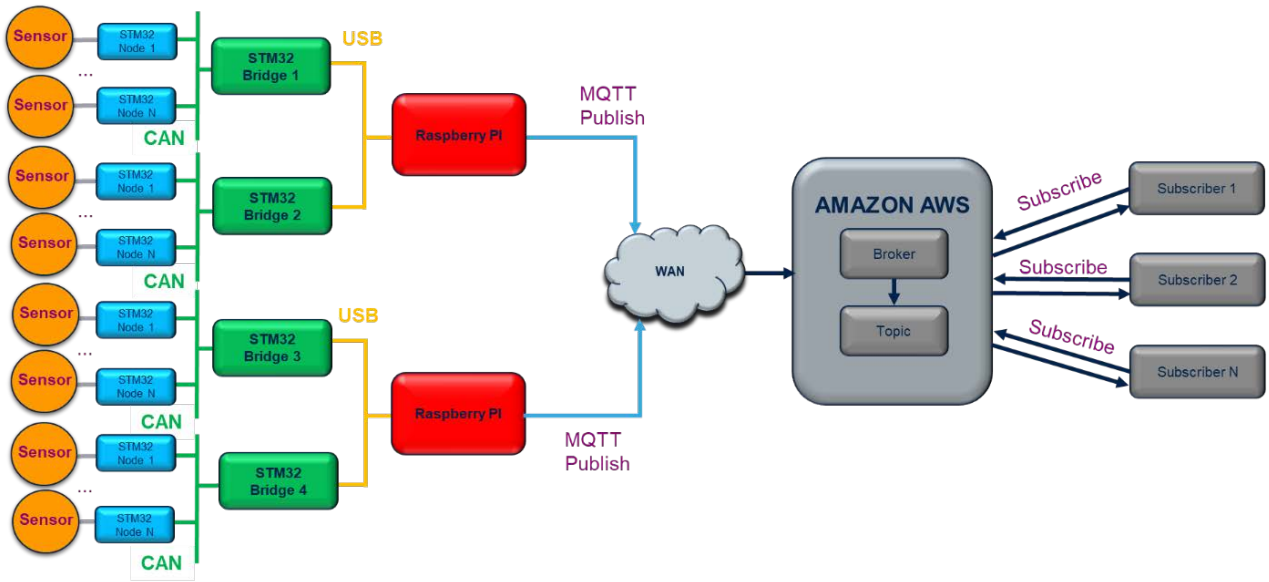


Figura 2.5.1 Schema del sistema RaspberryPi-based per l'acquisizione dei sensori STMicroelectronics.

Come illustrato in figura 2.5.1, il sistema di acquisizione è composto da diverse sottoreti, ciascuna composta da:

- Diversi nodi STM32, responsabili dell'acquisizione e del trasferimento dei dati dai sensori tramite CAN
- Una serie di bridge STM32, uno per ogni sottorete CAN, che richiedono ai nodi le acquisizioni dei sensori e le trasferiscono tramite USB
- Una Raspberry PI, che raccoglie le acquisizioni, le memorizza su un DB e le invia attraverso un router a un broker MQTT Amazon AWS

Viene quindi utilizzata una dashboard di Amazon AWS per monitorare e tracciare le acquisizioni in tempo reale.

Il sistema di anomaly detection interagisce strettamente con il sistema di acquisizione, ricavandone le acquisizioni ed avviando l'elaborazione.

Si è deciso di non implementare le funzionalità AI nella stesso Raspberry Pi utilizzata per gestire le acquisizioni per tre motivi principali:

1. distribuire i carichi di lavoro e abilitare l'analisi multisensoriale, che richiede un "concentratore", che può potenzialmente richiedere di acquisire dati da sensori appartenenti a diverse Raspberry Pi
2. aumentare la *fault tolerance*, poiché l'utilizzo di due sistemi separati rende un sistema indipendente dall'altro, e quindi qualsiasi guasto su uno non influisce sull'altro
3. velocizzare lo sviluppo

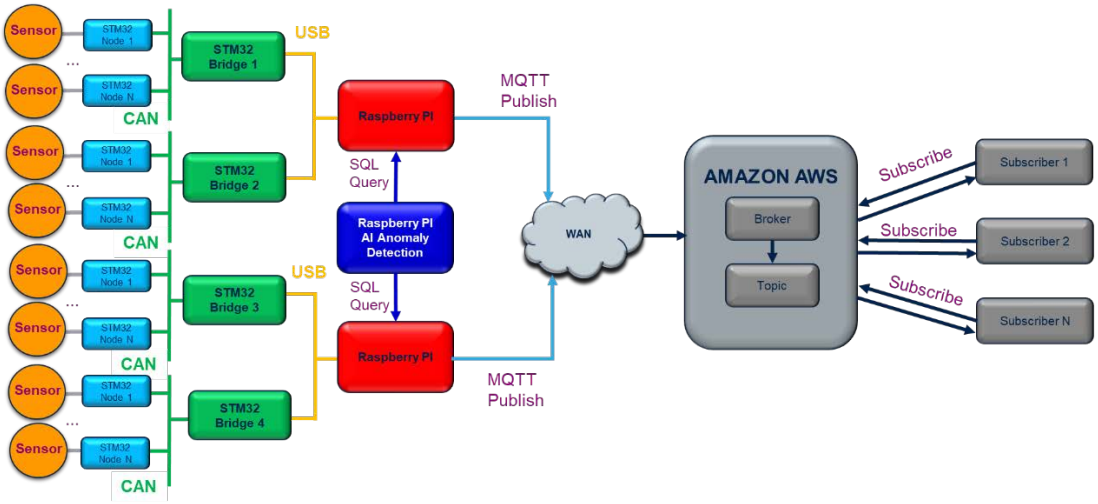


Figura 2.5.2 Schema del sistema RaspberryPi-based per l'acquisizione dei sensori STMicroelectronics con il sistema di anomaly detection integrato.

Come illustrato in figura 2.5.2, sostanzialmente il sistema di anomaly detection è composto da una o più RaspberryPi (a seconda delle esigenze computazionali dell'applicazione), ognuna delle quali è connessa alla RaspberryPi di acquisizione e invia periodicamente query SQL per ottenere i dati dei sensori ed elaborarli.

2.6 Risultati dell'attività: Monitoraggio in tempo reale

Per fornire un mezzo per analizzare facilmente l'andamento degli anomaly score relativi a un gruppo di sensori, e avvisare quando l'anomaly score supera una certa soglia di rischio, si è deciso di sviluppare uno script in Python per il monitoraggio in tempo reale del sistema in esame. Questo legge una tabella dal database creato dal sistema di anomaly detection (quella relativa al gruppo di sensori che ci interessa) e:

- traccia l'andamento degli anomaly score
- registra avvisi che possono aiutare un utente che monitora, se necessario, a intraprendere azioni

Lo screenshot di figura 2.6.1 mostra lo script durante l'esecuzione.

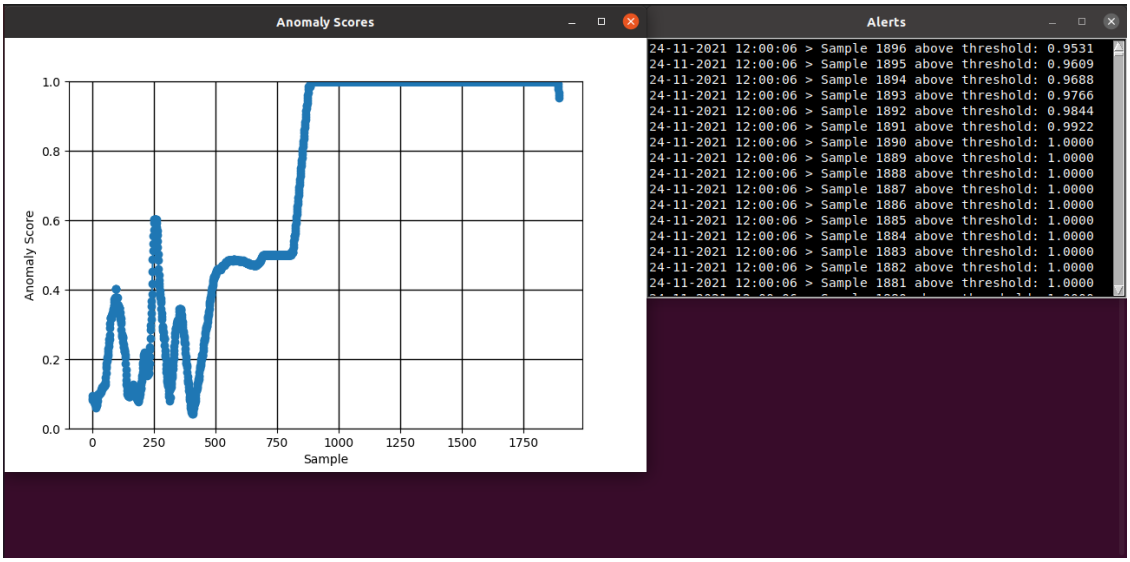


Figura 2.6.1 Script Python per il monitoraggio in tempo reale in esecuzione in un test del sistema RaspberryPi-based per l'acquisizione dei sensori STMicroelectronics con il sistema di anomaly detection integrato.

Gli avvisi vengono generati quando un gruppo di campioni con un anomaly score maggiore di una soglia configurabile (0.9 in figura) viene recuperato dal database.

3 Raccolta ed integrazione di dati acquisiti da sensori eterogenei

3.1 Descrizione attività

La grande diffusione dei sistemi di sensori, insieme alle loro numerose applicazioni, ha portato a un'enorme eterogeneità nella logica di interfacciamento e raccolta dei dati da questi sistemi. Uno dei principali problemi aperti nel monitoraggio e nella gestione ambientale è l'integrazione di dati eterogenei provenienti da fonti diverse per gestire ed elaborare strategie di mitigazione del rischio, compresa la pianificazione della gestione delle emergenze. Tuttavia, per automatizzare l'elaborazione di tali dati, sono necessari specifici framework di integrazione per l'accesso a diverse fonti di dati. Questi framework dovrebbero essere in grado di accedere ai dati rilevati da diverse infrastrutture di rilevamento, dovrebbero nascondere l'eterogeneità dei diversi sistemi di sensori (in termini di sensori, reti o tecnologie middleware) e fornire un modo standard per accedervi. Sono state condotte interessanti ricerche sulle tecniche di integrazione delle reti di sensori eterogenee, ma oggi sono state proposte solo poche architetture. La maggior parte di esse cerca di definire un meccanismo di scambio comune tra diversi sistemi di sensori per facilitare l'integrazione e di fornire un livello di integrazione software che permetta a diversi sistemi di sensori di collaborare per lo stesso scopo. Molto spesso queste soluzioni sono strettamente legate a tecnologie proprietarie e non sono né scalabili né aperte.

Negli ultimi anni, l'approccio SOA (Service-Oriented Architecture) è diventato una pietra miliare in molti progetti di reti di sensori. L'approccio SOA consente la cooperazione e l'interoperabilità di diverse piattaforme di sensori, in quanto fornisce la scoperta, l'accesso e la condivisione di servizi, dati, risorse di calcolo e di comunicazione attraverso l'adozione di standard aperti. I nuovi standard aperti consentono di definire ambienti cooperativi e si basano sull'adozione di modelli di dati comuni per definire e rappresentare formalmente la conoscenza dei dati. Come esempio di modello di dati, l'OpenGeospatial Consortium fornisce uno schema XML (il Sensor Model Language) per definire le caratteristiche geometriche, dinamiche e osservative dei sensori e altri standard come Observation and Measurement per descrivere i fenomeni osservati [1]. Gli standard: (i) forniscono informazioni generali sui sensori per supportare la scoperta dei dati, (ii) supportano l'elaborazione e l'analisi delle misurazioni dei sensori, (iii) supportano la geolocalizzazione dei dati misurati, (iv) forniscono caratteristiche di prestazione (ad esempio, accuratezza, soglia, ecc.) e (v) archiviano le proprietà e le ipotesi fondamentali relative ai sensori. In effetti, questi standard si concentrano sugli aspetti sintattici dei problemi di interoperabilità.

Nelle attività condotte, viene proposto un modello architettonico basato sui servizi per fornire interoperabilità tra diverse reti di sensori e viene introdotto e discusso in dettaglio un modello di dati per sensori basato sulla semantica. Come già detto, il grande vantaggio delle architetture di servizi web è quello di costruire sistemi informativi che permettano la creazione di applicazioni combinando servizi interoperabili e liberamente accoppiati senza la conoscenza dei sistemi sottostanti. D'altra parte, il modello di dati arricchito semanticamente garantisce l'interoperabilità tra sistemi multi-tecnologici, fornendo un modello formale per l'integrazione di dati raccolti da fonti diverse ed eterogenee. Un modello formale di dati specifica le relazioni tra i dati, la terminologia e i significati, ed è implementato attraverso un insieme di ontologie descritte formalmente in Web Ontology Language (OWL). Di seguito si presenta un'architettura innovativa per la gestione del rischio, completamente basata su servizi, che consente l'integrazione di dati eterogenei e l'implementazione di servizi Web standard applicativi per rendere disponibili i dati (grezzi o aggregati) agli utenti finali autorizzati e/o alle applicazioni che ne hanno bisogno. Verrà quindi descritto un servizio di raccolta dati con la definizione di un modello di dati comune per sensori specifici. I dati analizzati sono relativi a sensori sismici, vulcanici, pluviometrici e meteo-marini, saranno formalizzati secondo lo standard Observation and Measurement e saranno arricchiti con informazioni semantiche.

La necessità di garantire l'interoperabilità tra diversi sistemi di monitoraggio e l'integrazione di dati eterogenei può essere vista da diversi punti di vista: (i) sintattico, per superare l'eterogeneità tecnica; (ii) semantico, per superare le ambiguità e le diverse interpretazioni; (iii) applicativo, per fornire concetti e componenti sostenibili e riutilizzabili del dominio applicativo; e (iv) di osservazione dei fenomeni, per valutare il significato dell'osservazione da prospettive temporali, spaziali e tematiche. In letteratura sono disponibili alcune soluzioni ad aspetti specifici, ma non completi, dell'interoperabilità; l'OGC ha proposto Sensor Web Enablement (SWE) [1], una suite di specifiche per modellare le caratteristiche e i servizi dei sensori. In particolare, la suite comprende: (i) Sensor Model Language (Sensor ML), (ii) Observation & Measurement e (iii) Sensor Observation Service. Essi consentono di modellare il sensore e le osservazioni del sensore, il meccanismo di recupero dei dati e i servizi web (per l'accesso ai dati del sensore via web); è possibile specificare informazioni come coordinate e timestamp, ma non consentono di dichiarare le relazioni intrinseche tra i dati e il significato delle osservazioni del sensore, rendendo difficile l'interoperabilità, la valutazione dei fenomeni e la rilevazione della consapevolezza della situazione [7]. Un primo tentativo di definizione della semantica per il Web dei sensori è stato proposto nel framework SSW [8], in cui il significato delle osservazioni dei sensori viene rafforzato aggiungendo annotazioni semantiche ai linguaggi dei sensori standard esistenti della SWE, al fine di aumentare l'interoperabilità e fornire informazioni contestuali per la consapevolezza della situazione. Anche nel campo della modellazione dei dati sono

stati compiuti diversi sforzi: sono state proposte diverse ontologie per la rappresentazione di dati eterogenei dei sensori. L'ontologia presentata da Ceruti in [14] modella diversi concetti, come piattaforme e sensori, caratteristiche materiali e immateriali, relazioni e concetti come le combinazioni di dati. Un'altra ontologia per le reti di sensori, presentata da Eid et al. in [16], fornisce una rappresentazione semantica dei dati delle reti di sensori, con l'obiettivo di interpretare informazioni strutturate non ambigue.

Dal punto di vista dell'interoperabilità architetturale, l'attuale approccio di punta si basa sulle architetture orientate ai servizi (SOA). In [17] gli autori hanno introdotto i meccanismi del Web Service Resource Framework (WSRF) all'interno dell'implementazione dei servizi core della NICTA Open Sensor Web Architecture (NOSA). WSRF consente di eseguire interrogazioni osservative simultanee a reti di sensori eterogenee. Inoltre, il gruppo GeoICT dell'Università di York ha costruito un'infrastruttura Sensor Web conforme a OGC SWE [18], sviluppando un client Sensor Web per la visualizzazione di dati geospaziali [11] e un insieme di servizi Web stateless chiamato GeoSWIFT [19]. Inoltre, sono stati proposti due principali progetti europei di allerta precoce basati sui servizi: i progetti Wide Information Network (WIN) [10], [9] e ORCHESTRA [9]. WIN mira a sviluppare una piattaforma aperta e flessibile per supportare domini di rischio e pericolo multipli a livello europeo, integrando i flussi di dati nazionali e regionali nell'ambito di un'architettura di servizi Web e proponendo un insieme di servizi generici e componenti standard per la modellazione dei dati che possono essere utilizzati nella realizzazione di diversi casi. Il modello di metadati WIN si basa su standard esistenti, come Dublin Core [3], GML [4] e ISO19115 [5], e fornisce specifiche aggiuntive per WIN. Permette di gestire in modo uniforme tutti i metadati utilizzando lo standard Electronic Business Registry Information Model (ebRIM). I modelli di dati WIN sono implementati tramite il componente Data Web Service e i metadati tramite uno strumento ebXML Registry. Entrambi i componenti e gli strumenti utilizzano servizi Web di tipo SOAP/WSDL. D'altra parte, ORCHESTRA adatta il modello di riferimento ISO/IEC 10746 per l'elaborazione distribuita aperta alle architetture orientate ai servizi [6]; adattando, in particolare, il modello di riferimento per le reti di servizi geospaziali su un modello di processo conforme allo standard ISO RM-ODP, al fine di progettare e implementare SOA geospaziali. L'architettura ORCHESTRA utilizza la piattaforma W3C Web services e il Geography Mark-up Language (GML) per implementare i servizi web.

Sono stati sviluppati diversi metodi e tecniche per assicurare l'interoperabilità tra diversi sistemi di monitoraggio e l'integrazione di dati eterogenei. Essi rappresentano soluzioni a problemi specifici di interoperabilità e/o integrazione, ma, al meglio delle nostre conoscenze, non esiste ancora una proposta completa di architettura che fornisca una soluzione per l'intera gamma di esigenze (cioè in

grado di gestire l'osservazione dei fenomeni, l'interoperabilità tra diversi sistemi di monitoraggio e l'integrazione di dati eterogenei, e che possa essere facilmente integrata con altre piattaforme preesistenti).

3.2 Risultati attività

Per progettare un sistema aperto e gestire fonti di dati eterogenee di sensori, proponiamo un'architettura orientata ai servizi basata sulle tecnologie dei servizi Web. L'adozione di tecnologie standard come HTTP, XML, SOAP, WSDL e UDDI [2] consente l'adozione e la distribuzione pervasiva di servizi Web per raggiungere l'interoperabilità. Infatti, i servizi Web sono applicazioni basate su Internet che comunicano con altre applicazioni per offrire dati o servizi in modo standard.

L'architettura proposta offre diversi servizi e funzionalità, che possono essere classificati in due categorie principali: (a) servizi per elaborare dati da fonti eterogenee; (b) servizi per gestire tutte le fonti di dati (sensori, database, simulatori, ecc.). Abbiamo individuato tre livelli orizzontali: Data Collection Service Layer, Integration Service Layer, Application Service Layer come illustrato nella Fig. 1, inoltre possono essere introdotti alcuni strati trasversali per fornire servizi di sicurezza, gestione e interoperabilità. Ogni strato offre servizi diversi ed è caratterizzato da una profonda specializzazione funzionale; inoltre, ogni strato è in grado di comunicare con gli altri strati attraverso interfacce WSDL standard.

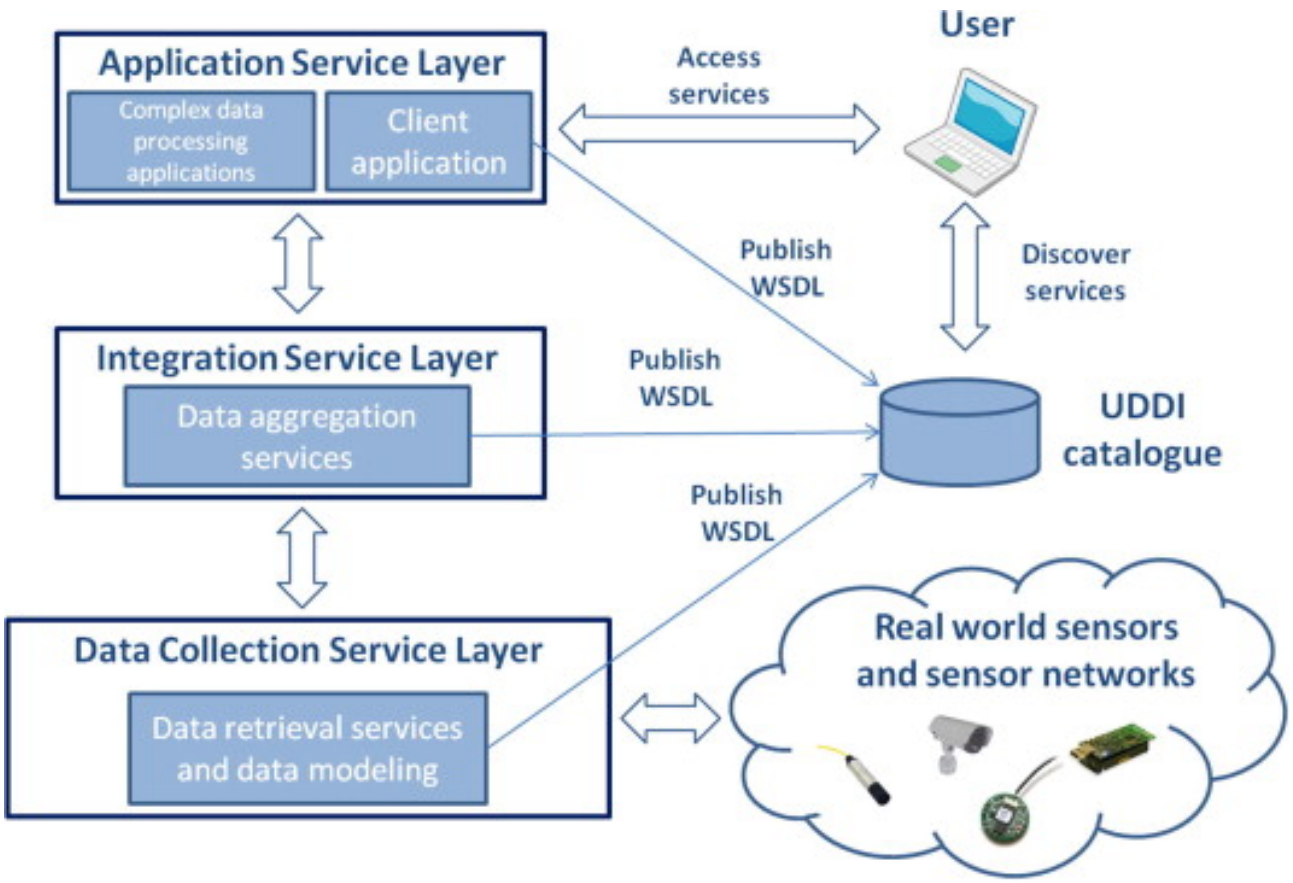


Fig.1 Schema di concetto dell'architettura

Raggruppare servizi simili in diversi livelli offre diversi vantaggi: i servizi più complessi sono costruiti sulla collaborazione di più elementi di minore complessità; dal punto di vista della sicurezza, ogni servizio può avere una politica di sicurezza specifica per controllare l'accesso e preservare la privacy dei dati. Questo approccio consente di costruire un'architettura più robusta, scalabile e manutenibile, con una migliore struttura logica.

Di seguito, forniamo una breve descrizione di ogni livello e dei relativi servizi specifici:

1. Data Collection Service Layer: le reti di sensori hanno linguaggi e paradigmi di interrogazione specifici e diversi, quindi il Data Collection Service Layer fornisce ai livelli superiori una visione omogenea delle reti. A tal fine, è necessario un modello di dati dei sensori, delle misure e dei fenomeni per condividere correttamente le informazioni rilevanti dei fenomeni osservati. L'approccio innovativo che proponiamo mira ad arricchire i dati rilevati con tecniche di modellazione ontologica, mappando i set di dati dei sensori da una rappresentazione grezza fornita dall'hardware/middleware del sensore a un modello di dati XML/RDF e a una rappresentazione standard O&M. Infatti, i servizi offerti in questo livello

traducono e arricchiscono semanticamente il formato dei dati proprietari in un modello di dati comune, come illustrato nelle prossime sezioni.

2. Integration Service Layer: i servizi di questo livello forniscono insiemi di dati aggregati a livello applicativo; in particolare, integrano reti diverse che osservano gli stessi fenomeni e che devono essere aggregate per comprendere meglio fenomeni complessi. I servizi raggruppano gli insiemi di dati di rete secondo il modello di dati definito. Consideriamo ad esempio uno scenario tipico in cui vengono adottate tecnologie diverse per rilevare gli stessi dati; mentre il livello di raccolta è responsabile del recupero di tali dati e della loro rappresentazione, questo livello è in grado di integrare i dati relativi agli stessi fenomeni ma osservati da reti diverse.
3. Application Service Layer: è possibile implementare diversi tipi di applicazioni per il monitoraggio e l'elaborazione dei dati rilevati. Possono implementare sistemi di supporto alle decisioni come modelli di soglia di allarme o notifica di eventi in tempo reale. Le applicazioni sono costruite invocando e componendo servizi definiti in altri livelli; tutti elaborano strutture di dati complesse formattate secondo il modello di dati definito e accessibili tramite WSDL standard.

Ogni servizio ha una propria politica di sicurezza ed è pubblicato in un registro pubblico.

Notiamo che i dati rilevati, che vengono memorizzati e utilizzati per proteggere le infrastrutture critiche, possono rappresentare informazioni sensibili, il che implica che tali dati devono essere gestiti solo da entità autorizzate. A questo scopo, i servizi di sicurezza (non riportati nella figura) applicano gli standard WS-Security per garantire l'accesso (autenticazione e autorizzazione) solo agli utenti autorizzati. Questi servizi si occupano anche della riservatezza e dell'integrità dei dati dei messaggi scambiati, del non ripudio delle richieste di messaggi e della resilienza agli attacchi denial-of-service.

Di seguito ci concentreremo sui livelli di servizio di integrazione e raccolta dati. Illustreremo un'architettura orientata ai servizi in grado di supportare le funzionalità di entrambi i livelli e proporremo un modello di dati comune arricchito da annotazioni semantiche per la rappresentazione dei dati e dei sistemi di sensori.

Una corretta modellazione dei dati è un compito essenziale per fornire una rappresentazione efficiente ed efficace dei fenomeni di interesse. Inoltre, è un passo cruciale per raggiungere la piena interoperabilità tra reti di sensori eterogenee.

La definizione e l'adozione di modelli è necessaria per cogliere le informazioni di interesse insieme ai formati e alle proprietà che devono essere rappresentate. Quindi, per i nostri obiettivi, uno degli

scopi principali del processo di modellazione è la raccolta e la strutturazione delle informazioni raccolte dai sensori che possono essere in un formato grezzo.

Il problema fondamentale da affrontare, quando si integrano dati provenienti da fonti eterogenee, deriva dalle differenze sintattiche, schematiche e semantiche nella rappresentazione. L'eterogeneità sintattica si riferisce alle differenze tra i paradigmi utilizzati per formattare i dati, come il testo semplice, i DB relazionali o i documenti xml. L'eterogeneità schematica si riferisce a diverse gerarchie di aggregazione o generalizzazione definite per gli stessi fenomeni. Infine, l'eterogeneità semantica riguarda il disaccordo sul significato o sull'interpretazione degli stessi dati [12], [13].

La nostra proposta mira ad affrontare e risolvere queste differenze. In particolare, l'eterogeneità sintattica viene risolta convertendo i formati di dati proprietari in un documento XML [25]; l'eterogeneità schematica viene affrontata adottando il linguaggio standard O&M [20]; infine l'eterogeneità semantica viene affrontata definendo un modello di dati comune descritto formalmente utilizzando un'ontologia.

L'adozione di un formalismo basato su un'ontologia è motivata dalla necessità di fornire un modello condiviso, basato sulla semantica, delle informazioni da elaborare. Inoltre, molti lavori recenti hanno dimostrato che la precisione nella scoperta dei servizi aumenta quando si utilizza una rappresentazione semantica anziché sintattica [15]. Abbiamo quindi definito un modello ontologico comune di dati, per descrivere i dati e le osservazioni dei sensori in modo univoco e non ambiguo.

Nell'ontologia del dominio dei sensori ricorriamo alla modellazione semantica dei dati per formalizzare una rappresentazione esplicita delle caratteristiche dei dati e per individuare relazioni e somiglianze tra i dati rilevati da fonti eterogenee che non possono essere ricavate direttamente dalla struttura sintattica dei dati. Il modello proposto è codificato in triple RDF, in conformità con l'approccio del Semantic Sensor Web [21], [22].

L'adozione di un'ontologia aiuta a fondere dati e significato, quindi, per la descrizione del modello, abbiamo adottato standard aperti per l'interoperabilità, arricchiti con informazioni semantiche. Infatti, una tecnologia basata sulla semantica ci permette di trattare i dati grezzi e di gestirli come una base di conoscenza globale, ottenendo una rappresentazione esplicita del significato dei dati e dei servizi, utile per estrarre le informazioni rilevanti e per integrarle. La conoscenza può essere gestita come un database che può essere interrogato in modo strutturato, consentendo operazioni avanzate come il ragionamento logico. Inoltre, è possibile applicare la verifica di coerenza sui dati modellati per verificare la conformità al modello e l'accettabilità dei valori rilevati.

Il modello ontologico risultante è costituito da due classi principali: la classe Sensore include le conoscenze sulle specifiche dei sensori; la classe Misura definisce le informazioni sui dati raccolti dai diversi sensori.

Come illustrato nella Fig. 2a, la classe Sensore modella le caratteristiche del sensore, quali Posizione, Identificazione, Misura, Tipo di sensore. Tra le altre, l'entità Location descrive le informazioni geografiche utilizzando le coordinate del sistema DMS o UTM, mentre l'entità Sensor Type specifica le informazioni sulla descrizione della tipologia del sensore.

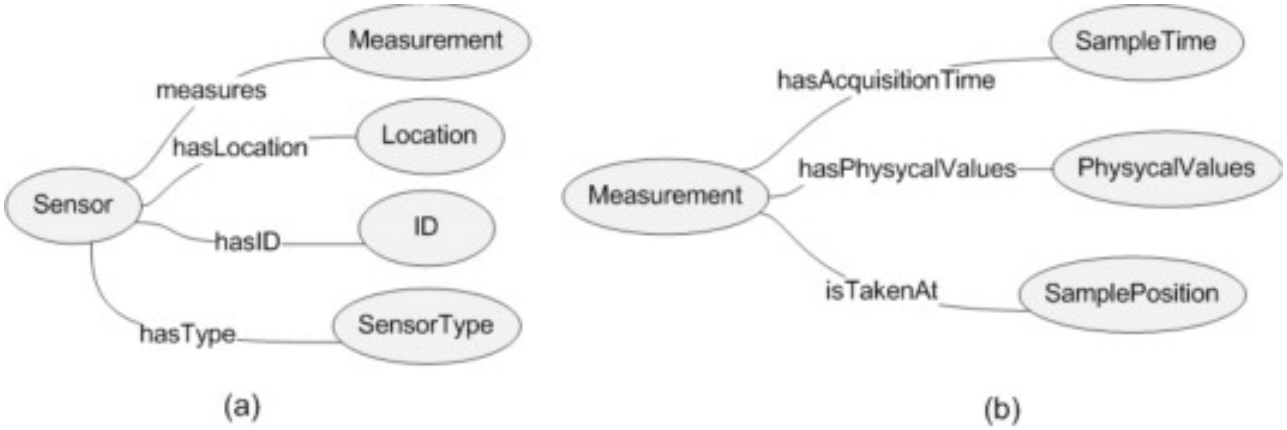


Fig.2 Le classi sensore e misura realizzate

La classe Misura, illustrata nella Fig. 2b, contiene informazioni sulle caratteristiche fisiche dei campioni raccolti. Ogni misura è caratterizzata da un insieme di parametri fisici come SampleTime, che indica il periodo di tempo di campionamento delle osservazioni dei fenomeni, e SamplePosition, che indica la posizione dei campioni all'inizio dell'osservazione, specificata nello stesso formato di dati della Location. Inoltre, i PhysicalValues sono espressi in termini di informazioni statistiche (come valori medi e covarianza) e sono coadiuvati da parametri del sensore (come potenza del sensore, tensione, rilevamento e direzione) che forniscono l'accuratezza e le unità di misura, definite in modelli standard esterni.

Si noti che tali informazioni semantiche non possono essere derivate dall'analisi dei dati del sensore o specificate utilizzando i linguaggi SWE; tali standard, infatti, consentono di caratterizzare concetti fisici, come le coordinate spaziali e temporali, ma non di specificare il significato delle osservazioni del sensore, che sono necessarie per consentire l'interoperabilità di sensori eterogenei e migliorare la consapevolezza della situazione.

Il modello proposto può essere esteso con ontologie esterne per aggiungere concetti e relazioni e fornire una descrizione arricchita di sensori, osservazioni e fenomeni; ad esempio, può essere esteso

per modellare come i sensori sono raggruppati in classi secondo associazioni dinamiche definite da specifici fenomeni sotto osservazione. Consideriamo, ad esempio, tre diversi tipi di sensori (di temperatura, di umidità e di vento) che possono essere collocati nella stessa area e che possono essere raggruppati dinamicamente per monitorare gli stessi fenomeni in modo collaborativo ai fini delle previsioni meteorologiche.

Infine, grazie a questo modello, è possibile condividere e standardizzare, a un livello superiore del servizio, il formato dei dati e risolvere così eventuali conflitti generati da fonti eterogenee.

Di seguito si illustra un'architettura orientata ai servizi, denominata InsistMod, che copre le funzionalità sia del servizio di integrazione che del servizio di raccolta dati. Per far fronte all'eterogeneità dei dati dei sensori, il nostro framework si basa su:

- un modello architetturale in grado di supportare in modo efficiente la gestione di tali dati anche se rilevati da reti diverse;
- un modello di dati in grado di rappresentare i dati dei sensori in una visione logica unica grazie a un arricchimento semantico, consentendo così una gestione semplice all'interno del framework.

Il modello architetturale è stato progettato sfruttando il paradigma wrapper-mediatore, una tecnica ben nota per integrare dati provenienti da fonti eterogenee [26]; secondo questo modello un mediatore può accedere a diverse fonti di dati per mezzo di connettori ad hoc (componenti wrapper). In uno scenario di lavoro tipico, ogni wrapper esplora e monitora la rete locale di sensori e invia al mediatore una descrizione appropriata delle informazioni correlate secondo un modello di dati comune. D'altro canto, il mediatore integra e organizza tali informazioni mantenendo una visione unica di tutti i sistemi, al fine di soddisfare le richieste dell'utente o dell'applicazione. Il modello di dati mira a rappresentare sia i nodi che le reti di sensori, combinando la descrizione strutturale e comportamentale dei sensori. Il modello di dati originale di InsistMod è stato arricchito con annotazioni semantiche, come descritto nella sezione successiva. Per avere un'architettura aperta e standard, abbiamo fornito sia i componenti del wrapper che del mediatore con interfacce di servizi web, in modo tale che il mediatore implementi le funzionalità del livello di servizio di integrazione, mentre un wrapper generico implementi quelle del livello di servizio di raccolta dati. L'architettura di riferimento è illustrata nella Fig. 3.

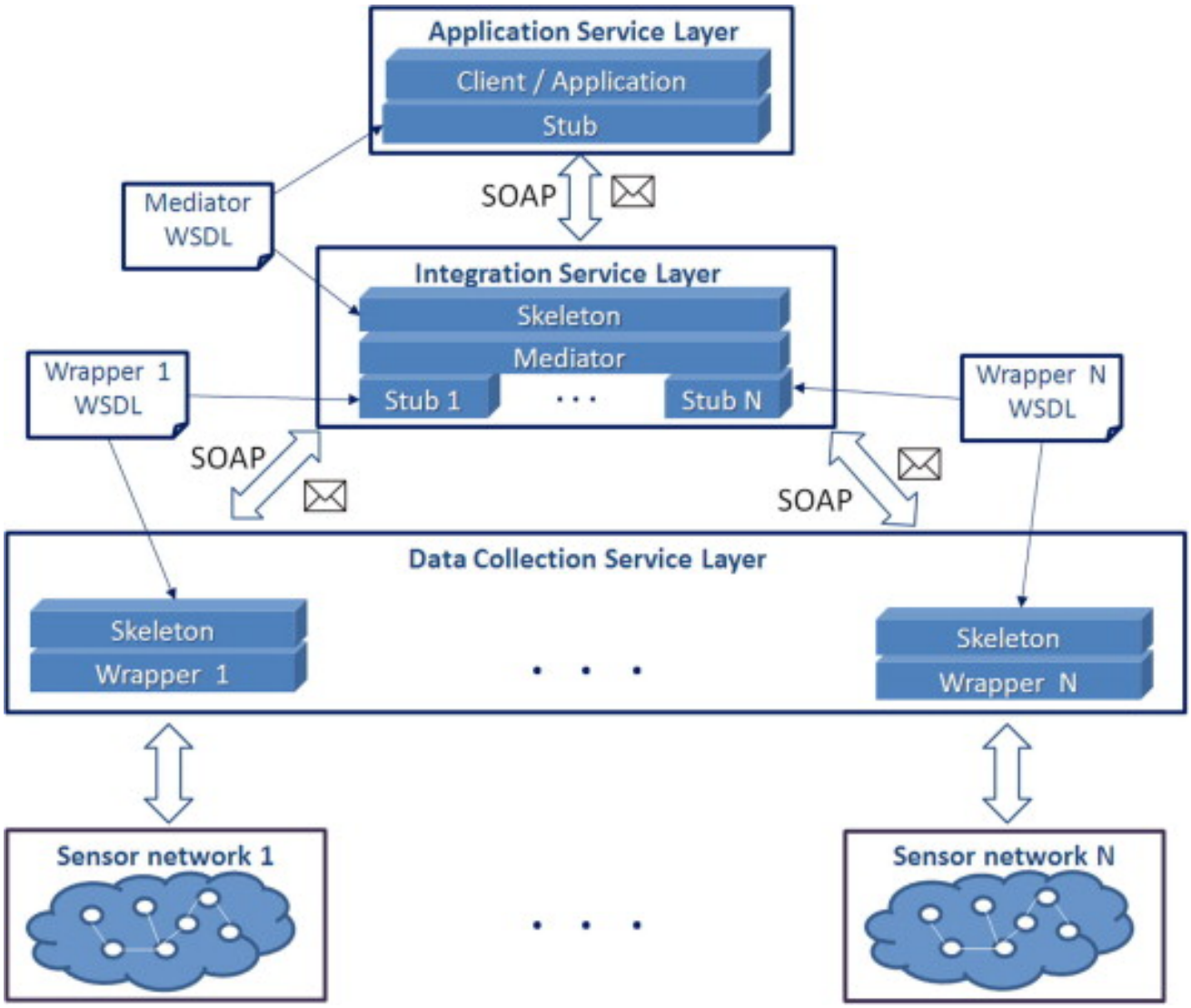


Fig. 3 Livelli di raccolta e integrazione dell'architettura InsistMod.

Il mediatore fornisce il livello di integrazione dei servizi e i suoi servizi vengono invocati da un utente generico o da un'applicazione in base all'interfaccia WSDL e al protocollo SOAP. Il componente mediatore ha lo scopo di classificare le caratteristiche delle reti, nonché di formattare e inoltrare le query a wrapper specifici; un DBMS è utilizzato per memorizzare i dati relativi alle reti con i loro sensori, le query e i relativi risultati. Per le comunicazioni tra il mediatore e i wrapper, in base ai servizi forniti, si utilizza anche SOAP.

La Fig. 4 mostra tutti i servizi forniti dal mediatore e dai wrapper e le loro interazioni.

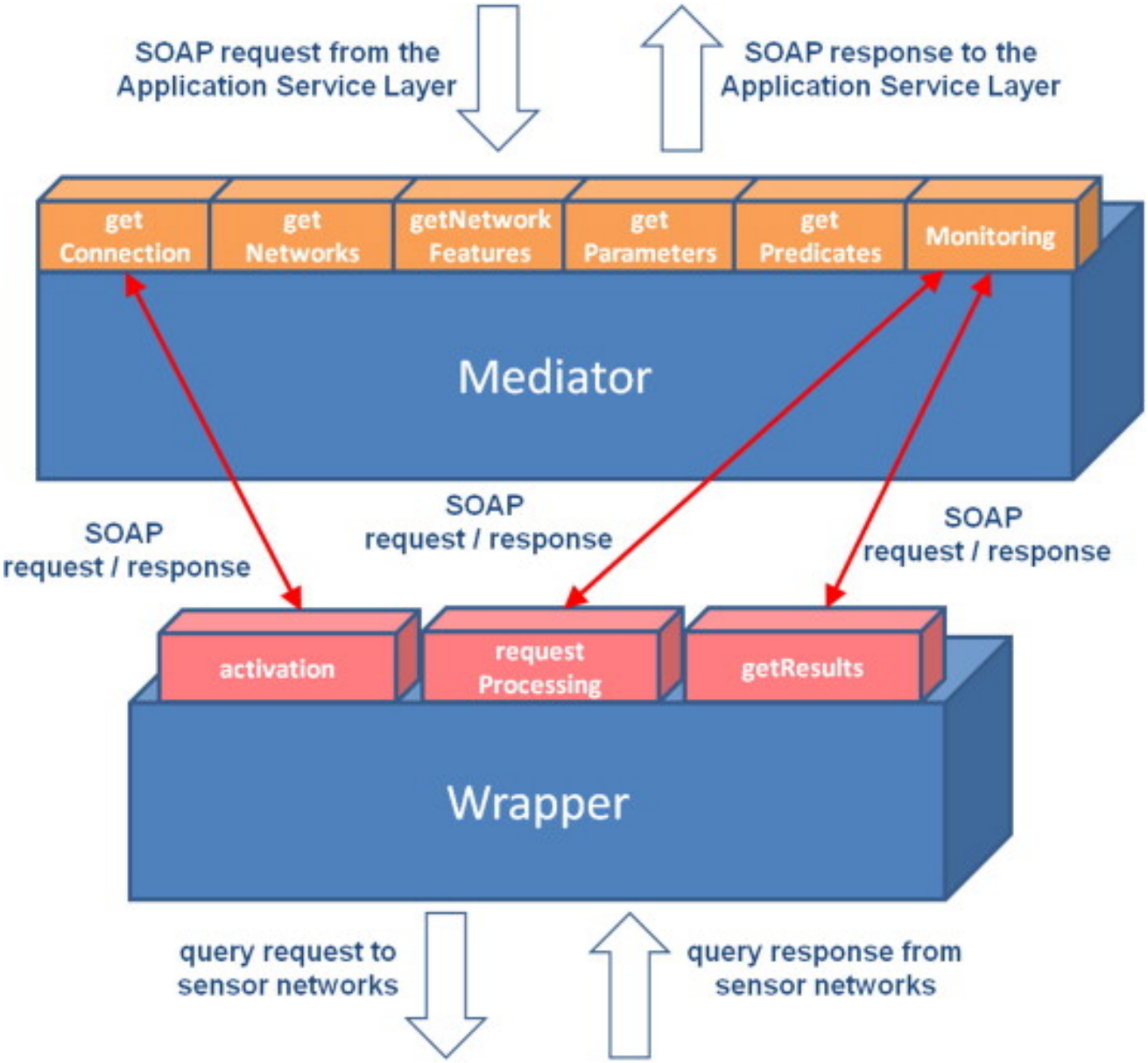


Fig.4 Servizi di raccolta e integrazione

Di seguito, verrà riportata una breve descrizione dei servizi del mediatore e nelle Fig. 5, 6 e 7 sono riportati alcuni diagrammi di sequenza UML illustrativi per mostrare l'interazione con i servizi del client e del wrapper:

1. getConnection consente la scoperta e la registrazione di uno specifico wrapper. A questo scopo, invoca il servizio di attivazione fornito dal wrapper (si veda il diagramma di sequenza in Fig. 5).

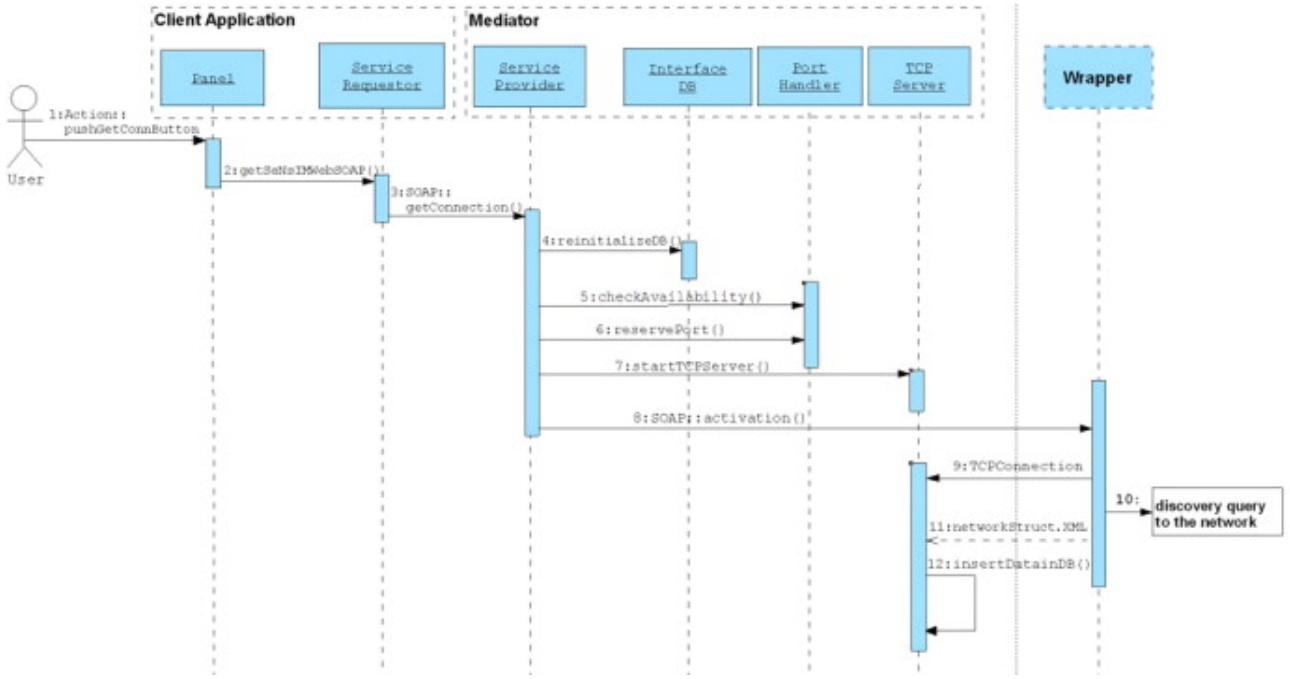


Fig. 5 Servizio getConnection.

2. getNetworks fornisce come output l'elenco di tutte le reti connesse al sistema, la loro topologia e le informazioni di base come gli identificatori di rete, il numero di sensori e la profondità massima per ogni rete.

3. getNetworkFeatures riceve come parametro di input l'identificativo della rete e fornisce in output: descrizione del sistema, tipo di middleware, numero di sensori, profondità massima. Inoltre, fornisce informazioni sul wrapper che gestisce la rete (indirizzo IP, tempo di registrazione e porte di comunicazione). Infine, fornisce informazioni sulla stazione base, la sua frequenza di lavoro e il suo collegamento di comunicazione con il wrapper (Fig. 6).

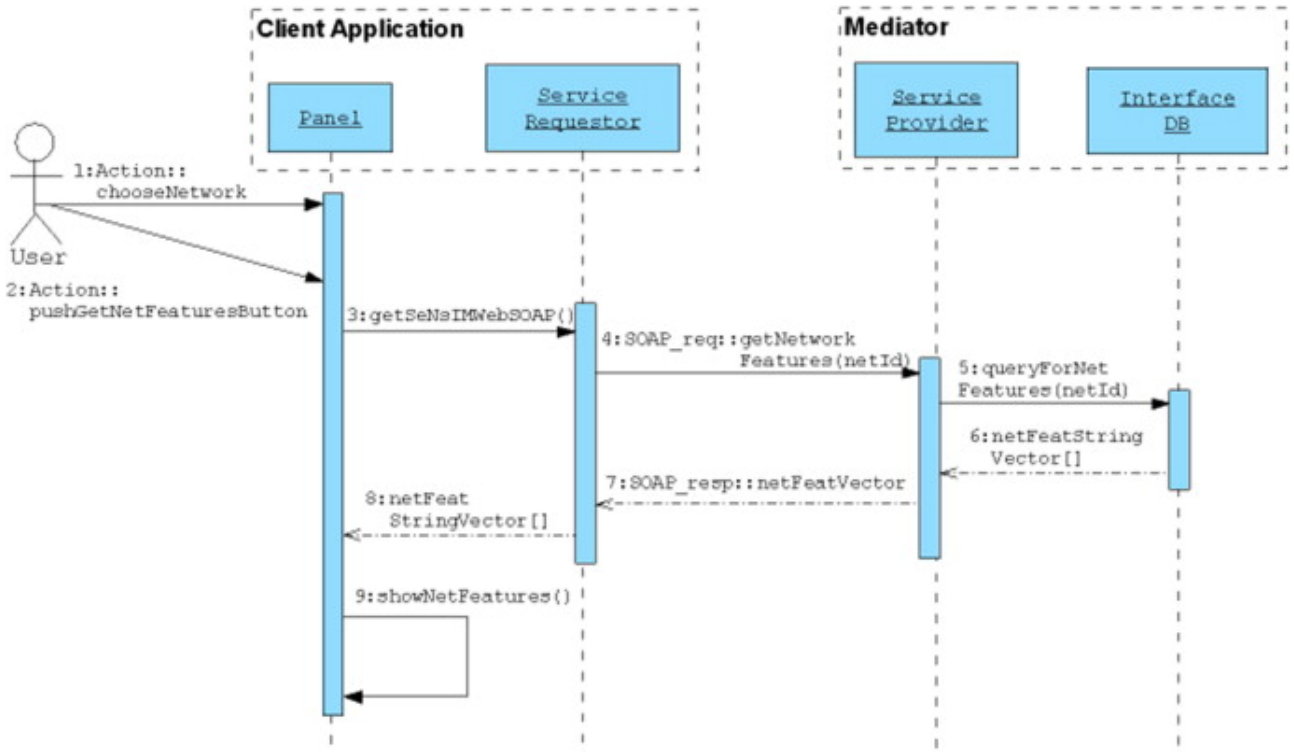


Fig.6 Servizio getNetworkFeatures.

4. getParameters riceve come input l'identificazione di un sensore specifico dato dalla tripla: identificatore di rete, identificatore di cluster (se definito) e identificatore del sensore. In uscita fornisce i parametri intrinseci del sensore, come la memoria disponibile, la tensione e la qualità del canale.

5. getPredicates riceve in ingresso l'identificazione di un sensore specifico e fornisce in uscita i predicati del sensore, cioè la variabile fisica misurata dal sensore.

6. Monitoring esegue un semplice compito di monitoraggio interrogando un sensore o un'intera rete. Riceve in ingresso i parametri dell'interrogazione: identificazione del sensore, periodo di campionamento, durata, intervallo di recupero, già definito in InsistMod [24] come l'intervallo di tempo in cui i wrapper possono raccogliere i risultati dell'interrogazione prima di inviarli al

mediatore. Come mostrato in Fig. 7, il processo di interrogazione può iniziare invocando il servizio requestProcessing fornito dal wrapper. I risultati vengono inviati al mediatore in un messaggio di risposta SOAP tramite il servizio wrapper getResults, che viene invocato nuovamente dal servizio Monitoring. Infine, i dati vengono memorizzati nel database del mediatore e mostrati all'utente.

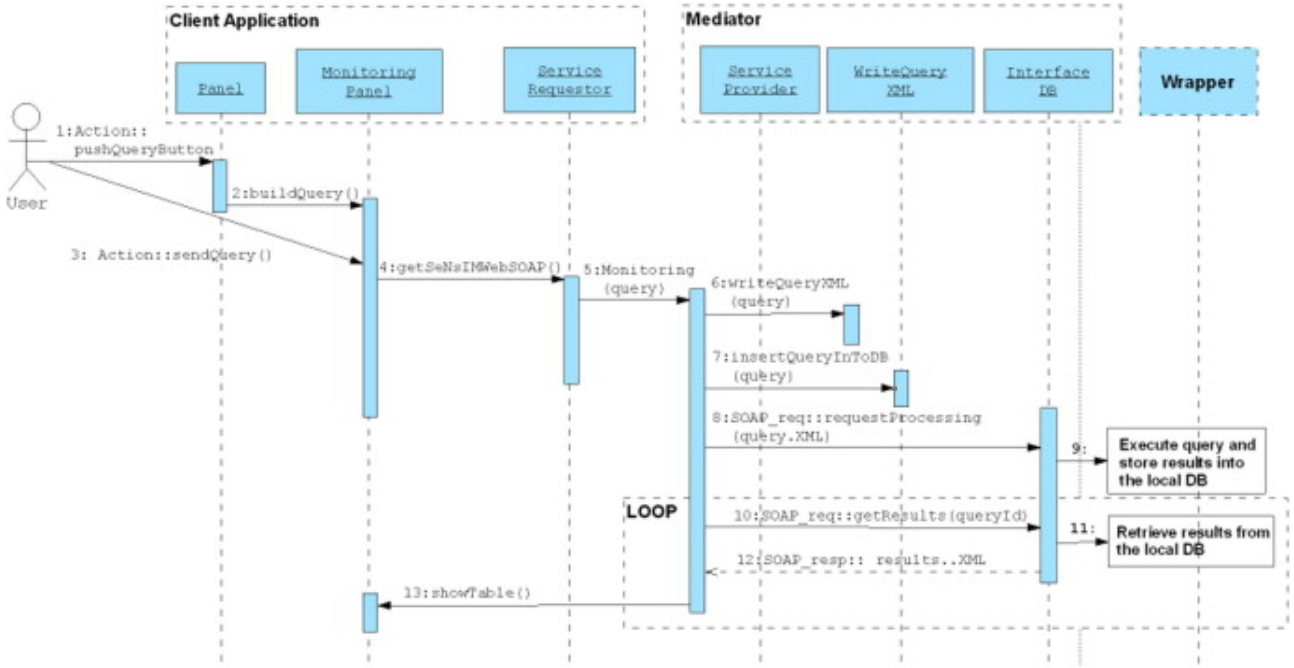


Fig. 7 Servizio Monitoring.

Dal lato del wrapper, i servizi forniti sono:

1. attivazione, invocata dal servizio getConnection, consente la registrazione di uno specifico wrapper all'interno del sistema. In particolare, il wrapper inietta una query di scoperta alla rete sottostante e successivamente costruisce una descrizione della rete che viene rinviata al mediatore (Fig. 5).

2. requestProcessing, invocato dal servizio di monitoraggio, ha lo scopo di inviare query alla rete sottostante.

3. getResults, invocato dal servizio di monitoraggio, consente di ottenere i risultati relativi a una specifica query (Fig. 7).

Come già detto, ogni wrapper del Data Collection Service Layer ha lo scopo di fornire i meccanismi necessari per l'interoperabilità tra reti eterogenee e di fornire un significato alle osservazioni dei sensori per consentire la consapevolezza della situazione. In particolare, consente a qualsiasi applicazione di accedere ai dati di reti di sensori eterogenee e di tradurre il formato dei dati proprietari nel modello di dati comune definito. Per raggiungere questo obiettivo sono necessari diversi moduli (vedi Fig. 8):

1. Modulo aggiuntivo per i sensori (aggregatore di dati): questo modulo è specifico per ogni tipo di rete di sensori. Recupera i campioni dai sistemi di sensori e li converte in un formato XML, secondo le regole di traduzione definite.

2. XML-RDF Wrapper: questo modulo avvolge gli insiemi di dati XML, traducendo il file in un documento RDF, secondo un insieme di ontologie del dominio dei sensori, per arricchire i dati con relazioni semantiche implicite.

3. O&M Data modeler: l'ultimo modulo recupera un documento RDF e lo converte in un documento XML standard O&M.

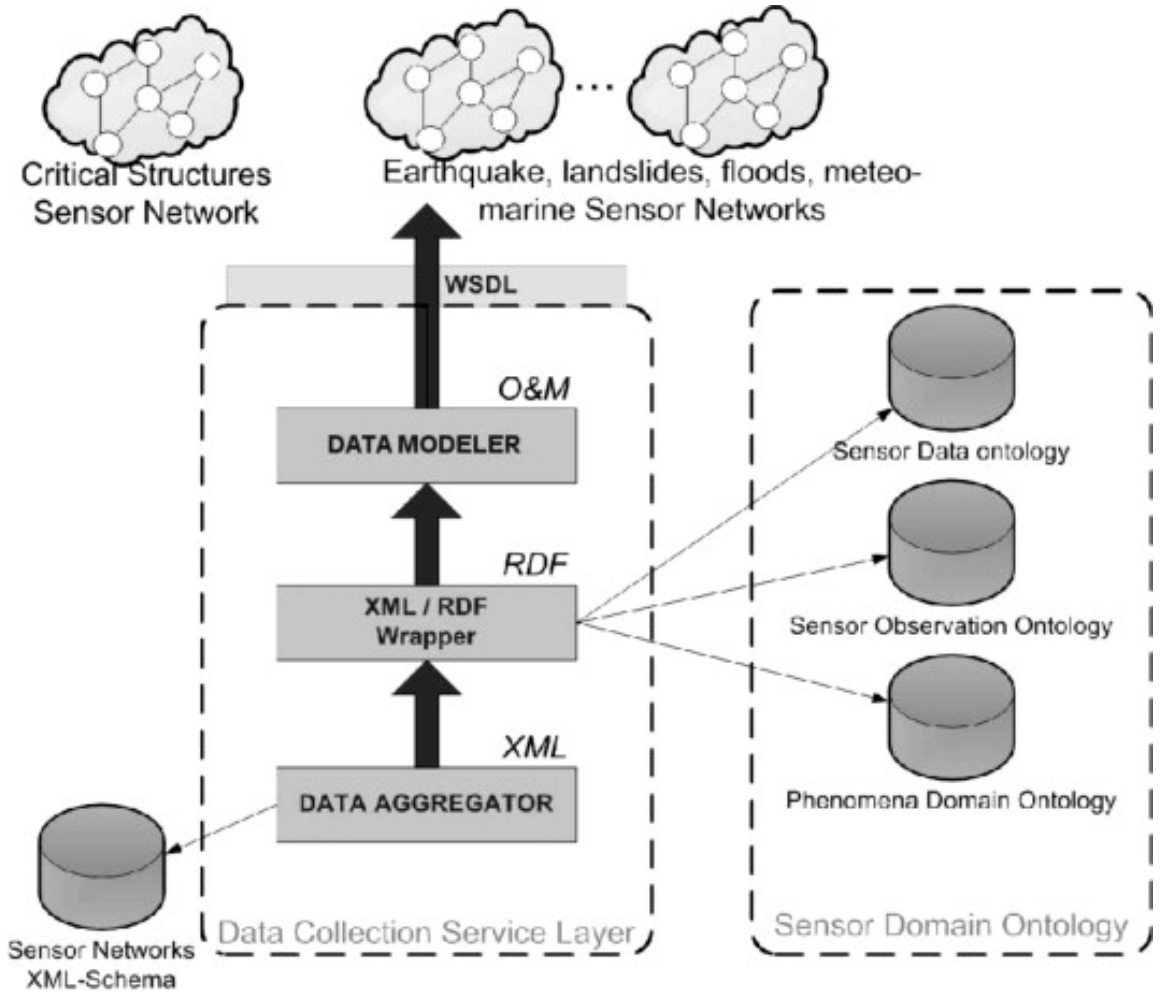


Fig. 8 Livello raccolta dati.

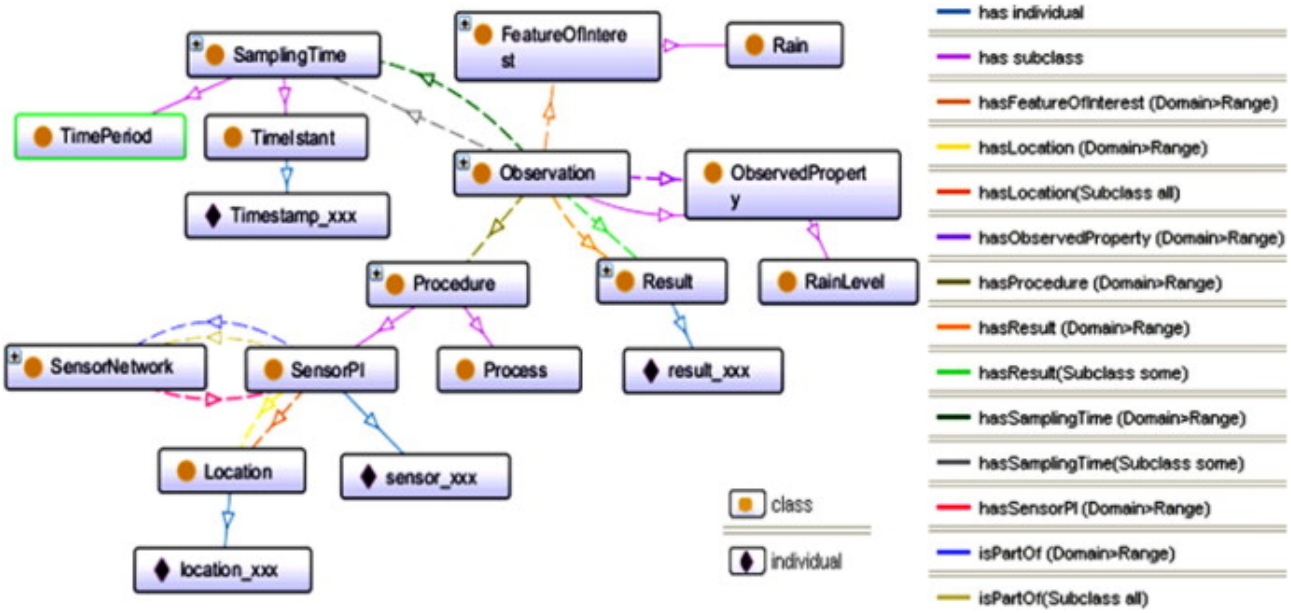


Fig. 9 Ontologia dei sensori codificato in schema RDFS.

Per spiegare il processo di trasformazione eseguito da questo modulo, ne illustreremo i passaggi attraverso un esempio concreto; in particolare, considereremo una rete di sensori pluviometrici per misurare il livello della pioggia.

Il modulo aggregatore di dati acquisisce i dati grezzi dei sensori da reti di sensori eterogenee e li codifica in un documento XML. La mappatura viene effettuata utilizzando uno schema XML definito per le reti di sensori e memorizzato nel Repository XML-Schema.

Lo schema XML è molto semplice; nel Fig.10 è riportato un esempio di dati rilevati codificati secondo questo schema; esso descrive le coordinate del sensore, espresse secondo i Datatypes di riferimento del W3C (tag origin), l'ID del mittente, il timestamp e il valore misurato espresso come float.

```
<Pluviometric xsi:noNamespaceSchemaLocation="Pluviometric.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <ID>2147483647</ID>
    <Network>String</Network>
    <Origin>
        <X_Utm>2147483647</X_Utm>
        <Y_Utm>2147483647</Y_Utm>
        <Z_999>2147483647</Z_999>
    </Origin>
    <Sample>
        <Timestamp>2009-01-26T21:42:52</Timestamp>
        <Level>3.14159E0</Level>
    </Sample>
</Pluviometric>
```

Fig. 10 Esempio di frammento di misura codificato XML.

4 ATTIVITÀ Descrizione degli Algoritmi di analisi descrittiva e cognitiva e dei relativi input

4.1 Analisi di Dati Eterogenei

I dati eterogenei sono dati con un'elevata variabilità di tipi e formati di dati. Possono essere ambigui e di bassa qualità a causa di valori mancanti, elevata ridondanza di dati e non veridicità. È difficile integrare i dati eterogenei per soddisfare le richieste di informazioni aziendali. Ad esempio, i dati eterogenei sono spesso generati dall'Internet degli oggetti (IoT). I dati generati dall'IoT presentano spesso le seguenti caratteristiche [1, 44]. In primo luogo, sono eterogenei. A causa della varietà dei dispositivi di acquisizione dei dati, anche i dati acquisiti sono di tipo diverso, con una certa eterogeneità. In secondo luogo, sono su larga scala. Le apparecchiature di acquisizione dei dati vengono utilizzate e distribuite in modo massiccio, per cui è necessario memorizzare non solo i dati acquisiti al momento, ma anche i dati storici entro un certo lasso di tempo. In terzo luogo, esiste una forte correlazione tra tempo e spazio. Ogni dispositivo di acquisizione dati è collocato in una specifica posizione geografica e ogni dato ha una marca temporale. La correlazione temporale e spaziale è una proprietà importante dei dati dell'IoT. In quarto luogo, i dati effettivi rappresentano solo una piccola parte dei big data. Durante l'acquisizione e la trasmissione dei dati nell'IoT può essere raccolta una grande quantità di rumori. Tra i set di dati acquisiti dai dispositivi di acquisizione, solo una piccola quantità di dati è preziosa. Esistono i seguenti tipi di eterogeneità dei dati [2, 44]:

- L'eterogeneità sintattica si verifica quando due fonti di dati non sono espresse nello stesso Linguaggio.
- L'eterogeneità concettuale, nota anche come eterogeneità semantica o disallineamento logico, indica le differenze nella modellazione dello stesso dominio di interesse.
- L'eterogeneità terminologica indica le variazioni nei nomi quando si fa riferimento alle stesse entità da fonti di dati diverse.
- L'eterogeneità semiotica, nota anche come eterogeneità pragmatica, indica la diversa interpretazione delle entità da parte delle persone.

La rappresentazione dei dati può essere descritta a quattro livelli [3]. Il livello 1 è costituito da dati grezzi diversi, di tipo diverso e provenienti da fonti diverse. Il livello 2 è chiamato "rappresentazione unificata". I dati eterogenei devono essere unificati. Inoltre, una quantità eccessiva di dati può

comportare elevati costi cognitivi e di elaborazione dei dati. Questo livello converte i singoli attributi in informazioni in termini di "cosa-quando-dove". Il livello 3 è l'aggregazione. I dati territoriali possono essere naturalmente rappresentati sotto forma di griglie spaziali con attributi tematici. Gli operatori di elaborazione sono la segmentazione e l'aggregazione, ecc. L'aggregazione facilita la visualizzazione e fornisce un'interrogazione intuitiva. Il livello 4 è chiamato "rilevamento e rappresentazione della situazione". La situazione in un luogo viene caratterizzata in base ai descrittori spazio-temporali determinati utilizzando gli operatori appropriati del livello 3. La fase finale del rilevamento della situazione è l'individuazione dei descrittori spazio-temporali. La fase finale del rilevamento della situazione è un'operazione di classificazione che utilizza la conoscenza del dominio per assegnare una classe appropriata a ciascuna cella.

I metadati sono fondamentali per le future interrogazioni. Per le tabelle relazionali e alcuni documenti Extensible Markup Language (XML), le definizioni esplicite dello schema in Structured Query Language (SQL), XML Schema Definition (XSD) o Document Type Definition (DTD) possono essere ottenute direttamente dalle fonti e integrate in un metamodello. La tecnica XML viene utilizzata per la traduzione dei dati. La parte difficile sono i dati semi-strutturati (come XML senza XSD, JSON o file Excel o CSV parzialmente strutturati) che contengono schemi impliciti. Pertanto, il componente Structural Metadata Discovery (SMD) si assume la responsabilità di scoprire i metadati impliciti (ad esempio, tipi di entità, tipi di relazioni e vincoli) dai dati semi-strutturati [4,5]. I problemi di gestione dei metadati sono importanti. Per un'interpretazione appropriata di big data eterogenei, sono necessari metadati dettagliati. Alcuni report contengono alcuni metadati, ma a scopo di ricerca sono necessari molti più dettagli, ad esempio sul sensore specifico utilizzato per la raccolta dei dati. La raccolta dei metadati e della provenienza dei dati è una sfida importante quando i dati vengono raccolti sotto pressione e in situazioni di stress [6].

Le sfide degli algoritmi per i Big Data si concentrano sulla progettazione di algoritmi per affrontare le difficoltà sollevate dai grandi volumi di dati, dalle distribuzioni distribuite dei dati e dalle caratteristiche complesse e dinamiche dei dati. Le sfide includono le seguenti fasi. In primo luogo, i dati eterogenei, incompleti, incerti, scarsi e provenienti da più fonti vengono pre-elaborati con tecniche di fusione dei dati. In secondo luogo, i dati dinamici e complessi vengono estratti dopo la pre-elaborazione. In terzo luogo, la conoscenza globale ottenuta dall'apprendimento locale e dalla fusione di modelli viene testata e le informazioni rilevanti vengono riportate alla fase di pre-elaborazione. Quindi il modello e i parametri vengono regolati in base al feedback. Nell'intero processo, la condivisione delle informazioni non è solo una promessa di sviluppo regolare di ogni fase, ma anche uno scopo dell'elaborazione dei big data [7].

La presente attività si concentra su quattro aspetti: 1) introduce i metodi di elaborazione dei dati, tra cui la pulizia dei dati, l'integrazione dei dati, la riduzione delle dimensioni e la normalizzazione dei dati per i dati eterogenei e l'analisi dei Big Data; 2) presenta i concetti di Big Data, l'analisi dei Big Data e gli strumenti per i Big Data; 3) confronta i metodi DM/ML tradizionali con l'apprendimento profondo, in particolare la loro fattibilità nell'analisi dei Big Data; 4) discute il potenziale delle confluenze tra l'analisi dei Big Data, l'apprendimento profondo, l'HPC e l'elaborazione eterogenea.

4.2 Metodi di elaborazione dei dati per dati eterogenei e Big Data Analytics

4.2.1. Pulizia dei dati

La pulizia dei dati è un processo volto a identificare i dati incompleti, inaccurati o irragionevoli e a modificarli o eliminarli per migliorarne la qualità [1]. Ad esempio, la natura multi-fonte e multi-modale dei dati sanitari comporta un'elevata complessità e problemi di rumore. Inoltre, esistono anche problemi di valori mancanti e impurità nei dati ad alto volume. Poiché la qualità dei dati determina la qualità delle informazioni, che finirà per influenzare il processo decisionale, è fondamentale sviluppare approcci efficienti di pulizia dei big data per migliorare la qualità dei dati per prendere decisioni accurate ed efficaci [8].

Un valore mancante per una variabile è un valore che non è stato inserito in un set di dati, ma di cui esiste un valore effettivo [9]. Spesso si ricorre all'imputazione semplice (non stocastica). Nell'imputazione semplice, i valori mancanti di una variabile vengono sostituiti con un singolo valore (ad esempio, media, mediana o modalità). Tuttavia, l'imputazione semplice produce risultati distorti per i dati che non sono completamente mancanti a caso (MCAR). In presenza di una quantità moderata o elevata di dati mancanti, è probabile che l'imputazione semplice sottovaluti gli errori standard, distorca le correlazioni tra le variabili e produca valori di p errati nei test statistici. Questo approccio dovrebbe essere evitato per la maggior parte dei problemi di dati mancanti [10]. Lo studio delle correlazioni lineari ha permesso di riempire alcuni nuovi valori sconosciuti. Si può cercare di esplorare le correlazioni tra le variabili con incognite e le variabili nominali di questo problema. I valori sconosciuti possono essere riempiti esplorando correlazioni più accurate. Quando si gestisce un insieme di dati con valori mancanti, si possono seguire diverse strategie. Le più comuni sono: 1) rimuovere i casi con incognite; 2) riempire i valori sconosciuti esplorando la somiglianza tra i casi; 3) riempire i valori sconosciuti esplorando le correlazioni tra le variabili; 4) utilizzare strumenti in grado di gestire questi valori [11].

Un database può anche contenere attributi irrilevanti. Pertanto, l'analisi della rilevanza, sotto forma di analisi delle correlazioni e selezione di sottoinsiemi di attributi, può essere utilizzata per

individuare gli attributi che non contribuiscono alla classificazione o alla previsione. L'inclusione di tali attributi potrebbe rallentare ed eventualmente fuorviare la fase di apprendimento. In genere, la pulizia e l'integrazione dei dati vengono eseguite come fase di pre-elaborazione. Le incoerenze nella denominazione degli attributi o delle dimensioni possono causare ridondanze nel set di dati risultante. La pulizia dei dati può essere eseguita per individuare e rimuovere le ridondanze che possono derivare dall'integrazione dei dati. La rimozione dei dati ridondanti è spesso considerata come una delle principali operazioni di pulizia e riduzione dei dati [12].

4.2.2 Integrazione dei dati

Nel caso dell'integrazione o dell'aggregazione dei dati, le serie di dati vengono abbinate e unite sulla base di variabili e attributi condivisi. Le tecniche avanzate di elaborazione e analisi dei dati consentono di combinare dati strutturati e non strutturati per ottenere nuove conoscenze; tuttavia, ciò richiede dati "puliti". Le tecniche di fusione dei dati sono utilizzate per abbinare e aggregare insiemi di dati eterogenei per creare o migliorare una rappresentazione della realtà che aiuti il data mining. Le metodologie di fusione dei dati di medio livello, che uniscono dati strutturati e prodotti dalle macchine, funzionano sostanzialmente bene. D'altra parte, i compiti di fusione dei dati di alto livello per unire più input analogici non strutturati dei sensori rimangono impegnativi [13].

Gli strumenti di integrazione dei dati si stanno evolvendo verso l'unificazione dei dati strutturati e non strutturati e inizieranno a includere funzionalità semantiche. Spesso è necessario strutturare i dati non strutturati e unire fonti e tipi di informazioni eterogenee in un livello di dati unificato. La maggior parte delle piattaforme di integrazione dei dati utilizza un modello di integrazione primario basato su tipi di dati relazionali o XML. Sono state proposte piattaforme avanzate di virtualizzazione dei dati che utilizzano un modello di dati di integrazione esteso con la capacità di memorizzare e leggere/scrivere tutti i tipi di dati nel loro formato nativo, come file relazionali, multidimensionali, semantici, gerarchici e indici, ecc. [14].

L'integrazione di fonti di dati eterogenee è impegnativa. Uno dei motivi è che spesso non esistono identificatori unici tra i record di due insiemi di dati diversi. Determinare quali dati debbano essere uniti può non essere chiaro fin dall'inizio. Lavorare con dati eterogenei è spesso un processo iterativo in cui il valore dei dati viene scoperto lungo il percorso e i dati più preziosi vengono poi integrati con maggiore attenzione [15]. Per l'eterogeneità dei dati, è stata proposta la seguente integrazione [2]: 1) integrazione degli schemi - la fase essenziale del processo di integrazione degli schemi consiste nell'identificare le corrispondenze tra entità semanticamente identiche degli schemi; 2) integrazione dei cataloghi - nelle applicazioni Business-to-Business (B2B), i partner commerciali archiviano le

informazioni sui loro prodotti in cataloghi elettronici. La ricerca di corrispondenze tra le voci dei cataloghi è un problema di corrispondenza tra cataloghi.

L'acquisizione di conoscenza da fonti di dati autonome, semanticamente eterogenee e distribuite, gli approcci query-centrici e federati all'integrazione dei dati sono di particolare interesse [5]. Per l'integrazione di dati strutturati e non strutturati si possono utilizzare i seguenti approcci [16]:

- Pipeline di elaborazione del linguaggio naturale: L'elaborazione del linguaggio naturale (NLP) può essere applicata direttamente a progetti che richiedono la gestione di dati non strutturati.

- Riconoscimento e collegamento delle entità: L'estrazione di informazioni strutturate da dati non strutturati è una fase fondamentale. Una parte del problema può essere risolta con tecniche di estrazione delle informazioni come il riconoscimento di entità, l'estrazione di relazioni e l'estrazione di ontologie. Questi strumenti aiutano a costruire automaticamente la conoscenza semi-strutturata. Esistono framework maturi per alcune classi di problemi di estrazione dell'informazione, anche se la loro adozione rimane limitata agli early-adopters.

- Uso dei dati aperti per integrare dati strutturati e non strutturati: Le entità presenti negli insiemi di dati aperti possono essere utilizzate per identificare entità denominate (persone, organizzazioni, luoghi), che possono essere utilizzate per categorizzare e organizzare i contenuti del testo. Gli strumenti di riconoscimento e collegamento delle entità denominate, come DBpedia Spotlight, possono essere utilizzati per collegare dati strutturati e non strutturati.

Quando si mettono insieme dati provenienti da sistemi eterogenei, ci sono tre fonti di errori nei dati: errori di inserimento, incompatibilità di tipo di dati e incompatibilità semantica nelle definizioni delle entità aziendali. Tradizionalmente le aziende utilizzavano l'ETL (Extract, Transform and Load) e i data warehouse (DW) per l'integrazione dei dati. Negli ultimi anni, tuttavia, una tecnologia nota come "Data Virtualization (DV)" si è affermata come soluzione alternativa per l'integrazione dei dati. La "virtualizzazione dei dati" è un database federato definito database composito. La virtualizzazione dei dati e la standardizzazione dei dati aziendali promettono di ridurre i costi e i tempi di implementazione dell'integrazione dei dati. A differenza del DW, il DV definisce la pulizia dei dati, le giunzioni e le trasformazioni dei dati in modo programmatico utilizzando le viste logiche. Il DV consente l'estensibilità e il riutilizzo grazie alla possibilità di concatenare le viste logiche. La standardizzazione dei dati aziendali evita soprattutto le discrepanze tra i tipi di dati e le incompatibilità semantiche dei dati. Il DV non sostituisce il DW; il DV potrebbe scaricare alcuni carichi di lavoro analitici dal DW. L'analisi di regressione, le strutture di dati multidimensionali e l'analisi di grandi quantità di dati richiedono principalmente il DW [17].

I data lake sono un approccio emergente e potente alle sfide dell'integrazione dei dati, in quanto le aziende aumentano la loro esposizione alle applicazioni mobili e basate su cloud e all'Internet delle cose (IoT) guidato dai sensori. I data lake sono archivi per grandi quantità e varietà di dati, sia strutturati che non strutturati. I data lake sono più adatti per i dati meno strutturati che le aziende devono elaborare. Tuttavia, le difficoltà associate alle sfide di integrazione dei data lakes includono, ma non solo, i seguenti aspetti: 1) sviluppare una gestione avanzata dei metadati sui dati grezzi estratti da fonti di dati eterogenee; 2) gestire i metadati strutturali delle fonti di dati e annotare i dati e i metadati con informazioni semantiche per evitare ambiguità. Senza metadati o gestione dei metadati, scaricare tutti i dati in un data lake porterebbe a una "palude di dati" e il data lake sarebbe difficilmente utilizzabile perché la struttura e la semantica dei dati non sono note [18].

4.3 Riduzione della dimensione e normalizzazione dei dati

Ci sono diverse ragioni per ridurre la dimensionalità dei dati. In primo luogo, i dati ad alta dimensionalità impongono sfide computazionali. In secondo luogo, l'elevata dimensionalità può portare a scarse capacità di generalizzazione dell'algoritmo di apprendimento in alcune situazioni (ad esempio, la complessità del campione aumenta esponenzialmente con la dimensione nei classificatori nearest neighbour). Infine, la riduzione della dimensionalità può essere utilizzata per trovare una struttura significativa dei dati, per la loro interpretabilità e per scopi illustrativi [19].

La selezione di sottoinsiemi di caratteristiche (FSS) è un compito ben noto del data mining e dell'apprendimento automatico. Gli algoritmi genetici (GA), Hill Climbing e Simulated Annealing, ecc. sono comunemente utilizzati per la selezione di sottoinsiemi di caratteristiche. La riduzione della dimensionalità operata da un processo FSS può fornire diversi vantaggi: 1) una più rapida induzione del modello di classificazione finale, 2) un miglioramento della comprensibilità del modello di classificazione finale e 3) un miglioramento dell'accuratezza della classificazione [20]. Le tecniche di selezione delle caratteristiche possono essere suddivise in due approcci: la classificazione delle caratteristiche e la selezione dei sottoinsiemi. Nel primo approccio, le caratteristiche vengono classificate in base ad alcuni criteri e poi vengono selezionate quelle che superano una soglia definita. Nel secondo approccio, si cerca in uno spazio di sottoinsiemi di caratteristiche il sottoinsieme ottimale. Inoltre, il secondo approccio può essere suddiviso in tre parti: 1) approcci a filtro - le persone selezionano prima le caratteristiche e poi utilizzano questo sottoinsieme per eseguire un algoritmo di classificazione; 2) approcci incorporati - la selezione delle caratteristiche avviene come parte di un

algoritmo di classificazione; e 3) approcci wrapper - un algoritmo di classificazione viene utilizzato sul set di dati per identificare le caratteristiche migliori [21].

Negli insiemi di dati con un gran numero di variabili, di solito c'è molta sovrapposizione nelle informazioni coperte dall'insieme delle variabili. Un modo semplice per trovare le ridondanze è controllare la matrice di correlazione ottenuta con l'analisi delle correlazioni [22]. L'analisi dei fattori è un metodo di riduzione della dimensionalità. È utile per comprendere le ragioni sottostanti alle correlazioni tra un gruppo di variabili. L'analisi fattoriale può essere utilizzata per ridurre il numero di variabili e rilevare la struttura delle relazioni tra le variabili. Pertanto, l'analisi fattoriale viene spesso utilizzata come metodo di rilevamento della struttura o di riduzione dei dati [23]. Inoltre, la PCA è utile quando ci sono dati su un gran numero di variabili ed è possibile che ci sia una certa ridondanza in queste variabili. In questa situazione, ridondanza significa che alcune variabili sono correlate tra loro [21]. La PCA è molto veloce, efficace, semplice e ampiamente utilizzata. Ci sono diversi modi in cui la PCA può essere utile [24]:

- Pre-elaborazione: L'apprendimento di modelli complessi di dati ad alta dimensionalità è spesso molto lento e soggetto a overfitting. Il numero di parametri di un modello è solitamente esponenziale rispetto al numero di dimensioni. Con la PCA si può anche sbiancare la rappresentazione, riequilibrando i pesi dei dati per ottenere, in alcuni casi, prestazioni migliori.
- Modellazione: La PCA apprende una rappresentazione che a volte viene utilizzata come un modello completo, ad esempio una distribuzione precedente per i nuovi dati.
- Compressione: La PCA può essere utilizzata per comprimere i dati, sostituendoli con la loro rappresentazione a bassa dimensione.

I passaggi chiave per l'utilizzo della PCA o dell'EFA (Analisi fattoriale esplorativa) sono i seguenti: 1) preparare i dati, come ad esempio controllare che non ci siano valori mancanti; 2) selezionare un modello fattoriale, decidendo se la PCA (riduzione dei dati) o l'EFA (scoperta della struttura latente) sia più adatta agli obiettivi della ricerca e scegliendo un metodo di fattorizzazione specifico (ad esempio, la massima verosimiglianza) se si sceglie un approccio EFA; 3) decidere quante componenti/fattori estrarre; e 4) estrarre le componenti/fattori. Per quanto riguarda la scelta del numero di componenti da estrarre, sono disponibili diversi criteri per decidere quante componenti mantenere in una PCA. Tra questi vi sono: 1) basare il numero di componenti su esperienze precedenti e sulla teoria; 2) selezionare il numero di componenti necessario per rappresentare una certa soglia di varianza cumulativa delle variabili (ad esempio, l'80%); 3) selezionare il numero di componenti da mantenere esaminando gli autovalori della matrice di correlazione tra le variabili [10].

Alcuni algoritmi richiedono che i dati siano normalizzati (standardizzati) prima che l'algoritmo possa essere efficacemente implementato. Normalizzazione (o standardizzazione) significa sostituire ogni variabile originale con una versione standardizzata della variabile che ha varianza unitaria. L'effetto di questa normalizzazione (standardizzazione) è quello di dare a tutte le variabili la stessa importanza in termini di variabilità. I dati vengono spesso normalizzati prima di eseguire la PCA [22].

4.4 Big Data, Big Data Analytics e strumenti per i Big Data

Esistono delle lacune tra i Big Data e le attuali capacità di analisi dei dati nei sistemi di auditing continuo (CA). In particolare, le tre V originali (grande volume, alta velocità e grande varietà) introducono le lacune della coerenza dei dati, dell'identificazione dei dati e dell'aggregazione dei dati per collegare i database in un sistema di CA. Ognuna di queste lacune crea le sfide corrispondenti nella Tabella 1. L'identificazione dei dati si riferisce ai record che collegano due o più informazioni registrate separatamente sullo stesso individuo o entità. Quando i dati sono strutturati, l'identificazione è facile. Tuttavia, l'identificazione diventa difficile nell'audit dei big data, dove molti dati potrebbero essere non strutturati. La mancanza di integrità dei dati è solitamente legata a manomissioni e dati incompleti. La coerenza dei dati è il problema più importante per la CA dei big data e riguarda l'interdipendenza dei dati tra le applicazioni e all'interno di un'organizzazione. La riservatezza dei dati significa che alcuni dati o le associazioni tra i punti di dati sono sensibili e non possono essere divulgati ad altri. Nell'era dei big data, i dati possono essere facilmente associati ad altri dati. Una volta che alcuni dati sensibili sono trapelati, possono propagarsi ad alta velocità e collegarsi a una grande quantità di dati correlati. Pertanto, la riservatezza dei big data diventa ancora più urgente e importante per preservare l'immagine del marchio e garantire il vantaggio competitivo. L'aggregazione dei dati è necessaria per il normale funzionamento dell'audit continuo che utilizza i big data per riassumere e semplificare in modo significativo i big data che molto probabilmente provengono da fonti diverse [25].

Tabella 1. L'analisi dei Big Data nell'Audit continuo [25]

Caratteristiche dei Big Data:	• Volume • Varietà • Velocità • Veridicità • Valore
Lacune nei Big Data	• Identificazione dei dati • Integrità dei dati • Coerenza dei dati • Riservatezza dei dati • Aggregazione dei dati
Le sfide dell'audit continuo	• Dati contrastanti • Dati incompleti • Dati con varie identificazioni • Dati con formati diversi • Dati asincroni • Dati manomessi illegalmente • Ricerca di dati criptati - Verifica dei dati criptati • Verifica dei dati aggregati

L'eterogeneità è una delle caratteristiche essenziali dei big data. I dati provenienti da fonti diverse e varie possiedono intrinsecamente molti tipi e forme di rappresentazione differenti e possono essere interconnessi, correlati e rappresentati in modo incoerente. L'eterogeneità dei big data implica anche l'obbligo di acquisire e trattare simultaneamente dati strutturati, semi-strutturati e persino interamente non strutturati [26]. I big data mettono in relazione insiemi di dati complessi e di grande volume con molteplici fonti indipendenti. L'analisi dei big data può essere problematica perché spesso comporta la raccolta e l'archiviazione di dati misti basati su modelli o regole diverse (dati misti eterogenei) [27]. È importante aumentare la consapevolezza del contesto. Ad esempio, i dati esistenti nel settore manifatturiero non hanno alcuna relazione con il contesto della storia, degli orari, delle abitudini, delle attività e della posizione degli utenti, ecc. Nel contesto dei big data, la contestualizzazione può essere un paradigma interessante per combinare flussi di dati eterogenei per migliorare la qualità di un processo di estrazione o di un classificatore. Inoltre, la consapevolezza del contesto ha dimostrato di essere utile per ridurre il consumo di risorse, concentrando i processi di generazione dei big data (ad esempio, il monitoraggio di situazioni del mondo reale tramite sistemi cyber-fisici) solo sulle fonti che si prevede siano le più promettenti in base al contesto attualmente applicabile (e spesso specifico dell'applicazione) [28]. In generale, i valori nascosti nei big data dipendono dalla freschezza dei dati. Pertanto, è necessario sviluppare un principio di importanza legato al valore analitico per decidere

quali dati devono essere scartati e quali conservati [1]. Per una soluzione di big data analytics, è necessario completare quanto segue [30]:

- Caricamento dei dati - È necessario sviluppare un software per caricare i dati da fonti multiple e diverse. Un sistema deve gestire la natura distribuita di Hadoop da un lato e la natura non distribuita dell'origine dei dati dall'altro. Il sistema deve gestire i record corrotti e deve fornire servizi di monitoraggio.
- Analisi dei dati - La maggior parte delle fonti di dati fornisce dati in un determinato formato che deve essere analizzato nel sistema Hadoop. Alcuni formati, come JSON, sono complicati da analizzare perché un record può essere composto da molte righe di testo e non da una sola riga per record.
- Analisi dei dati - Una soluzione di analisi dei big data deve supportare iterazioni rapide per poter analizzare correttamente i dati.

I tipi di metodi analitici dei Big Data includono generalmente [40]: 1) analitica descrittiva - che comporta la descrizione e la sintesi dei modelli di conoscenza; 2) analitica predittiva - previsione e modellazione statistica per determinare le possibilità future; e 3) analitica prescrittiva - che aiuta gli analisti nel processo decisionale determinando le azioni e valutandone l'impatto. Per i big data sono stati utilizzati sistemi distribuiti, database massive parallel processing (MPP), database non relazionali o in-memory. I database MPP offrono elevate prestazioni di interrogazione e scalabilità della piattaforma. I database non relazionali, come Not Only SQL (NoSQL), sono utilizzati per archiviare e gestire dati non strutturati o non relazionali e mirano a una scalabilità massiccia, alla flessibilità del modello di dati e alla semplificazione dello sviluppo e della distribuzione delle applicazioni. I database in-memory gestiscono i dati nella memoria del server, eliminando così l'input/output (I/O) su disco e consentendo risposte in tempo reale da parte del database. Inoltre, i database in-memory sono stati utilizzati per l'analisi avanzata dei Big Data, in particolare per velocizzare l'accesso e lo scoring dei modelli analitici per l'analisi [41]. Esistono alcuni strumenti per i Big Data come Hive, Splunk, Tableau, Talend, RapidMiner e MarkLogic. Hive semplifica la gestione e l'interrogazione di grandi insiemi di dati che risiedono in uno storage distribuito. Splunk si concentra sullo sfruttamento dei dati macchina creati da varie fonti, come sensori e siti web. Tableau è uno strumento di visualizzazione dei dati che consente agli utenti di creare grafici a barre, diagrammi e mappe. Talend è uno strumento open source per sviluppare, testare e distribuire prodotti per la gestione dei dati e l'integrazione delle applicazioni. RapidMiner offre alle aziende una soluzione centralizzata dotata di un'interfaccia grafica potente e robusta che consente agli utenti di creare,

mantenere e fornire analisi predittive. MarkLogic può essere utilizzato per gestire carichi di dati ingenti e consentire agli utenti di accedervi tramite aggiornamenti e avvisi in tempo reale [42].

Quando la velocità diventa molto elevata, gli strumenti per i big data saranno probabilmente l'unica opzione. Gli strumenti per i big data sono in grado di estrarre e analizzare i dati da enormi insiemi di dati in modo molto rapido, il che è particolarmente utile per i dati in rapida evoluzione che possono essere analizzati attraverso l'elaborazione in-memory. Gli strumenti per i Big Data sono in grado di distribuire lavori di elaborazione complessi a un gran numero di nodi, riducendo la complessità computazionale [31]. Hadoop su HDFS è l'architettura di base. Per gestire il volume e la quantità di dati vengono utilizzati Oozie e Elastic MapReduce (EMR) con Flume e Zookeeper, strumenti standard per la gestione dei Big Data [32]. MapReduce lavora con valori numerici e nominali. È in grado di elaborare un lavoro massiccio in un breve periodo di tempo; tuttavia, gli algoritmi devono essere riscritti ed è necessaria la comprensione dell'ingegneria dei sistemi [23]. Con YARN, Hadoop supporta ora diversi modelli di programmazione e output sia in tempo quasi reale che in batch [18]. Esistono molti strumenti software di qualità che consentono di sfruttare i vantaggi dei big data. Ad esempio, Kitenga Analytics Suite di Dell è una piattaforma di ricerca e analisi dei big data leader del settore, progettata per integrare informazioni di ogni tipo in visualizzazioni di facile utilizzo. Kitenga è abilitato a Hadoop per la scalabilità dei big data e consente l'integrazione di fonti di dati eterogenee e l'archiviazione a costi contenuti di volumi di dati crescenti. Kitenga può analizzare direttamente i risultati di Hadoop utilizzando strumenti di visualizzazione delle informazioni che si legano direttamente ai file di Hadoop Distributed File System (HDFS) e indicizzano i dati e i metadati creati in una forma ricercabile con funzionalità di visualizzazione incorporate [33]. Gli amministratori di sistema responsabili della manutenzione delle piattaforme di calcolo per i Big Data utilizzano spesso una delle tre strategie seguenti [34]:

- Cluster di calcolo interno. Per l'archiviazione a lungo termine di dati unici o sensibili, spesso ha senso creare e mantenere un cluster Apache Hadoop utilizzando una serie di server collegati in rete all'interno della rete interna di un'organizzazione.
- Cluster di calcolo esterno. Nel settore IT si sta diffondendo la tendenza a esternalizzare elementi dell'infrastruttura a fornitori di servizi di "utility computing". Alcune organizzazioni consentono agli amministratori di sistema di noleggiare cluster Apache Hadoop e sistemi di archiviazione dei dati precostituiti.

- Cluster di calcolo ibrido. Un'opzione ibrida comune è quella di fornire risorse di cluster di calcolo esterne utilizzando servizi per le attività di analisi dei Big Data on-demand e creare un cluster di computer interno modesto per l'archiviazione dei dati a lungo termine.

L'analisi dei big data comporta diverse fasi distinte, come mostrato nella Tabella 2 [35], ognuna delle quali introduce delle sfide. Molte sfide significative vanno oltre la fase di analisi. Ad esempio, i Big Data devono essere gestiti in un contesto che può essere rumoroso, eterogeneo e non includere un modello iniziale. Ciò comporta la necessità di tracciare la provenienza e di gestire errori e incertezze [35]. Oltre alle sfide tecniche generali dei big data, vi sono altre sfide [16]: 1) rendere i dati più accessibili strutturandoli attraverso l'aggiunta di metadati e consentendo l'integrazione di silos di dati separati; 2) risolvere le questioni normative relative alla proprietà dei dati e alla privacy dei dati; 3) sfruttare i vantaggi delle fonti di Open Data e Linked Data già disponibili. Le sfide dei Big Data comprendono non solo la scala, ma anche l'eterogeneità, la mancanza di integrazione delle strutture, la qualità dei dati, la privacy e la sicurezza, ecc. Il raggiungimento del grande potenziale dei Big Data richiede un approccio ponderato e olistico alla gestione, all'analisi e all'intelligence dei dati [43].

Tabella 2. La pipeline e le sfide dell'analisi dei Big Data

Aspetti	Descrizione
Le fasi principali dell'analisi dei Big Data	Acquisizione/Registrazione, Pulizia/Estrazione/Annotazione, Integrazione/Aggregazione/Rappresentazione, Analisi/Modellazione, Interpretazione
Sfide durante le fasi	Eterogeneità, tempestività, scala, privacy, collaborazione umana

Anche se una correlazione può rivelarsi solida in un determinato periodo, l'analisi dei Big Data da sola non è in grado di fornire indicazioni su ciò che potrebbe causare l'interruzione della correlazione, né su quale modello potrebbe emergere al suo posto. Una critica all'analisi dei Big Data è che l'esistenza di enormi insiemi di dati non elimina le tradizionali trappole statistiche, in particolare quelle dell'errore campionario e della parzialità del campione. Le affermazioni relative all'ubiquità delle reti di sensori e di altre fonti di Big Data sono spesso esagerate o ignorano fonti specifiche e consistenti di distorsione [13]. I requisiti futuri che la prossima generazione di tecnologie Big Data dovrebbe soddisfare [16]:

- Gestire la crescita di Internet - Con l'aumento degli utenti online, le tecnologie Big Data dovranno gestire volumi di dati maggiori.
- Elaborazione in tempo reale - L'elaborazione dei Big Data veniva inizialmente effettuata in batch di dati storici. Negli ultimi anni sono diventati disponibili sistemi di elaborazione dei flussi, come Apache Storm, che consentono nuove funzionalità applicative.
- Elaborare tipi di dati complessi - I dati come i grafi e altri tipi di strutture di dati più complicati devono essere facilmente elaborati dalle tecnologie Big Data.
- Indicizzazione efficiente - L'indicizzazione è fondamentale per la ricerca online dei dati ed è quindi essenziale nella gestione di grandi collezioni di documenti e dei relativi metadati.
- Orchestrazione dinamica dei servizi in contesti multi-server e cloud - La maggior parte delle piattaforme odierne non è adatta al cloud e mantenere la coerenza dei dati tra i diversi data store è una sfida.
- Elaborazione simultanea dei dati - La possibilità di elaborare grandi quantità di dati in modo simultaneo è molto utile per gestire grandi volumi di utenti contemporaneamente.

4.5 Data Mining e Machine Learning, Deep Learning e Big Data Analytics

Il rilevamento degli outlier è uno dei compiti del data mining. I metodi informatici per il rilevamento degli outlier possono essere classificati in quattro approcci: l'approccio statistico, l'approccio basato sulla densità degli outlier locali, l'approccio basato sulla distanza e l'approccio basato sulla deviazione [12]. Il Local Outlier Factor (LOF) è un algoritmo per l'identificazione di outlier locali basati sulla densità. La densità locale di un punto viene confrontata con quella dei suoi vicini attraverso il LOF. Se la prima è significativamente inferiore alla seconda (con un valore LOF maggiore di uno), il punto

si trova in una regione più scarsa rispetto ai suoi vicini, il che suggerisce che si tratta di un outlier. Un difetto del LOF è che funziona solo su dati numerici. Un altro modo per individuare gli outlier è il clustering. Le tecniche di clustering possono essere utilizzate per identificare cluster di uno o pochi record distanti dagli altri. Raggruppando i dati in cluster, i dati che non sono assegnati a nessun cluster vengono considerati outlier [22,36].

Tabella 3. Algoritmi di Deep Learning, Data Mining (DM) e Machine Learning (ML) [8]

Algoritmi e categorie DM/ML	Pro	Contro
Clustering basato sulla densità	gestire dati complessi e non statici, rilevare valori anomali e forme arbitrarie	lento, difficile selezione dei parametri, non adatto a grandi insiemi di dati
Partizione Clustering	Semplice, veloce, utile per gestire grandi insiemi di dati	elevata sensibilità all'inizializzazione, al rumore e agli outlier
Raggruppamento gerarchico	capacità di visualizzazione	lenti, poco precisi, scarsi in visualizzazione di dati di grandi dimensioni, che utilizzano un'enorme quantità di memoria
Macchina a vettori di supporto (SVM) per la classificazione	elevata precisione	formazione lenta, costosa dal punto di vista computazionale
Albero decisionale (DT) per la classificazione	semplice, facile da implementare	limitazione dello spazio, overfitting
Rete neurale per la classificazione	gestire dati rumorosi, individuare relazioni non lineari	lento, bassa precisione, modelli black-box, costoso dal punto di vista computazionale
Ensemble per la classificazione	predittivo, generalizzazione, alte prestazioni, superamento dell'overfitting	difficile da analizzare, costoso dal punto di vista computazionale
Apprendimento profondo per la classificazione	generalizzazione, apprendimento, apprendimento semi-supervisionato, apprendimento di caratteristiche non supervisionate, multi-task, architettura profonda, dataset di grandi dimensioni	difficile da interpretare, costoso dal punto di vista computazionale

Le variabili non numeriche o alfa presentano alcuni problemi diversi da quelli delle variabili numeriche. Alcuni strumenti, come gli alberi decisionali, possono gestire i valori alfa nella loro forma alfa. Altri strumenti, come le reti neurali, possono gestire solo una rappresentazione numerica del valore alfa [9]. A causa della diversità dei tipi di database, alcuni di essi possono contenere oggetti di dati complessi, tra cui dati temporali, dati spaziali, dati di transazioni, ipertesti o dati multimediali. Non è realistico aspettarsi che un unico sistema sia in grado di estrarre tutti i tipi di dati. Pertanto, si può pensare di avere diversi sistemi di data mining per diversi tipi di dati [12]. Esiste una direzione di ricerca chiamata privacy-preserving data mining che mira a conciliare la tensione tra big data e privacy [28]. È stato proposto un framework per affrontare il problema dell'apprendimento da dati eterogenei, chiamato Collective Data Mining (CDM) [5].

A causa di quantità di dati senza precedenti o della loro complessità, è spesso necessario un data mining ad alte prestazioni. Prestazioni elevate nel data mining significano sfruttare sistemi di gestione di database paralleli e CPU aggiuntive per ottenere vantaggi in termini di prestazioni. L'obiettivo principale del parallelismo è ottenere un miglioramento delle prestazioni. Esistono due misure principali di miglioramento delle prestazioni. La prima è il throughput, ovvero il numero di operazioni che possono essere completate in un determinato intervallo di tempo. La seconda è il tempo di risposta, ovvero il tempo necessario per completare un singolo task dal momento in cui viene inviato. Le due misure sono normalmente quantificate dalle seguenti metriche: scale up e speed up [20]. È stata presentata e descritta l'architettura di riferimento per i Big Data e gli Analytics, che fornisce: 1) un approccio alla gestione delle informazioni che unifica tutte le forme di dati, compresi quelli strutturati, semi-strutturati e non strutturati; 2) la capacità di gestire flussi di dati in batch e in tempo reale; 3) analisi in-database e in-memory ad alte prestazioni [37].

Gli algoritmi di apprendimento profondo utilizzano un'enorme quantità di dati non supervisionati per estrarre automaticamente rappresentazioni complesse. Le architetture di apprendimento profondo hanno la capacità di generalizzare in modo non locale e globale. Estraggono rappresentazioni direttamente da dati non supervisionati senza interferenze umane. Uno dei vantaggi principali dell'apprendimento profondo è l'analisi e l'apprendimento di enormi quantità di dati non supervisionati, che lo rendono uno strumento prezioso per l'analisi dei Big Data, dove i dati grezzi sono in gran parte non etichettati e non classificati [38]. L'apprendimento profondo e l'HPC che lavorano con i Big Data migliorano l'intelligenza di calcolo e il successo; l'apprendimento profondo e il calcolo eterogeneo (HC) che lavorano con i Big Data aumentano il successo [39]. L'analisi dei dati con gli algoritmi di deep learning è un processo di addestramento di modelli di rappresentazione

dei dati di alto livello basati su una parte del corpus di input disponibile, per poi utilizzare il corpus di input rimanente con i modelli appresi per estrarre le astrazioni e le rappresentazioni dei dati. Una domanda da esplorare è quale volume di dati di input sia generalmente necessario per addestrare rappresentazioni di dati utili o buone da parte degli algoritmi di deep learning. Questo può essere un lavoro futuro sull'apprendimento profondo nell'analisi dei Big Data [38]. La tabella 3 illustra i vantaggi e gli svantaggi del deep learning e degli algoritmi tradizionali di data mining e machine learning nelle applicazioni sanitarie. I metodi tradizionali di data mining e machine learning hanno difficoltà a gestire grandi volumi di dati, dati ad alta dimensione, dati non categorizzati e non supervisionati, ecc. Pertanto, presentano limitazioni nell'analisi dei Big Data.

È importante sviluppare approcci efficienti di pulizia dei big data per migliorare la qualità dei dati. La virtualizzazione dei dati e i laghi di dati sono approcci potenti che aiutano l'integrazione dei dati. PCA ed EFA sono spesso utilizzati per ridurre le dimensioni dei dati. L'eterogeneità dei big data implica anche la gestione simultanea di dati strutturati, semi-strutturati e non strutturati. Ogni fase dell'analisi dei Big Data presenta delle sfide. Queste includono l'elaborazione in tempo reale, la gestione di tipi di dati complessi e l'elaborazione simultanea dei dati, ecc.

I metodi tradizionali di data mining e machine learning presentano limitazioni nell'analisi dei Big Data. L'apprendimento profondo è in grado di analizzare e apprendere quantità massicce di dati non supervisionati; pertanto, ha un potenziale nell'analisi dei Big Data in cui i dati grezzi sono in gran parte non etichettati e non classificati. Le confluenze tra Big Data analytics, heterogeneous computing (HC), HPC e Deep learning possono essere un tema di ricerca per i Big Data eterogenei.

Riferimenti

[1] Chen M, Mao S, Liu Y. I grandi dati: Un'indagine. Reti e applicazioni mobili. 2014 Apr 1; 19(2): 171-209.

[2] Jirkovský V, Obitko M. Riduzione dell'eterogeneità semantica per i Big Data nell'automazione industriale. InITAT 2014.

[3] Singh VK, Gao M, Jain R. Situation recognition: an evolving problem for heterogeneous dynamic big multimedia data. InProceedings of the 20th ACM international conference on Multimedia 2012 Oct 29 (pp. 1209-1218). ACM.

[4] [4] Hai R, Geisler S, Quix C. Constance: Un sistema di data lake intelligente. In Proceedings of the 2016 International Conference on Management of Data 2016 Jun 26 (pp. 2097-2100). ACM.

[5] [5] Caragea D. Learning classifiers from distributed, semantically heterogeneous, autonomous data sources (tesi di dottorato, Iowa State University), 2004, 1-225.

[6] Anderson K, Arora A, Aoi S, Fujinuma K, et al. Big Data and Disaster Management, Technical Report No. GIT-CERCS-13-09; Georgia Institute of Technology, CERCS. A Report from the JST/NSF Joint Workshop, JST/NSF Joint Workshop Report on Big Data and Disaster Management, a cura di C. Pu e M. Kitsuregawa, maggio 2013.

[7] Wu, X., Zhu, X., Wu, G. Q., & Ding, W. (2014). Data mining con i big data. IEEE Transactions on Knowledge and Data Engineering, 26(1), 97-107.

[8] Fang R, Pouyanfar S, Yang Y, Chen SC, Iyengar SS. L'informatica sanitaria computazionale nell'era dei big data: un'indagine. ACM Computing Surveys (CSUR). 2016 Jun 14; 49(1): 12.

[9] [9] Pyle D. Preparazione dei dati per il data mining. Morgan Kaufmann, 1999.

[10] Kabacoff R. R in action: data analysis and graphics with R. Manning Publications Co.; 2015 Mar 3.

[11] Torgo L, Torgo L. Data mining with R: learning with case studies. Boca Raton, FL: Chapman & Hall/CRC; 2011.

[12] Han J, Pei J, Kamber M. Data mining: concepts and techniques. Elsevier; 2011 Jun 9.

[13] Dati B. Trasporti: Comprendere e valutare le opzioni. Rapporto del Consiglio di partenariato aziendale, Organizzazione per la cooperazione e lo sviluppo economico (OCSE)/Forum internazionale dei trasporti, maggio 2015, 1-66.

[14] Viña A. Data Virtualization Goes Mainstream, White Paper, Denodo Technologies, Inc, USA, 2015, 1-18.

[15] Rudin C., Dunson D., Irizarry R., Ji H., Laber E., Leek J., & Wasserman L. Discovery with data: Sfruttare la statistica e l'informatica per trasformare la scienza e la società. 2 luglio 2014, 1-27.

[16] Curry E, Kikiras P, Freitas A. et al. Big Data Technical Working Groups, White Paper, BIG Consortium, 2012.

[17] Pullokkaran LJ. Analisi della virtualizzazione dei dati e della standardizzazione dei dati aziendali nella business intelligence (tesi di dottorato, Massachusetts Institute of Technology), 2013.

[18] Stein B, Morrison A. Il lago di dati aziendali: Migliore integrazione e analisi più approfondite. Previsioni tecnologiche di PwC: Ripensare l'integrazione. 2014(1), 1-9.

[19] Shalev-Shwartz S, Ben-David S. Capire l'apprendimento automatico: Dalla teoria agli algoritmi. Cambridge University Press; 2014 maggio 19.

[20] Abbass HA, editore. Estrazione dei dati: A Heuristic Approach: Un approccio euristico. IGI Global; 2001 Jul 1.

[21] Wikibook, Algoritmi di data mining in R - Wikibooks, libri aperti per un mondo aperto. PDF generato utilizzando il toolkit open source mwlib. Vedere http://code.pediapress.com/, 2014 14 Jul.

[22] [22] Galit S, Nitin P, Peter B. Data Mining in Excel: Lecture Notes and Cases. Resampling Stats, Inc., USA, 2005, 30 dicembre.

[23] [23] Harrington P. Machine learning in action. Greenwich, CT: Manning; 2012 Apr 16.

[24] Hertzmann A, Fleet D. Machine Learning and Data Mining Lecture Notes. Dipartimento di Informatica, Università di Toronto. 2010.

[25] Zhang J, Yang X, Appelbaum D. Verso un'efficace analisi dei Big Data nella revisione contabile continua. Orizzonti contabili. 2015 Jun; 29(2):469-76.

[26] Tak PA, Gumaste SV, Kahate SA, The Challenging View of Big Data Mining, International Journal of Advanced Research in Computer Science and Software Engineering, 5(5), May 2015, 1178-1181.

[27] Yenkar V, Bartere M. Recensione su "Data Mining con Big Data". International Journal of Computer Science and Mobile Computing, 3(4), aprile 2014, 97-102.

[28] NESSI, Big Data: A New World of Opportunities, NESSI White Paper, Networked Software and Services Initiative (NESSI), dicembre 2012, 1-25.

[29] Stanford Law School, Big Data and Privacy: Making Ends Meet, The Future of Privacy Forum (FPF), 10 settembre 2013, 1-122.

[30] Datameer, Inc. The Guide to Big Data Analytics, White Paper, 2013, 1-39.

[31] Schotman R, Mitwalli A. Big Data per il marketing: Quando i Big Data sono la scelta giusta? Canopy - The Open Cloud Company, 2013, p8.

[32] Jaseena KU, David JM. Problemi, sfide e soluzioni: big data mining. NeTCoM, CSIT, GRAPH-HOC, SPTM-2014. 2014:
131-40.

[33] Daniel D. Gutierrez, InsideBIGDATA Guide to Big Data for Finance, White Paper, DELL e intel, Whitepaper, 2015, 1-14.

[34] [34] Kreuter F, Berg M, Biemer P, Decker P, Lampe C, Lane J, O'Neil C, Usher A. AAPOR Report on Big Data. Mathematica Policy Research; 2015 Feb 12.

[35] Labrinidis A, Jagadish HV. Sfide e opportunità dei big data. Atti del VLDB Endowment. 2012 Aug 1; 5(12): 2032-2033.

[36] Zhao Y. R e data mining: Esempi e casi di studio. Academic Press; 2012 Dec 31.

[37] Chappelle D. Big Data & Analytics Reference Architecture, Oracle White Paper, Oracle Enterprise Transformation Solutions Series, settembre 2013, 1-39.

[38] Najafabadi MM, Villanustre F, Khoshgoftaar TM, Seliya N, Wald R, Muharemagic E. Deep learning applications and challenges in big data analytics. Journal of Big Data. 2015 Feb 24; 2(1): 1. [39] Wu R, Deep learning meets heterogeneous computing, Workshop, Baidu Inc., 2014.

[40] Sivarajah U, Kamal MM, et al. Critical Big Data Analysis Challenges and Analytical Methods, Journal of Business Research, 70, (2017, 263-286.

[41] Elgendy N e Elragal A, Big Data Analytics: A Literature Review Paper. P. Perner (a cura di): ICDM 2014, LNAI 8557, Springer International Publishing Switzerland, 2014, 214-227.

[42] [42] Yusuf Perwej, An Experiential Study of the Big Data, International Transaction of Electrical and Computer Engineers System, 2017, Vol. 4, No. 1, 14-25.

[43] Almeida FL, Calistru C. Le principali sfide e problematiche della gestione dei big data. International Journal of Research Studies in Computing. 2013 Oct 9; 2(1).

[44] Lidong Wang, “Heterogeneous Data and Big Data Analytics.” Automatic Control and Information Sciences, vol. 3, no. 1 (2017): 8-15. doi: 10.12691/acis-3-1-3.

5 ATTIVITÀ Modello di base per le informazioni raccolte dai sensori e Sistemi di Monitoraggio e early warning.

5.1 Sistema a supporto della rilevazione di condizioni di errore

Nei moderni sistemi di supporto alle decisioni è necessario migliorare le prestazioni in termini di rilevamento, affidabilità e capacità in tempo reale. Queste caratteristiche sono solitamente inversamente proporzionali. In questo lavoro proponiamo un approccio innovativo per il rilevamento intelligente degli eventi e la comprensione arricchita dei fenomeni. In particolare, l'approccio proposto si basa su un processo in due fasi che cerca di identificare rapidamente una condizione di allarme e poi di elaborare la base di conoscenza acquisita con un post reasoner per affinare la decisione finale e dare agli operatori maggiori sensazioni sulla valutazione della situazione e sugli allarmi generati.

La pressante necessità di proteggere il territorio e di mettere in campo adeguate strategie di prevenzione dei disastri ha portato le agenzie di protezione, così come la comunità scientifica internazionale, a uno sforzo volto alla definizione di nuove strategie e strumenti di homeland security. Un'attività centrale in ogni sistema di Homeland Security è il monitoraggio e l'osservazione di diversi fenomeni, finalizzati a fornire una descrizione aggiornata e significativa dello scenario monitorato, nonché delle sue possibili evoluzioni, per consentire adeguate contromisure per la protezione e la sicurezza di persone e cose. In questi scenari, non sono necessari solo sistemi di sorveglianza e di allarme intelligenti, ma sono auspicabili sistemi di supporto alle decisioni (DSS) arricchiti. Tali sistemi si basano su strumenti eterogenei di acquisizione dei dati (sensori, video, dati storici e simulati, ...), sull'elaborazione dei dati per eliminare le informazioni non significative e sull'interpretazione del significato di alto livello di ciò che i dati rappresentano, con l'obiettivo di ridurre i falsi positivi e di rilevare le condizioni di allarme deboli. La disponibilità di tecniche di monitoraggio avanzate e di fonti informative eterogenee ha aumentato l'accuratezza nell'osservare, misurare e descrivere la natura dei fenomeni: l'attuale livello tecnologico in questo campo rappresenta un'opportunità per migliorare la comprensione dei fenomeni osservati ma, allo stesso tempo, introduce un elevato grado di complessità nell'elaborazione e nella fusione dei dati.

I sistemi intelligenti di supporto alle decisioni sono necessari per consentire, quando possibile, l'adozione automatica di contromisure in caso di allarme o per supportare gli utenti finali durante le attività decisionali (quando un numero troppo elevato di sensori, dispositivi o telecamere posizionati

all'interno del sito da proteggere produce un'ampia quantità di dati da elaborare). Tuttavia, molti sistemi di rilevamento automatici e intelligenti generano avvisi non necessari (falsi allarmi); questo problema, purtroppo, limita fortemente l'utilizzo di questi sistemi per attivare contromisure automatiche o parzialmente automatiche, tanto che negli ultimi anni l'attenzione del mondo scientifico è rivolta sia alla gestione delle informazioni con approcci di information e decision fusion, sia alla stima quantitativa dell'affidabilità di questi sistemi.

D'altra parte, per migliorare la valutazione della situazione, è possibile adottare diversi tipi di modelli per la descrizione della base di conoscenza, per la correlazione degli eventi e per la definizione della situazione e l'identificazione delle minacce. Un modello molto promettente è quello basato su modelli semantici e ontologici.

Il modello semantico può essere utilizzato per la comprensione dei fenomeni osservati. In particolare, tutti i sensori devono condividere lo stesso modello di dati e la stessa interpretazione dei dati. Il modello di dati deve fornire meccanismi di interoperabilità sintattica e procedure di arricchimento semantico per costruire modelli al fine di (i) garantire un'interpretazione corretta e condivisa delle informazioni, (ii) aggregare i dati grezzi in eventi (semplici e composti), che saranno poi utilizzati per la valutazione della situazione prima di una decisione finale.

In letteratura sono disponibili alcuni approcci per il rilevamento di eventi e il supporto alle decisioni basati su regole di inferenza semantica per la comprensione dei fenomeni. Tuttavia, a causa dell'overhead introdotto, la base di conoscenza viene solo inferita in modalità offline.

In questo lavoro proponiamo un approccio innovativo per il rilevamento intelligente degli eventi e la comprensione arricchita dei fenomeni: la base di conoscenza sarà inferita in tempo reale, per il rilevamento degli eventi, e un classificatore intelligente leggero emetterà un allarme. L'approccio proposto si basa su due fasi:

1. Un motore di inferenza on-line intelligente e leggero per lanciare un allarme in caso di rilevamento di un evento di minaccia;

2. Un motore di inferenza off-line post reasoner, per comprendere l'evento e le sue cause.

Il primo ha il compito di rilevare, con vincoli di tempo reale, condizioni di pericolo, dando un pre-allarme; il secondo svolge un'attività di ragionamento più complessa per aiutare gli utenti a comprendere la situazione di pericolo e affinare la decisione.

Il promemoria dell'articolo è strutturato come segue: nella Sezione 2 sono riportati alcuni lavori correlati, nelle Sezioni 3 e 4 sono presentati il modello e la relativa architettura del sistema di monitoraggio proposto. Nella Sezione 5 viene presentato un semplice caso di studio sul classificatore intelligente e, infine, nella Sezione 6 vengono discusse alcune conclusioni.

In letteratura sono disponibili alcuni approcci semantici per gestire dati eterogenei provenienti da sensori e inferirli per il rilevamento di eventi. Questi approcci sfruttano l'inferenza offline per estrarre la conoscenza implicita dai dati dei sensori.

D'altra parte, alcuni approcci stanno iniziando a utilizzare tecniche in linea sia per arricchire il modello semantico dei dati sia per gestire la rilevazione degli eventi in tempo reale; molto spesso, viene associata un'inferenza offline per la comprensione degli eventi e l'analisi dei fenomeni.

In (Huang e Javed, 2008) viene proposta un'architettura per la descrizione e l'elaborazione delle informazioni dei sensori, denominata SWASN (Semantic Web Architecture for Sensor Network). L'architettura si basa su quattro livelli: il primo è il livello fisico, composto da diverse reti di sensori. Ogni rete di sensori gestisce un proprio formato di dati. I dati vengono elaborati in un livello ontologico, in cui ogni rete ha un'ontologia locale. L'unione di queste ultime fornisce un'ontologia globale. L'ontologia globale è costruita su un vocabolario comune e viene elaborata nel livello semantico per l'estrazione della conoscenza, attraverso l'inferenza e il ragionamento semantico. Infine, a livello di utente, è possibile interrogare l'ontologia per elaborare i dati.

Architetture simili sono presentate in (Gomez e Laube 2009; Konstantinou, Solidakis, Zoi, Zafeiropoulos, Stathopoulos e Mitrou 2007; Konstantinou, Solidakis, Zoi, Zafeiropoulos, Stathopoulos e Mitrou 2010). In particolare, in (Gomez e Laube 2009) e (Solidakis, Zoi, Zafeiropoulos, Stathopoulos e Mitrou 2007) viene proposto un processo automatico di trasformazione di dati XML in RDF, guidato da regole di mappatura e ragionamento semantico. La trasformazione avviene in tempo reale, ma non viene proposto alcun sistema di rilevamento. In (Konstantinou, Solidakis, Zoi, Zafeiropoulos, Stathopoulos e Mitrou 2010) viene presentata un'architettura middleware per gestire il rilevamento di eventi in tempo reale. Si tratta di un'architettura middleware per l'annotazione automatica, in tempo reale e non supervisionata di caratteristiche di contesto di basso livello e la corrispondente mappatura a semantiche di alto livello. Consente la composizione di semplici regole attraverso interfacce specifiche, che possono avviare un sistema consapevole del contesto che annoterà i contenuti senza la necessità di competenze tecniche da parte dell'utente. Il middleware ha un modello semantico solo per la gestione degli eventi. Non esistono modelli per i dati acquisiti dai sensori.

5.2 Modello di monitoraggio del sistema

Un sistema di monitoraggio può essere composto da due livelli principali: la rete di sensori e il sistema di monitoraggio. Come illustrato nella figura che segue , la rete di sensori può essere caratterizzata dalle proprietà fisiche dei sensori, Tipologia di misura e Topologia.

D'altra parte il sistema di monitoraggio, basato su motori di inferenza, può essere caratterizzato da Conoscenza acquisita in tempo reale, Conoscenza inferita in tempo reale e Conoscenza acquisita a valle del processamento dei dati. Ogni nodo sensore è responsabile della misurazione di parametri specifici.

Il livello “Sensor physical features” modella le caratteristiche fisiche di un singolo nodo sensore e dell'intera rete di sensori.

Il livello “Measurement Typology” definisce i tipi di misure raccolte dai sensori.

il livello “Topology” modella le informazioni sulla distribuzione del sistema, descrivendo la posizione dei sensori nell'area di interesse.

Figura 11 Livelli di Elaborazione

Il sistema di monitoraggio agisce a due livelli diversi: una reazione in tempo reale e un'attività off-line (post reasoner). La prima mira a fornire allarmi adeguati quando si verificano eventi pericolosi, la seconda a fornire un quadro completo e dettagliato della situazione che possa essere utile agli operatori sia per la comprensione della situazione che per il supporto alle decisioni.

Il sistema di monitoraggio funziona quindi in queste due fasi, attraverso un classificatore veloce e un ragionatore a posteriori. È modellato da tre livelli.

Il livello “Real Time Acquired Knowledge” ha il compito di modellare la tipologia, la struttura e i valori dei dati grezzi e strutturati acquisiti e trasmessi dai sensori, arricchiti con informazioni semantiche su di essi (Amato, C 2011). A questo livello i dati vengono formalizzati ed elaborati sia dal classificatore veloce che dal post ragionatore.

Il linguaggio di codifica, utilizzato per elaborare e trasmettere le informazioni, è lo standard RDF.

Il livello “Real Time inferred Knowledge” modella la conoscenza derivata dall'applicazione, sui dati rilevati. Questo livello lavora già su dati arricchiti semanticamente, l'overhead per collegare i dati con le informazioni su di essi, infatti, è necessario a questo livello perché esiste una moltitudine di eventi che possono essere rilevati solo combinando informazioni provenienti da collezioni di sensori che sono eterogenei sia per la tipologia di misura effettuata che per il formato dei dati in cui vengono inviati ai nodi centralizzatori. A questo livello, molti dettagli sulla situazione attuale vengono astratti, per consentire al classificatore di eseguire compiti decisionali efficienti, anche se non è in grado di ricavare la conoscenza completa dell'ambiente monitorato. Le informazioni estratte vengono poi riconsiderate nel modello astratto di Post Reasoner Knowledge, che funziona senza vincoli sul tempo.

Il livello “Post Reasoner Knowledge” mira a modellare tutti gli aspetti rilevanti dell'ambiente monitorato. Il suo scopo è quello di ricavare conoscenze utili per avere una visione dettagliata della situazione, finalizzata a dare informazioni su ciò che sta accadendo e a dare supporto nel processo decisionale.

I dati acquisiti vengono arricchiti con informazioni semantiche RDF ed elaborati da un reasoner basato su Pellet (Kaplanski 2012).

Il ragionatore si basa su un componente con regole generali e su un componente specializzato che implementa le regole calibrate sull'ambiente da monitorare.

La conoscenza del dominio rilevante viene codificata con l'aiuto di esperti del dominio utilizzando strutture dati appropriate, le ontologie, che modellano gli elementi di interesse in termini di concetti e relazioni relative ai fenomeni da monitorare, agli eventi e alle azioni associate da eseguire. Questa ontologia viene utilizzata per collegare ogni elemento prodotto dal ragionatore con una descrizione adeguata e con informazioni appropriate che possono essere sfruttate per aiutare gli utenti a comprendere la situazione. Il sistema che implementa il ragionatore semantico è molto più costoso dal punto di vista computazionale rispetto ai classificatori utilizzati per i decisori in tempo reale. A questo livello, infatti, i dati inferiti in uscita sono progettati per dare supporto agli utenti con funzionalità di ragionamento offline e di data mining, che possono essere sfruttate per ottenere una conoscenza completa delle situazioni, anche in un momento successivo.

5.3 Architettura di riferimento

L'architettura proposta combina diversi approcci di trattamento dati. Come illustrato in figura 12, l'architettura è composta da

- Un classificatore intelligente di eventi (che implementa i livelli di conoscenza acquisita e inferita in tempo reale);
- Un Post-Reasoner (che implementa il livello di conoscenza).

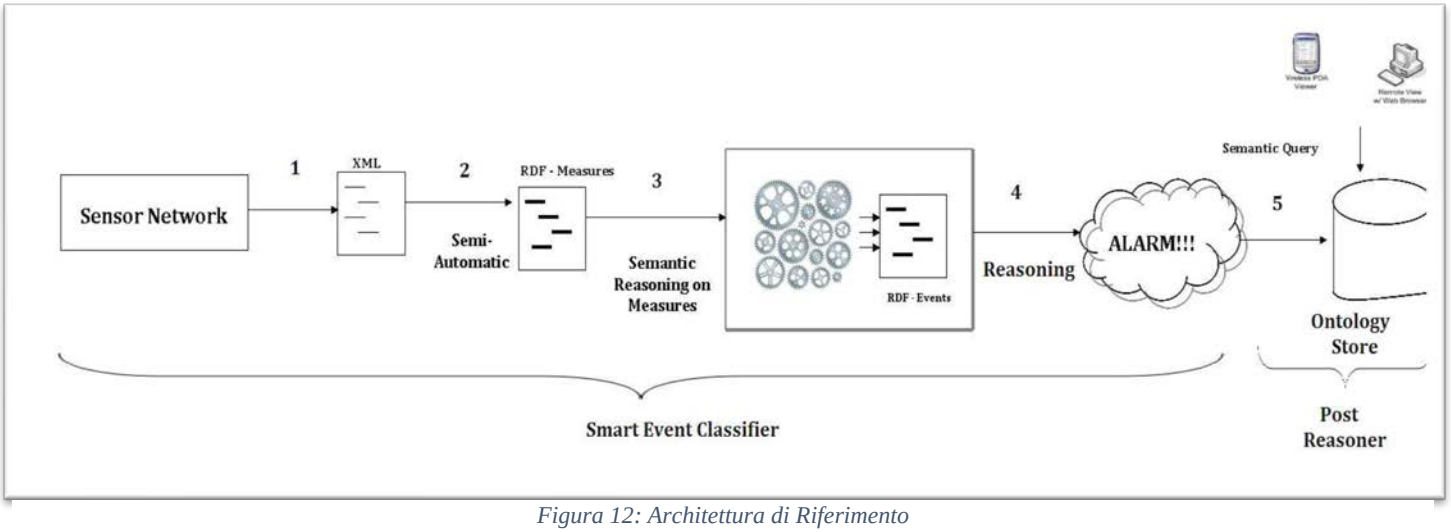


Figura 12: Architettura di Riferimento

Un classificatore può rilevare potenziali condizioni di pericolo e quindi, se necessario, lanciare un allarme. Il rilevamento degli eventi viene effettuato mediante la correlazione di dati provenienti da diversi sensori. Infatti, in situazioni reali, il potenziale pericolo non può essere rilevato utilizzando i dati provenienti da un singolo dispositivo.

Il classificatore ha una struttura standard, composta dai componenti learner e predictor; il primo costruisce il modello predittivo e il secondo sfrutta questo modello per il rilevamento degli eventi. Il predictor ha il compito di classificare i dati raccolti dalla rete di sensori per decidere se si sono verificate condizioni di allarme.

Abbiamo considerato un classificatore basato su regole, implementato come albero decisionale. Come di consueto nel meccanismo dell'albero decisionale, l'insieme delle regole decisionali è modellato come un albero in cui le foglie rappresentano le classi associate agli eventi da rilevare e i rami rappresentano le congiunzioni di caratteristiche, cioè le condizioni sui dati rilevati, che portano a quelle classi di eventi.

Per definire le regole del ramo, gli esperti di dominio classificano manualmente un insieme di dati di eventi (insieme di addestramento). Questi dati vengono utilizzati dal modulo di apprendimento per impostare i parametri del predictor, che regolano il rilevamento automatico delle condizioni di allarme. Per aumentare le prestazioni del sistema, le regole possono essere tagliate ricorrendo a un affinamento manuale effettuato dagli esperti di dominio (Liu, MA, Young 2000).

Nella figura 3 riportiamo un piccolo esempio di regola codificata come ramo di un albero. La regola codificata è:

(S1.location = 41°53'24″ N, 12° 29' 32″ E, S1.Pressure > 101.325 kPa,

S2.Pressione > 30 inHg,

S2.location= 41° 53' 37″N,12°29'11″ E)

==>Allarme

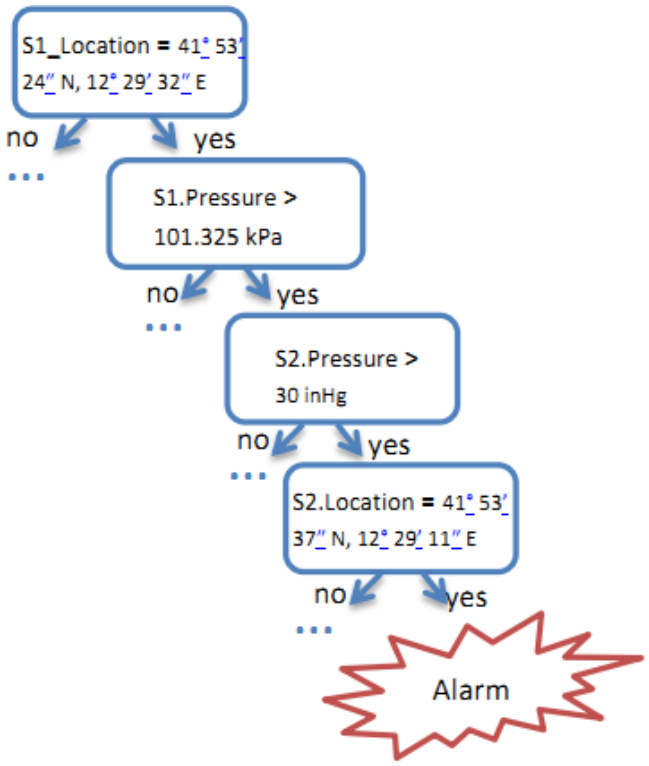


Figura 13: Ramo dell'albero codificato

Il predictor è realizzato come un sistema parametrico, i cui parametri possono variare, o in presenza di nuovi dati che influenzano l'uscita del sistema, o per intervento dell'operatore che può decidere di modificarli manualmente al verificarsi di nuove condizioni.

Per soddisfare i vincoli di tempo reale, consentendo al sistema di reagire in tempo utile, il classificatore può essere sintetizzato in hardware; in particolare, per gestire le variazioni sui parametri, il predittore è sintetizzato su un dispositivo riconfigurabile (FPGA) che consente la riconfigurazione del sistema quando è necessario (Wittig e Chow 1996). Dal punto di vista dell'arricchimento semantico, il classificatore intelligente di eventi svolge le seguenti azioni:

1. Le reti di sensori raccolgono dati e li formattano in file XML (Konstantinou, 2010), che codificano sia le proprietà del sensore sia i valori misurati;

2. Con un'attività semiautomatica, i file XML vengono arricchiti semanticamente e trasformati in file RDF. Questo file è conforme all'ontologia del dominio ed è adatto per eseguire ragionamenti semantici;

3. Con il linguaggio SWRL (Horrocks 2004) è possibile eseguire diverse inferenze per comporre eventi semplici in eventi composti.

4. Su eventi composti è possibile rilevare eventi di minaccia e generare allarmi.

Queste azioni vengono eseguite in tempo reale e alimentano la base di conoscenza per il componente Post Reasoner. L'ontologia appena costruita viene memorizzata in un repository (Triple Stores) (Broekstra, Kampman Van Harmelen 2002) e può essere utilizzata off-line attraverso l'adozione di linguaggi di interrogazione semantica come SPARQL (Prud'hommeaux e Seaborne 2004). Attraverso le query, il post reasoner è in grado di comprendere e spiegare agli utenti finali il significato degli allarmi e le loro cause.

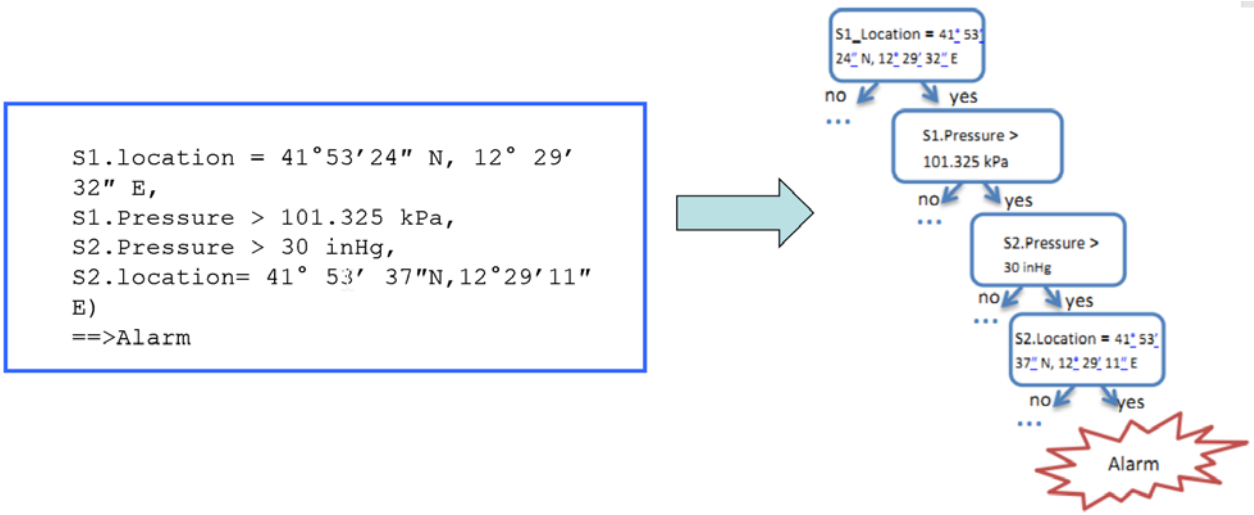


Figura 13: Mapping Regola - Ramo dell'albero decisionale

La base di conoscenza può essere vista come un archivio di informazioni su un particolare dominio di interesse (ad esempio, una stazione della metropolitana). Le tipiche basi di conoscenza sono costituite da concetti, proprietà e istanze. Abbiamo codificato la base di conoscenza utilizzando l'ontologia. L'ontologia è un insieme di classi, proprietà e istanze. Le classi definiscono i concetti del dominio; le proprietà definiscono la relazione tra le classi (dominio e intervallo). Le proprietà possono essere tra due classi o una proprietà di una classe (attributo).

Durante il ragionamento, vengono fatte delle inferenze, classificando le istanze dell'ontologia e associando nuove proprietà alle istanze, mantenendo la coerenza logica.

Il reasoner, basato su Pellet (Kaplanski 2010), è in grado di inferire conseguenze logiche da un insieme di fatti asseriti sul sistema di monitoraggio definiti dagli esperti dell'utente. In particolare, è composto da due componenti, uno che implementa le regole di inferenza generali e l'altro le regole

specialistiche, definite dagli esperti di dominio per catturare la conoscenza rilevante sull'ambiente da monitorare.

Il sistema utilizza la logica dei predicati del primo ordine per eseguire il ragionamento. Le inferenze procedono sia per concatenamento in avanti sia per concatenamento all'indietro (Kaplanski 2010). Non avendo vincoli di tempo reale, il post reasoner non è necessariamente implementato su un sistema embedded.

Nel listato seguente è riportato il listato XML con un esempio di output di sensore, in cui si rileva la possibile presenza di un pericolo chimico nell'ambiente.

```
<?xml version='1.0' encoding='UTF-8'?>
  <result>
   <nodeid value='1'/>
   <location value='Station1'>
   <name value='Chemical_Presence'/>
   <data value='true'/>
<timestampvalue='2012-05-13T09:00:03+01:00'/>
            </result>
```

Mentre nel listato seguente è riportato il corrispondente codice RDF che lo codifica in formato standard.

```
<!-- http://www.owl-ontologies.com/Ontology1.owl#CBRN1 -->

 <owl:NamedIndividual rdf:about="http://www.owl-ontologies.com/Ontology1.owl#CBRN1">
  <rdf:type rdf:resource="http://www.owl-ontologies.com/Ontology1.owl#CBRNe"/>
  <hasType rdf:resource="http://www.owl-ontologies.com/Ontology1.owl#Chemical"/>
  <hasLocation rdf:resource="http://www.owl-ontologies.com/Ontology1.owl#Station1"/>
 </owl:NamedIndividual>

<!--http://www.owl ontologies.com/Ontology1.owl#Chemical -->

<owl:NamedIndividual rdf:about="http://www.owl-ontologies.com/Ontology1.owl#Chemical">
  <rdf:type rdf:resource="http://www.owl-ontologies.com/Ontology1.owl#Sensor_Type"/>
```

```
</owl:NamedIndividual>


<!-- http://www.owl ontologies.com/Ontology1.owl#Chem2 -->
 <owl:NamedIndividual rdf:about="http://www.owl-ontologies.com/Ontology1.owl#Chem2">
  <rdf:type rdf:resource="http://www.owl-ontologies.com/Ontology1.owl#Chemical_presence"/>
  <atTimeDate rdf:datatype="&xsd;dateTimeStamp">2012-05-13T09:00:03+01:00</atTimeDate>
  <hasChem rdf:datatype="&xsd;boolean">true</hasChem>
 </owl:NamedIndividual>
```

Nel listato seguente è invece riportato un evento composto.

```
<!--http://www.owl-ontologies.com/Ontology1.owl#Detect_Explosive -->
 <owl:Thing rdf:about="&Ontology1;Detect_Explosive">
  <rdf:type rdf:resource="&Ontology1;Detect_Event"/>
  <rdf:type rdf:resource="&Ontology1;Explosive_Detection"/>
  <rdf:type rdf:resource="&Ontology1;Measurement"/>
  <rdf:type rdf:resource="&Ontology1;Simple_Event"/>
  <rdf:type rdf:resource="&owl;NamedIndividual"/>
  <Ontology1:is_Explosive_Detection rdf:datatype="&xsd;boolean">true </Ontology1:  is_Explosive_Detection>
  <Ontology1:Detect_by rdf:resource="&Ontology1;CBRN1"/>
  <Ontology1:MeasureFrom rdf:resource="&Ontology1;Chem2"/>
 </owl:Thing>
<!--http://www.owl-ontologies.com/Ontology1.owl#Detect_Possible_Explosive -->
   <owl:Thing rdf:about="&Ontology1;Detect_Possible_Explosive">
      <rdf:type rdf:resource="&Ontology1;Composed_Event"/>
      <rdf:type rdf:resource="&Ontology1;Detect_Event"/>
      <rdf:type rdf:resource="&Ontology1;Possible_Explosive"/>
      <rdf:type rdf:resource="&Ontology1;Simple_Event"/>
      <rdf:type rdf:resource="&owl;NamedIndividual"/>
      <Ontology1:Detect_Time rdf:datatype="&xsd;dateTime">2012-05-13T09:00:07+01:00 </Ontology1:Detect_Time>
      <Ontology1:is_Possible_Explosive rdf:datatype="&xsd;boolean">true</Ontology1:is_Possible_Explosive>
      <Ontology1:Composed_From rdf:resource="&Ontology1;Detect_Explosive"/>
      <Ontology1:Composed_From rdf:resource="&Ontology1;Detect_Intrusion"/>
```

```
</owl:Thing>
```

La condizione di allarme è codificata di seguito.

```
<!-- http://www.owl-ontologies.com/Ontology1.owl#Allarme -->

<owl:Thing rdf:about="&Ontology1;Allarme">
  <rdf:type rdf:resource="&Ontology1;Alarm"/>
  <rdf:type rdf:resource="&owl;NamedIndividual"/>
  <Ontology1:message rdf:datatype="&xsd;string"></Ontology1:message>
  <Ontology1:message rdf:datatype="&xsd;string">Attention Explosive Presence </Ontology1:message>
     <Ontology1335263048:Alarmfrom rdf:resource="&Ontology1;Detect_Drop_Explosive"/>
  </owl:Thing>
```

In questa attività abbiamo definito un sistema di monitoraggio intelligente basato su un processo in due fasi che cerca di identificare rapidamente un allarme e poi di elaborare la base di conoscenza acquisita con un post reasoner per affinare la decisione finale e dare agli operatori maggiori sensazioni sulla valutazione della situazione e sugli allarmi sollevati. L'approccio proposto combina i risultati significativi disponibili in letteratura per arricchire i modelli di dati con informazioni semantiche e, allo stesso tempo, utilizza un classificatore intelligente per rendere più rapido il processo di rilevamento. Abbiamo illustrato un semplice esempio, incentrato sul processo di arricchimento per costruire la base di conoscenza su cui un post ragionatore può dedurre ulteriori informazioni per la valutazione della situazione. Nei prossimi lavori intendiamo completare l'implementazione dell'architettura anche con questo componente e utilizzare questo approccio per arricchire i sistemi di supporto alle decisioni disponibili che si basano su modelli di rilevamento diversi, come quelli statistici o matematici.

5.4 Riferimenti

Konstantinou, E. Solidakis, A. Zafeiropoulos, P. Stathopoulos, N. Mitrou, 2010. Un middleware consapevole del contesto per l'arricchimento semantico in tempo reale di metadati multimediali distribuiti. *International Journal of Multimedia Tools and Applications* (MTAP), Springer, special issue on Data Semantics for Multimedia Systems, 46(2): 425-461.

Kaplanski, P. , 2010. Generatore di applicazioni centrate sui dati basato sulla logica di descrizione. *Atti della 2a Conferenza internazionale sulla tecnologia dell'informazione* (ICIT),. P 53-56. 2010. IEEE Publishing.

Liu, B., Ma, Y., Wong, C., 2000. *Miglioramento di un classificatore basato su regole di associazione. Journal of Principles of Data Mining and Knowledge Discovery.* P.P. 293-317.... Springer Verlag.

Wittig, R.D., Chow, P. , 1996. *OneChip: Un processore FPGA con logica riconfigurabile.* Atti della conferenza IEEE Symposium on FPGAs for Custom Computing Machines P. 126-135. 1996 IEEE Publishing

N. Konstantinou, E. Solidakis, S. Zoi, A. Zafeiropoulos, P. Stathopoulos, N. Mitrou, 2007. Priamos: A Middleware Architecture for Real Time Semantic Annotation of Context Features. *Terza conferenza internazionale IET sugli ambienti intelligenti.*

Gomez, L. e Laube, A. 2009. Middleware ontologico per l'elaborazione dinamica dei dati dei sensori wireless. In *Proceedings of the 2009 Third International Conference on Sensor Technologies and Applications* IEEE Computer Society, Washington, DC, USA, pp. 145-151.

Amato, F., Casola, V., Esposito, M., Mazzocca, N., & Mazzeo, A. A Smart Monitoring System Based On A Fast Classifier And A Semantic Post Reasoner.

Amato, F., Casola, V., Gaglione, A., Mazzeo, A. 2011. *Un modello di dati arricchito semanticamente per l'interoperabilità delle reti di sensori*. Journal of Simulation Modelling Practice and Theory. V 19. N 8. P 1745-1757.. Elsevier

V. Huang e M. Javed, *2008. Descrizione ed elaborazione semantica delle informazioni dei sensori.* In 2nd Conferenza internazionale sulle tecnologie e le applicazioni dei sensori....

Broekstra, J., Kampman, A., van Harmelen, F. 2002. *Sesamo: un'architettura generica per la memorizzazione e l'interrogazione di RDF e RDF Schema*. In ISWC. (luglio)

Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., & Dean, M. (maggio 2004). *SWRL: Un linguaggio di regole semantiche per il web che combinaOWL e RuleML*.

V. Casola, A. De Benedictis, A. Mazzeo e N. Mazzocca, 2011. *SeNsIM-SEC: sicurezza nelle reti di sensori eterogenee*, maggio, SARSSI2011.

Prud'hommeaux, E., Seaborne, A., 2004. Linguaggio di interrogazione SPARQL per RDF.

6 Algoritmi per il condizionamento distribuito del segnale

6.1 Descrizione attività

L'obiettivo è studiare, sviluppare e testare modelli di intelligenza artificiale per la compensazione del bias e della deriva delle piattaforme inerziali MEMS. I sensori MEMS sono l'integrazione di componenti elettrici e meccanici su scala nanometrica e microscopica. Questo tipo di sensori è stato identificato come una delle tecnologie più promettenti per il 21° secolo e ha il potenziale per rivoluzionare i prodotti industriali e di consumo, combinando la microelettronica basata sul silicio con la tecnologia di microlavorazione, insieme a tecniche e dispositivi basati su microsistemi.

Un sistema di navigazione inerziale (INS) comprende una piattaforma mobile contenente sensori inerziali come accelerometri, giroscopi o altri dispositivi di rilevamento del movimento. L'INS è utilizzato in un'ampia gamma di applicazioni, tra cui la navigazione di aerei, veicoli spaziali, sottomarini, navi e le applicazioni di monitoraggio.

6.1.1 Intelligenza Artificiale (IA)

Il termine Intelligenza Artificiale (IA) si riferisce in generale alle applicazioni della tecnologia per svolgere compiti che assomigliano alle funzioni cognitive umane ed è definito in generale come la capacità di una macchina di imitare il comportamento umano intelligente. L'IA coinvolge tipicamente la teoria e lo sviluppo di sistemi informatici in grado di svolgere compiti che normalmente richiedono l'intelligenza umana, come la percezione visiva, il riconoscimento vocale, il processo decisionale e la traduzione tra le lingue. Con riferimento alle definizioni date da Hintze, l'IA può essere catalogata in quattro tipi:

- Macchine reattive: questi sistemi di IA non hanno memoria e sono specifici per ogni compito. Un esempio è Deep Blue, il programma di scacchi dell'IBM che ha battuto Garry Kasparov negli anni '90. Deep Blue è in grado di identificare i pezzi sulla scacchiera e di fare previsioni, ma non avendo memoria non può utilizzare le esperienze passate per informare quelle future.

- Memoria limitata: questi sistemi di intelligenza artificiale hanno memoria, quindi possono utilizzare le esperienze passate per prendere decisioni future. Alcune delle funzioni decisionali delle auto a guida autonoma sono progettate in questo modo. Ma queste semplici informazioni sul passato sono solo transitorie, non vengono salvate come parte della libreria di esperienze da cui l'auto può imparare.

- Teoria della mente: è un termine della psicologia. Applicato all'IA, significa che il sistema avrà l'intelligenza sociale per comprendere le emozioni. Questo tipo di IA sarà in grado di dedurre le intenzioni umane e di prevedere il comportamento, un'abilità necessaria affinché i sistemi di IA diventino membri integranti dei team umani.

- Consapevolezza di sé: in questa categoria, i sistemi di IA hanno un senso di sé, che conferisce loro coscienza. Le macchine dotate di autocoscienza comprendono il proprio stato attuale.

Questo tipo di IA non esiste ancora. Tuttavia, l'IA è usata come termine generico che comprende un ampio spettro di tecnologie e applicazioni diverse, come mostrato nella Figura 2.1.

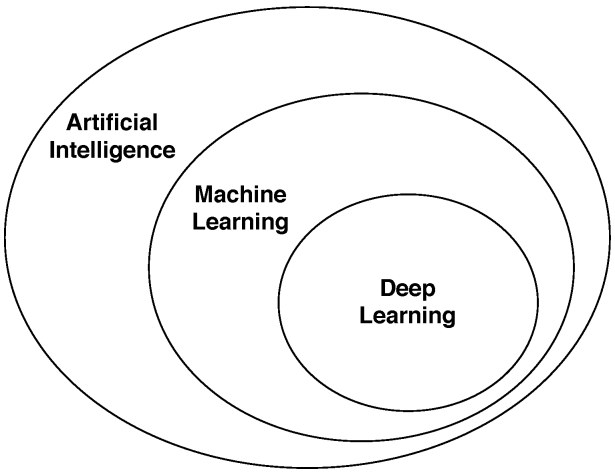


Figura 2 – Relazione tra Artificial Intelligence, Machine Learning e Deep Learning.

L'apprendimento automatico (ML) è un campo dell'informatica che utilizza algoritmi per elaborare grandi quantità di dati e imparare da essi. A differenza della programmazione tradizionale basata su regole, i modelli di ML imparano dai dati in ingresso per fare previsioni o identificare modelli significativi senza essere esplicitamente programmati per farlo. Esistono diversi tipi di modelli di ML, a seconda della loro funzione e struttura:

- Apprendimento automatico supervisionato: il modello viene addestrato con dati di input etichettati che si correlano a un output specifico. Viene continuamente perfezionato per fornire risultati più accurati man mano che si rendono disponibili ulteriori dati di addestramento. Dopo che il modello ha imparato dai modelli presenti nei dati di addestramento, può analizzare altri dati per produrre l'output desiderato. I risultati dei modelli di ML supervisionati sono in genere rivisti da persone che ne verificano l'accuratezza e vengono reinseriti nel modello per un ulteriore perfezionamento. L'apprendimento automatico supervisionato ha successo quando il modello è in grado di produrre costantemente previsioni accurate quando gli vengono forniti nuovi set di dati.

- Apprendimento automatico non supervisionato: i dati di input non sono etichettati e l'output non è specificato. Al contrario, i modelli vengono alimentati con grandi quantità di dati grezzi e gli algoritmi sono progettati per identificare qualsiasi modello significativo sottostante. modelli significativi. Gli algoritmi possono raggruppare dati simili, ma lo fanno senza alcuna nozione preconcetta dell'output. I risultati di questi modelli vengono poi interpretati dall'uomo per determinare se sono significativi e rilevanti.

- Apprendimento per rinforzo: il modello impara dinamicamente a ottenere l'output desiderato attraverso tentativi ed errori. Se l'algoritmo del modello funziona correttamente e ottiene l'output desiderato, viene premiato. Al contrario, se non produce l'output desiderato, viene penalizzato. Di conseguenza, il modello impara nel tempo a funzionare in modo da massimizzare la ricompensa netta. Ad esempio, nell'industria dei titoli, si stanno esplorando modelli di apprendimento rinforzato per il pricing e l'hedging delle opzioni.

Il deep learning (DL) è un sottodominio dell'apprendimento automatico. Con una potenza di calcolo accelerata e grandi insiemi di dati, gli algoritmi di apprendimento profondo sono in grado di auto apprendere modelli nascosti all'interno dei dati per fare previsioni. Nel tentativo di creare sistemi che apprendano in modo simile a quello degli esseri umani, l'architettura di base del deep learning è stata ispirata dalla struttura del cervello umano. Per questo motivo, alcune terminologie fondamentali dell'apprendimento profondo possono essere ricondotte alla neurologia. Analogamente a come i neuroni costituiscono i mattoni fondamentali del cervello, l'architettura del deep learning contiene un'unità computazionale che consente di modellare funzioni non lineari, chiamata perceptron. Analogamente a come un neurone nel cervello umano trasmette impulsi elettrici in tutto il nostro sistema nervoso, il perceptron riceve un elenco di segnali in entrata e li trasforma in segnali in uscita. Il perceptron mira a comprendere la rappresentazione dei dati impilando molti strati, dove ogni strato è responsabile della comprensione di una parte dell'input. Ogni strato di percettori è responsabile dell'interpretazione di un modello specifico all'interno dei dati. Una rete di questi perceptron imita il modo in cui i neuroni del cervello formano una rete, per cui l'architettura è chiamata rete neurale artificiale.

6.1.2 Artificial Neural Network

Il campo delle reti neurali artificiali è spesso chiamato semplicemente reti neurali o perceptron multistrato, uno dei tipi di rete neurale più utilizzati. Come già detto, il percettrone è un

modello a singolo neurone che è stato un precursore di reti neurali più grandi. Si tratta di un campo di studi che studia come semplici modelli di cervelli biologici possano essere utilizzati per risolvere compiti computazionali difficili, come la modellazione predittiva nell'apprendimento automatico. L'obiettivo non è creare modelli realistici del cervello, ma piuttosto sviluppare algoritmi e strutture di dati robusti da utilizzare per strutture di dati robusti che possiamo usare per modellare problemi difficili. La potenza delle reti neurali deriva dalla loro capacità di apprendere la rappresentazione dei dati di addestramento e il modo migliore per metterla in relazione con la variabile di uscita che si vuole prevedere. La capacità predittiva delle reti neurali deriva dalla struttura gerarchica o multistrato delle reti.

6.1.2.1 Neuroni

L'elemento costitutivo delle reti neurali sono i neuroni artificiali (Figura 2.2). Si tratta di semplici unità di calcolo che hanno segnali di ingresso ponderati e producono un segnale di uscita utilizzando una funzione di attivazione.

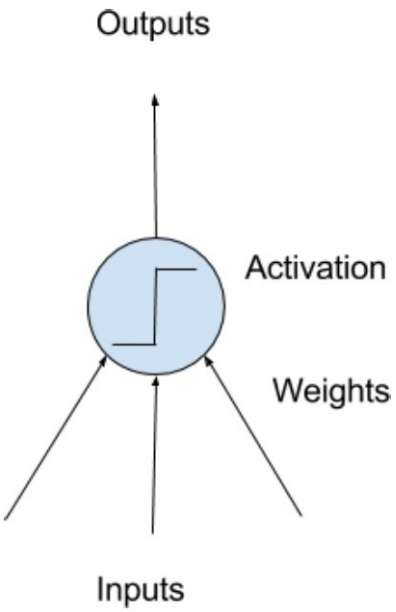


Figura 3 - Modello del neurone.

I pesi sono spesso inizializzati a piccoli valori casuali, come ad esempio valori compresi tra 0 e 0,3; pesi più grandi indicano una maggiore complessità e fragilità del modello. È auspicabile mantenere piccoli i pesi nella rete e si possono utilizzare tecniche di regolarizzazione. Ogni neurone ha anche un bias, che può essere considerato come un ingresso che ha sempre il valore 1,0 e anch'esso deve essere ponderato. Ad esempio, un neurone può avere due ingressi, nel qual caso richiede tre pesi: uno per ciascun ingresso e uno per il bias.

6.1.2.2 Funzioni di attivazione

Gli ingressi ponderati vengono sommati e fatti passare attraverso una funzione di attivazione, detta anche funzione di trasferimento. Una funzione di attivazione è una semplice mappatura degli ingressi ponderati sommati all'uscita del neurone. Si chiama funzione di attivazione perché regola la soglia di attivazione del neurone e l'intensità del segnale di uscita. I primi esempi di funzioni di attivazione erano semplici funzioni di attivazione a gradini in cui se la somma degli input era superiore a una soglia, ad esempio 0,5, il neurone emetteva un valore di 1,0, altrimenti emetteva un valore di 0,0. Tuttavia, la ragione principale per cui si utilizza una funzione di attivazione è che il neurone si attiva per un valore di 1,0, altrimenti si attiva per un valore di 0,0.

Tuttavia, la ragione principale per utilizzare una funzione di attivazione è quella di introdurre una non linearità nella rete neurale, per cui si utilizzano funzioni di attivazione non lineari. Ciò consente alla rete di combinare gli ingressi in modi più complessi e, a sua volta, di fornire una capacità più ricca di funzioni da modellare.

Di seguito vengono presentate le funzioni di trasferimento più comuni.

- Sigmoid

  La funzione

  $$y = \sigma(x) = \frac{1}{1 + exp^{-x}}$$

  si chiama funzione sigmoid. Una funzione sigmoid è una funzione limitata e differenziata, non decrescente e con un solo punto di inflessione. In parole povere, una funzione sigmoid è una curva liscia a forma di "S" (Figura 2.3). Ha una lunga tradizione nella teoria e nella pratica delle reti neurali. Una motivazione è rappresentata dai modelli di attivazione a forma di sigmoid osservati nelle neuroscienze. Un'altra motivazione è stata l'interpretazione delle funzioni sigmoid come approssimazioni matematicamente trattabili delle funzioni a gradini introdotte in precedenza.

  La ragione principale dell'uso della funzione sigmoid è che esiste tra (0 e 1). Pertanto, è particolarmente utilizzata per i modelli progettati per prevedere la probabilità come output. Poiché la probabilità di qualsiasi cosa esiste solo tra 0 e 1, la sigmoid è la scelta migliore.

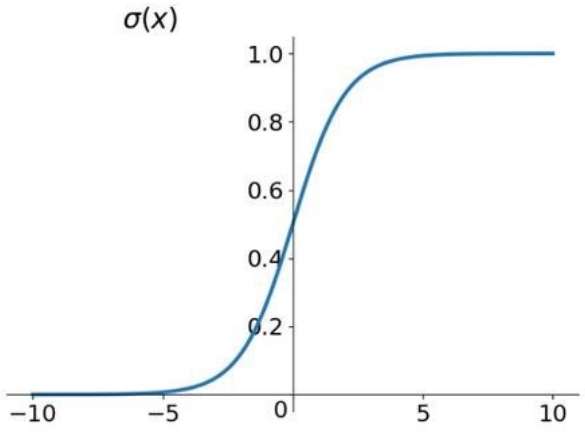


Figura 4 - Sigmoid shape.

- Tanh

  La funzione

$$y = \tanh(x) = \frac{exp^x - exp^{-x}}{exp^x + exp^{-x}}$$

  si chiama funzione tangente iperbolica (Tanh). È molto simile alla funzione di attivazione sigmoid e ha persino la stessa forma a S (Figura

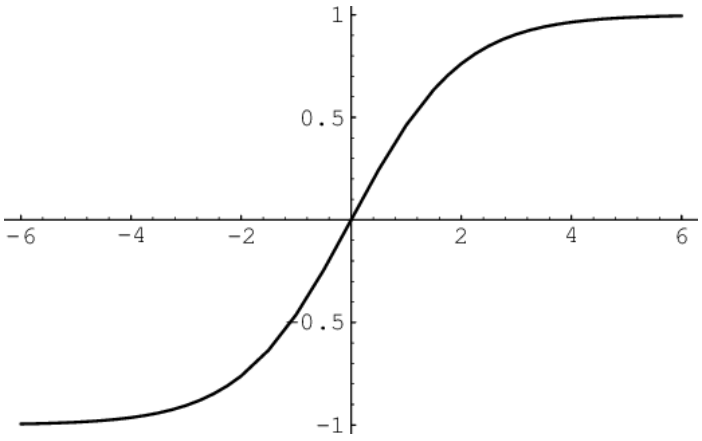


Figura 5 - Tanh shape.

  2.4).

  Il vantaggio principale rispetto alla funzione sigmoid è che gli ingressi negativi saranno mappati fortemente negativi e gli ingressi nulli saranno mappati vicino allo zero nel grafico tanh.

- ReLU

  La funzione

$$y = max\,(0, x)$$

è chiamata funzione di attivazione lineare rettificata. È anche chiamata funzione a parte positiva: la funzione a parte positiva è la funzione identità (lineare con pendenza 1) per argomenti positivi e la funzione

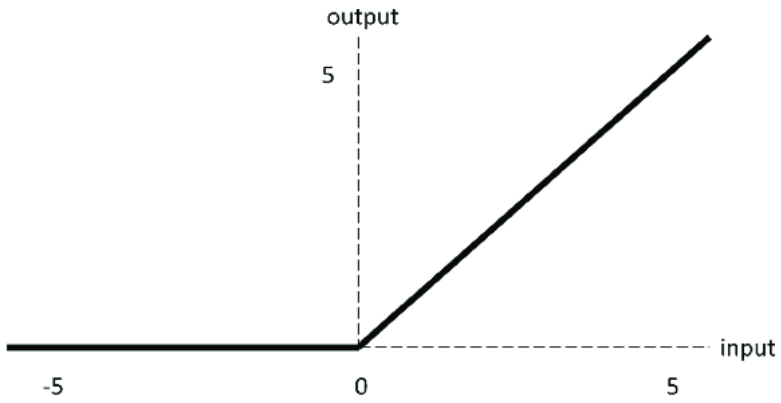


Figura 6 - ReLU shape.

costante con valore zero altrimenti (Figura 2.5).
La funzione a parte positiva lascia passare inalterati gli input positivi ma taglia quelli negativi, cioè trasforma gli input negativi e non negativi in output non negativi. Un chiaro vantaggio della ReLU è che sia la funzione stessa che le sue derivate sono facili da implementare e computazionalmente poco costose: per questo motivo i modelli che utilizzano la funzione di attivazione ReLU convergono più velocemente.

Il problema principale di questa funzione è il fenomeno della morte di ReLU. Un neurone ReLU è "morto" se è bloccato sul lato negativo e produce sempre 0. Poiché anche la pendenza di ReLU nell'intervallo negativo è pari a 0, una volta che un neurone diventa negativo, è improbabile che si riprenda. Questi neuroni non svolgono alcun ruolo nella discriminazione dell'input e sono essenzialmente inutili. Nel corso del tempo si può arrivare a una gran parte della rete che non fa nulla. La rilevanza del fenomeno "dying-ReLU", in pratica, rimane poco chiara. Tuttavia, alcune variazioni di questa funzione hanno portato ad altre funzioni di attivazione che mitigano questo problema, come LeakyReLU o PReLU.

- LeakyReLU

  La funzione

  $$y = max\,(0, x) + min\,(0, \alpha x)$$

è chiamata funzione di attivazione Leaky Rectified Linear Unit. L'idea alla base di LeakyReLU è quella di imitare ReLU, ma di evitare il fenomeno della morte di ReLU (Figura 2.6).
LeakyReLU è uguale a ReLU nel caso α = 0; per i parametri positivi α,

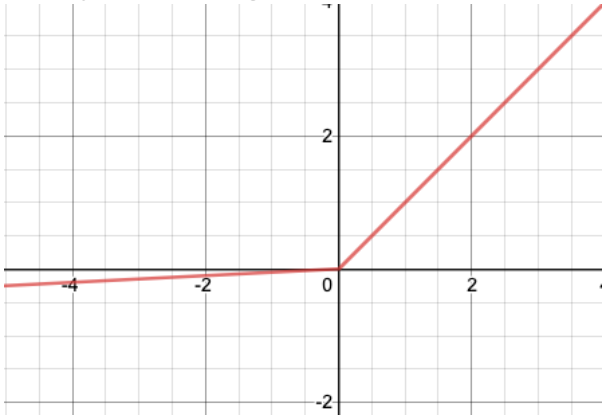


Figura 7 - LeakyReLU shape.

tuttavia, le funzioni differiscono per gli ingressi negativi, soprattutto nelle loro derivate. Una sfida pratica inflitta da LeakyReLU è la scelta del parametro α. Come si è visto, α è la pendenza di LeakyReLU per gli ingressi negativi. Di solito viene scelto tra 0 (dove LeakyReLU è uguale a ReLU) e 1 (dove LeakyReLU è uguale alla funzione lineare), ma non c'è ulteriore consenso: una raccomandazione popolare è quella di impostare α = 0,01; un'altra possibilità è quella di introdurre l'unità lineare leaky rectifier randomizzata (RReLU), che sostituisce il parametro fisso con uno stocastico. Tuttavia, i vantaggi di queste scelte rispetto alle altre e rispetto alla ReLU vanilla devono ancora essere esplorati.

- PReLU

  La funzione

  $$f(x) = max\,(0, x) + min\,(0, \alpha x)$$

  è chiamata funzione di attivazione Parametric Rectified Linear Unit. Come si può notare, è formalmente identica alla LeakyReLU; la differenza principale è che il termine α è esso stesso un parametro della rete che deve essere addestrato.

  In generale, PReLU introduce un numero molto ridotto di parametri aggiuntivi. Il numero di parametri aggiuntivi è pari al numero totale di canali, trascurabile se si considera il numero totale di pesi. Non si prevede quindi alcun rischio aggiuntivo di overfitting.

  Si può prendere in considerazione anche una variante con condivisione dei canali:

  $$f(x_i) = max\,(0, x_i) + min\,(0, \alpha x_i)$$

Il coefficiente è condiviso da tutti i canali di un livello. Questa variante introduce un solo parametro aggiuntivo in ogni strato.

6.1.2.3 Architetture delle ANN

I neuroni sono organizzati in reti di neuroni. Una fila di neuroni è chiamata strato e una rete può avere più strati. Fin dalla loro nascita, le reti neurali hanno sviluppato varie forme e tipi, tra cui le categorie di reti neurali superficiali e profonde. I termini superficiale e profondo si riferiscono al numero di strati di una rete neurale; le reti neurali superficiali si riferiscono a una rete neurale che ha un numero ridotto di strati, di solito considerati come aventi un singolo strato nascosto, mentre le reti neurali profonde si riferiscono a reti neurali che hanno più strati nascosti. Entrambi i tipi di rete svolgono determinati compiti meglio degli altri e la scelta della giusta profondità della rete è importante per creare un modello di successo.

Come mostrato nella figura successiva, in un'architettura generale di rete neurale si possono individuare tre tipi di strati:

- strato di input: è lo strato che prende gli input dal set di dati; è anche chiamato strato visibile, perché è la parte esposta della rete.
- strato nascosto: è uno strato intermedio tra gli strati di input e di output, in cui i neuroni artificiali accolgono una serie di input ponderati e producono un output attraverso una funzione di attivazione.
- strato di uscita: è l'ultimo strato nascosto ed è responsabile dell'uscita di un valore o di un vettore di valori che corrispondono al formato richiesto dal problema.

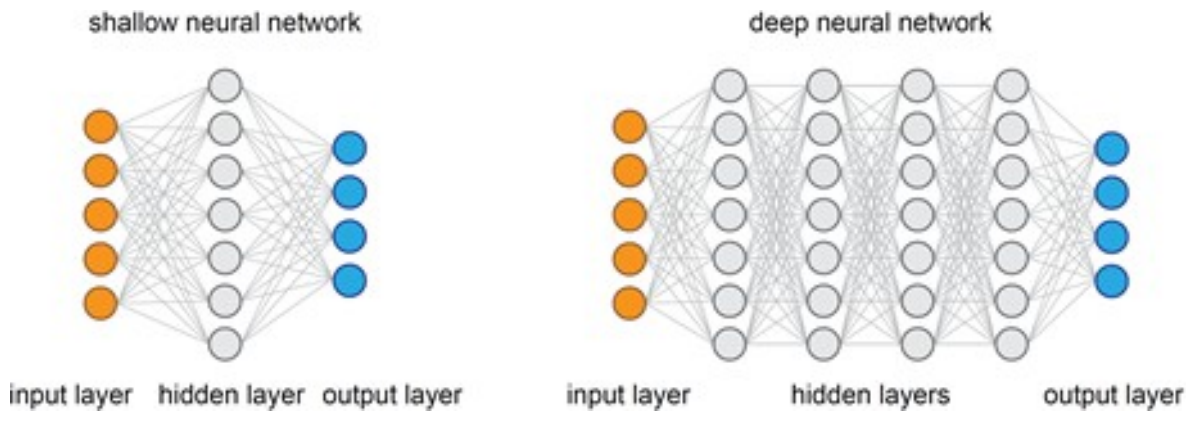


Figura 8 – Confronto tra l'architettura di rete neurale Shallow e Deep..

Quando le reti neurali sono state inizialmente sviluppate, sono nate prima le reti neurali poco profonde e la forma più comune è il perceptron multistrato, derivato dal modello di perceptron di Rosenblatt. Questa struttura permette alle reti neurali superficiali di apprendere caratteristiche importanti indipendentemente da altre caratteristiche, il che è più adatto per compiti di apprendimento con dati a bassa dimensionalità e un numero relativamente piccolo di caratteristiche.

Le reti neurali profonde, invece, eccellono in compiti molto complessi, con dati in ingresso di grandi dimensioni e ad alta dimensionalità. Queste reti hanno più strati nascosti, con strati inferiori che apprendono caratteristiche di basso livello e strati superiori che apprendono caratteristiche di livello superiore che dipendono dai modelli e dalle conoscenze derivate dagli strati inferiori. Il fattore dipendente dallo strato delle reti neurali profonde significa che l'apprendimento dipende da strato a strato e che gli strati nascosti riutilizzano le caratteristiche apprese dagli strati inferiori per apprendere caratteristiche più complesse e risolvere compiti più complessi.

6.1.3 Rete Neurale (Neural Netwrok NN)

La rete neurale opera in due modalità: formazione e predizione. Per l'addestramento e la predizione è necessario dividere il set di dati in set di addestramento e set di test. Come già detto, la modalità di addestramento inizia con valori arbitrari dei pesi e procede iterativamente. Ogni iterazione dell'intero set di addestramento è chiamata epoch. In ogni epoca la rete regola i pesi nella direzione che riduce l'errore (attraverso l'algoritmo di backpropagation). Continuando il processo iterativo di aggiustamento incrementale, i pesi convergono gradualmente verso l'insieme di valori localmente ottimali. Di solito sono necessarie molte epoche prima che l'addestramento sia completato.

In modalità di previsione, le informazioni scorrono attraverso la rete, dagli ingressi alle uscite. In questo caso non è stato utilizzato lo strato di backpropagation, ma solo lo strato feedforward. La rete elabora un campione di dati alla volta, producendo una stima dei valori di uscita in base ai valori di ingresso. L'errore risultante viene utilizzato come stima della qualità della previsione della rete addestrata.

Una proprietà importante delle reti neurali è la loro capacità di generalizzazione. Si dice che una rete generalizza bene quando la relazione ingresso-uscita calcolata dalla rete è corretta (o quasi) per modelli di ingresso/uscita mai utilizzati nell'addestramento della rete.

Durante il processo di addestramento di una rete neurale possono verificarsi due diversi tipi di problemi:

- Comportamento di underfitting: un modello che non riesce ad apprendere sufficientemente il problema e si comporta male su un set di dati di addestramento e non si forma bene su un campione di attesa.

- Comportamento di overfitting: un modello che apprende troppo bene il set di dati di addestramento, ottenendo buone prestazioni sul set di dati di addestramento ma non su un campione in attesa.

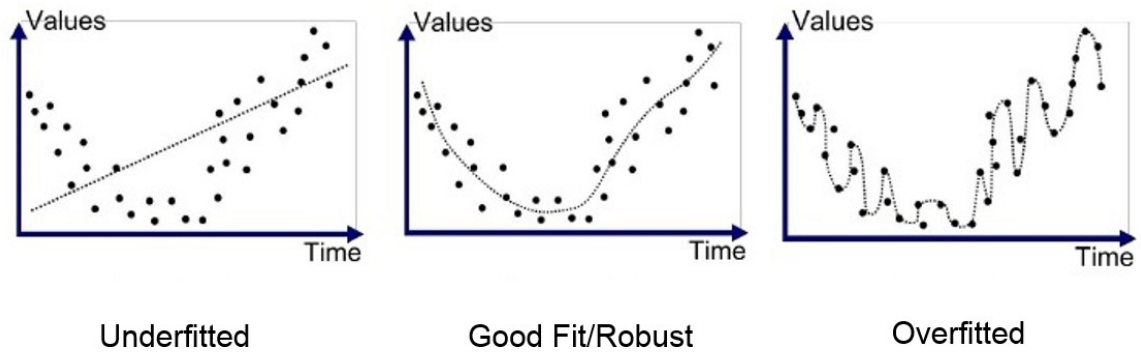


Figura 9 – Problemi della fase di Training.

L'underfitting può essere risolto aumentando la capacità del modello: una maggiore capacità significa che un modello può adattarsi a più tipi di funzioni per la mappatura degli input e degli output. L'aumento della capacità di un modello si ottiene facilmente modificando la struttura del modello, ad esempio aggiungendo più strati e/o più nodi agli strati.

L'overfitting è facilmente diagnosticabile monitorando le prestazioni del modello durante l'addestramento, valutandolo sia su un set di dati di addestramento sia su un set di dati di convalida in attesa. I grafici delle prestazioni del modello durante l'addestramento, chiamati curve di apprendimento, mostreranno due curve che sono specularmente la curva di

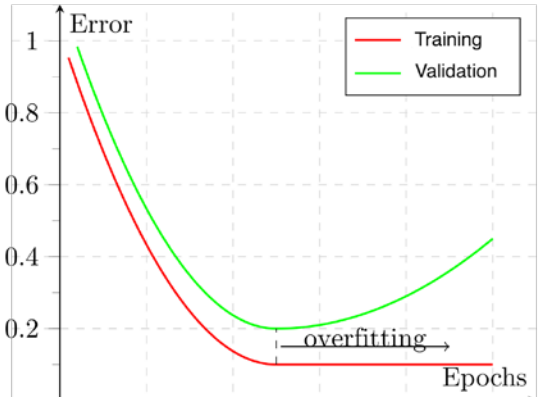


Figura 10

addestramento e la curva di validazione: se le due curve sono vicine significa che il modello sta generalizzando bene; se la curva di validazione si allontana da quella di addestramento significa che c'è un problema di overfitting (Figura 2.10).

Le possibili soluzioni per evitare l'overfitting sono la semplificazione del modello o l'utilizzo di tecniche di regolarizzazione come L1, L2, Dropout e Early Stopping.

6.1.4 Quantizzazione delle reti neurali

Le reti neurali sono algoritmi ad alta intensità di risorse, non solo in termini di costi computazionali ma anche di consumo di memoria. Anche se le risorse computazionali disponibili in commercio aumentano di giorno in giorno, l'ottimizzazione dell'addestramento e dell'inferenza delle reti neurali è estremamente importante. Quando i modelli vengono eseguiti sul bordo, l'ottimizzazione della rete diventa ancora più importante. Se i modelli vengono eseguiti su smartphone o dispositivi embedded, le limitazioni dell'hardware sono immediatamente evidenti.

Poiché sempre più modelli si spostano dai server all'edge, è essenziale ridurre le dimensioni e la complessità computazionale. Una tecnica è la quantizzazione, che sostituisce i numeri in virgola mobile con numeri interi all'interno della rete. L'idea fondamentale alla base della quantizzazione è che se si convertono i pesi e gli ingressi (che sono memorizzati in float32) in tipi interi, si consuma meno memoria e su determinati hardware i calcoli sono più veloci.

Tuttavia, c'è un compromesso: con la quantizzazione si può perdere una notevole precisione. Esistono due modi principali per effettuare la quantizzazione nella pratica:

-post-formazione;
-l'addestramento consapevole della quantizzazione.

6.1.4.1 Quantizzazione Post-training

La quantizzazione post-training è una tecnica di conversione che può ridurre le dimensioni del modello, migliorando al contempo la latenza della CPU e dell'acceleratore hardware, con un lieve degrado dell'accuratezza del modello. È possibile scegliere tra diverse opzioni di quantizzazione post-training, che vengono ora presentate.

- **Quantizzazione del range dinamico**

  La forma più semplice di quantizzazione post-addestramento quantizza staticamente solo i pesi da virgola mobile a numero intero, che ha una precisione di 8 bit. Al

momento dell'inferenza, i pesi vengono convertiti da 8 bit di precisione in virgola mobile e calcolati utilizzando kernel in virgola mobile. Questa conversione viene effettuata una sola volta e memorizzata nella cache per ridurre la latenza.
Per migliorare ulteriormente la latenza, gli operatori "dynamic-range" quantificano dinamicamente le attivazioni in base al loro intervallo a 8 bit ed eseguono i calcoli con pesi e attivazioni a 8 bit. Questa ottimizzazione consente di ottenere latenze vicine all'inferenza a virgola fissa. Tuttavia, le uscite sono ancora memorizzate in virgola mobile, per cui la velocità di calcolo con gli operatori a intervallo dinamico è inferiore a quella di un calcolo a virgola fissa completo.

- **Quantizzazione full integer**

  È possibile ottenere ulteriori miglioramenti della latenza, riduzioni dei picchi di utilizzo della memoria e compatibilità con dispositivi hardware o acceleratori solo interi assicurandosi che tutta la matematica del modello sia quantizzata in numeri interi.
  Per una quantizzazione intera completa, è necessario calibrare o stimare l'intervallo, cioè (min, max) di tutti i tensori in virgola mobile del modello. A differenza dei tensori costanti, come i pesi e i bias, i tensori variabili, come l'input del modello, le attivazioni (uscite degli strati intermedi) e l'output del modello, non possono essere calibrati a meno che non si eseguano alcuni cicli di inferenza. Di conseguenza, il convertitore necessita di un set di dati rappresentativo per la calibrazione. Questo set di dati può essere un piccolo sottoinsieme (circa 100-500 campioni) dei dati di addestramento o di validazione.

- **Quantizzazione Float16**

  Un'altra possibilità è quella di ridurre le dimensioni di un modello in virgola mobile quantizzando i pesi in float16, lo standard IEEE per i 16 bit. I vantaggi della quantizzazione in float16 sono:
  - ridurre le dimensioni del modello fino alla metà (poiché tutti i pesi diventano la metà della loro dimensione originale).
  - perdita minima di accuratezza.
  - il supporto di alcuni delegati (ad esempio il delegato della GPU) che possono operare direttamente sui dati float16, con una conseguente esecuzione più rapida rispetto ai calcoli float32.
  Gli svantaggi della quantizzazione in float16 sono invece:
  - non riduce la latenza quanto una quantizzazione in virgola fissa.
  - Per impostazione predefinita, un modello quantizzato in float16 "dequantizza" i valori dei pesi in float32 quando viene eseguito sulla CPU.

6.1.4.2 Training con quantizzazione

Come già detto, quando si passa a una precisione inferiore rispetto al float, in genere si nota un calo significativo dell'accuratezza, poiché si tratta di un processo con perdita. Il meccanismo dell'addestramento consapevole della quantizzazione è semplice: si inseriscono moduli di quantizzazione fasulli, chiamati moduli Quantize-Dequantize (QDQ), nei punti in cui avviene la quantizzazione durante la conversione del modello in virgola mobile in modello intero quantizzato, per simulare gli effetti del clamping e dell'arrotondamento dovuti alla quantizzazione in numeri interi. I moduli di quantizzazione fittizia monitoreranno anche le scale e i punti zero dei pesi e delle attivazioni. Una volta

terminato l'addestramento consapevole della quantizzazione, il modello in virgola mobile può essere convertito immediatamente in un modello intero quantizzato, utilizzando le informazioni memorizzate nei moduli di quantizzazione fittizi.

Pertanto, il grafico di addestramento viene modificato per simulare il comportamento a bassa precisione nel passaggio successivo del processo di addestramento. Questo introduce gli errori di quantizzazione come parte della perdita di addestramento, che l'ottimizzatore cerca di minimizzare durante l'addestramento. In questo modo, il QAT aiuta a modellare gli errori di quantizzazione durante l'addestramento e a mitigarne gli effetti sull'accuratezza del modello al momento dell'implementazione.

Tuttavia, il processo di modifica del grafico di addestramento per simulare un comportamento di precisione inferiore è complesso. Non tutti i tipi di livelli di attivazione sono compatibili con la quantizzazione. Ad esempio, le reti con strati di regressione richiedono che i tensori di uscita di questi strati non siano vincolati dall'intervallo della quantizzazione a 8 bit e possono richiedere una granularità di rappresentazione più fine di quella che la quantizzazione a 8 bit può fornire. Questi strati funzionano meglio se sono esclusi dalla quantizzazione. Anche il posizionamento di questi nodi QDQ è fondamentale per non compromettere le capacità di apprendimento del modello.

6.2 Risultati attività

6.2.1 Data-set preprocessing

Il dataset, già acquisito, consiste in circa 1.600.000 campioni di dati grezzi provenienti da un accelerometro ST a 3 assi. Poiché è meglio avere un set di dati bilanciato, in quanto la valutazione è più facile da eseguire poiché non ci sono bias, e di conseguenza gli errori di previsione sono inferiori, il bilanciamento del set di dati viene eseguito tramite l'algoritmo SMOTE.

Un approccio per affrontare gli insiemi di dati sbilanciati consiste nel sovracampionare la classe minoritaria. L'approccio più semplice prevede la duplicazione degli esempi nella classe di minoranza, anche se questi esempi non aggiungono nuove informazioni al modello. Invece, i nuovi esempi possono essere sintetizzati a partire da quelli esistenti. Questo è un tipo di incremento dei dati per la classe di minoranza e viene chiamato Synthetic Minority Oversampling Technique, o in breve SMOTE.

La SMOTE funziona selezionando esempi vicini nello spazio delle caratteristiche, tracciando una linea tra gli esempi nello spazio delle caratteristiche e disegnando un nuovo campione in un punto lungo tale linea. Questa procedura può essere utilizzata per creare tutti gli esempi sintetici necessari per la classe di minoranza. Come descritto nell'articolo, si suggerisce di utilizzare prima un sottocampionamento casuale per ridurre il numero di esempi nella classe maggioritaria, quindi di utilizzare SMOTE per sovracampionare la classe minoritaria e bilanciare la distribuzione della classe. L'approccio è efficace perché vengono creati nuovi esempi sintetici della classe di minoranza che sono plausibili, cioè relativamente vicini nello spazio delle caratteristiche agli esempi esistenti della classe di minoranza. Per questo problema, il set di dati è stato prima ridotto a circa 500.000 campioni per accelerare il processo di addestramento e poi è stato utilizzato SMOTE per bilanciare il set di dati da una distribuzione di classi dell'85%-15% a una distribuzione di classi del 50%-50%.

Prima di essere caricato sulla rete, il dataset viene sottoposto a una fase di pre-elaborazione che consiste in:

a. codifica delle etichette delle classi in uscita;
b. standardizzazione del dataset attraverso la funzione StandardScaler;
c. suddivisione del dataset in due parti: train e test.

La codifica delle etichette si riferisce alla conversione delle etichette in una forma numerica in modo da convertirle in una forma leggibile dalla macchina. Gli algoritmi di apprendimento automatico possono quindi decidere in modo migliore come devono essere gestite le etichette. Si tratta di un'importante fase di preelaborazione dei set di dati strutturati nell'apprendimento supervisionato. Quindi le due classi Ancora e Camminare sono codificate rispettivamente in 0 e 1.

Per quanto riguarda la standardizzazione dei dati, StandardScaler trasforma i dati in modo che la loro distribuzione abbia un valore medio pari a 0 e una deviazione standard pari a 1. Nel caso di dati multivariati, questa operazione viene eseguita in modo indipendente per ogni colonna dei dati. Data la distribuzione dei dati, a ogni valore del set di dati verrà sottratto il valore medio e quindi diviso per la deviazione standard dell'intero set di dati (o caratteristica nel caso multivariato).

Per stimare e validare le prestazioni della rete neurale, è stata utilizzata la procedura di divisione Train-Test. Questa procedura consente di effettuare previsioni sul dataset di test indipendentemente dal dataset di addestramento. La divisione del dataset in due parti permette alla Rete Neurale di sfruttare il dataset Train per l'addestramento e il dataset Test per la validazione. Per i modelli di rete neurale utilizzati in questo progetto, il dataset Train utilizza l'80% del dataset di partenza, mentre il dataset Test utilizza il restante 20% del dataset di partenza.

6.2.2 Implementazione della rete neurale

La prima linea di simulazioni si è concentrata sulla riduzione del numero di parametri e ha portato al modello poco profondo illustrato nella Figura seguente:

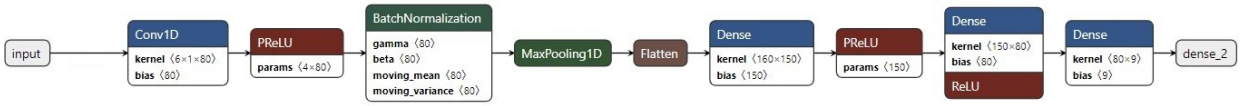


Figura 11 – Modello per la rete neurale Shallow Conv1D.

Include uno strato conv1D come strato di ingresso, deputato all'estrazione delle caratteristiche. Utilizza una funzione di attivazione PReLU ed è seguito da uno strato di normalizzazione dei lotti e da uno di pooling.

La normalizzazione dei lotti è una tecnica di addestramento che standardizza gli ingressi a uno strato per ogni mini-lotto. Questa procedura ha l'effetto di stabilizzare il processo di apprendimento e di ridurre drasticamente il numero di epoche di addestramento necessarie per formare le reti.

Un problema delle mappe di caratteristiche in uscita, che sono una mappatura di dove si trova un certo tipo di caratteristica nei dati in ingresso, è che sono sensibili alla posizione delle caratteristiche in ingresso. Un approccio per risolvere questo problema è quello di ridurre il campionamento delle mappe di caratteristiche. Ciò ha l'effetto di rendere le mappe di caratteristiche risultanti, campionate al ribasso, più robuste ai cambiamenti di posizione delle caratteristiche nel set di dati, definita "invarianza di traslazione locale". I livelli di pooling forniscono un approccio al campionamento ridotto delle mappe di caratteristiche sommando la presenza di caratteristiche in patch della mappa di caratteristiche. Due metodi comuni di pooling sono il pooling medio e il pooling massimo che riassumono rispettivamente la presenza media di una caratteristica e la presenza più attivata di una caratteristica. Per questo modello viene utilizzato il metodo max pooling.

L'ultima sezione della rete è una rete neurale completamente connessa che utilizza l'output del processo di convoluzione e predice gli output in base alle caratteristiche estratte nelle fasi precedenti.

6.2.3 Training ed addestramento

La curva di addestramento illustra come la rete ottimizza i suoi parametri durante la fase di addestramento. La curva di validazione illustra le prestazioni della rete sul set di dati Test. La curva di addestramento relativa al dataset Train illustra il comportamento di "apprendimento" del modello. Il dataset Test non è coinvolto in questa fase di addestramento. Inoltre, la valutazione della rete sul dataset Test fornisce un quadro chiaro della capacità inferenziale del modello. In questo caso, le due curve quasi si sovrappongono, indicando una buona progettazione della rete neurale.

Il Root Mean Square Error (RMSE) dei dati Cube rispetto al target (dati STIM) è pari a 5,20333. In questo caso, l'RMSE dei valori previsti (generati dalla Rete) rispetto ai valori target (STIM) è pari a 0,03362.

In termini di Errore Assoluto Medio (MAE) tra l'input (Cube) e il target (STIM), il risultato è pari a 4,00721. In questo caso, il MAE tra i valori predetti (generati dalla Rete) e i valori target (STIM) è pari a 0,02179.

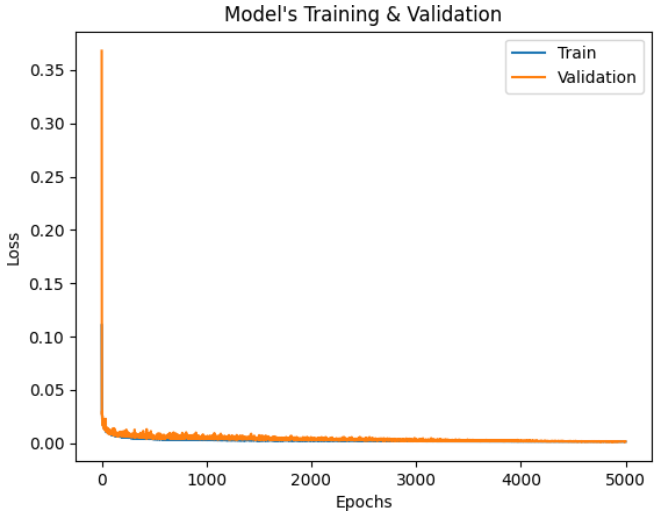


Figura 12 – Risultati di Training e Validation.

6.2.4 Implementazione della rete neurale sul microcontrollore

Per implementare i modelli di reti neurali sviluppati vengono utilizzati alcuni strumenti STM, come STM32CubeMX e STM32CubeIDE.

STM32CubeMX è uno strumento grafico che consente una configurazione molto semplice dei microcontrollori e microprocessori STM32. Il primo passo consiste nel selezionare un microcontrollore STM32 STMicrolectronics, un microprocessore o una piattaforma di sviluppo che corrisponda al set di periferiche richiesto. Per questo progetto è stato utilizzato un Nucleo STM32F401RE.

La seconda fase consiste nella configurazione di ogni software embedded richiesto, grazie a un risolutore di conflitti di piedinatura, un assistente per l'impostazione dell'albero di clock, un calcolatore del consumo energetico e un'utilità che configura le periferiche (come GPIO o USART) e gli stack di middleware (come USB o TCP/IP). Per distribuire i modelli di rete neurale sul micro si utilizza il pacchetto X-CUBE-AI.

X-CUBE-AI è un pacchetto di espansione STM32Cube che fa parte dell'ecosistema STM32Cube.AI e che estende le funzionalità di STM32CubeMX con la versione automatica di algoritmi di Intelligenza Artificiale pre-addestrati, tra cui reti neurali e modelli classici di apprendimento automatico e l'integrazione della libreria ottimizzata generata nel progetto dell'utente.

Offre inoltre diversi metodi per convalidare gli algoritmi di intelligenza artificiale sia su PC desktop che su STM32, nonché di misurare le prestazioni sui dispositivi STM32 senza che l'utente abbia realizzato codice C ad hoc.

È possibile caricare modelli sia nella versione Keras che in quella TFlite. Per questo compito viene scelta la versione Keras; quindi, il modello caricato viene analizzato e viene eseguita una convalida su desktop. A questo punto, è possibile generare il progetto che include il codice C con la rete neurale, e questo progetto viene importato automaticamente nello strumento STM32CubeIDE.

STM32CubeIDE è uno strumento di sviluppo multi-OS all-in-one, che fa parte dell'ecosistema software STM32Cube. STM32CubeIde è una piattaforma di sviluppo C/C++ avanzata con funzioni di configurazione delle periferiche, generazione del codice, compilazione del codice e debug per i microcontrollori e i microprocessori STM32.

Per quanto riguarda il codice C, nel file principale vengono forniti un vettore di ingressi e le classi di uscita reali corrispondenti per eseguire l'inferenza sul microcontrollore.

7 A.2.1. (RI) Raccolta della Base di Conoscenza da diverse sorgenti

Nella presente attività si organizzerà la raccolta dei dati: questi sono costituiti in parte dalle misurazioni provenienti dai sensori installati nelle strutture monitorate e in parte arricchiti da conoscenze scientifiche raccolte da pubblicazioni, brevetti, database, oltre che risultati di simulazioni numeriche on-demand.

Attività Svolta:

ATTIVITA' A) Analisi dei dati provenienti da varie fonti. Prove sulle tavole vibranti dell'università di Bergamo (*vedi capitolo specifico su "A.2.3. (RI) Definizione degli algoritmi di analisi descrittiva e cognitiva e dei relativi dati di input"*)

ATTIVITA' B) Pilot installation Technical Report (first trimester 2020)

8 Introduction

On December 4th an installation of 6 Sensors and communication Gateway took place in Beijing.

Over the monitored period and until the date of this report, 2 electrical power failures have been detected which required manual settings. The sBB system has been operational from January 11th up until February 11th and from February 27th up until March 31st and above, during which period the following Service Measures have been taken. It has to be noted that sensor #1 – the one in the lift pit - for unknown reasons stopped working from March, 1st up until March 20th.

Remark: it would however be appropriate to provide an extra sensor on the lift sling/cabin in order to perform a cross data analysis about Lift performance.

9 Service: Basement Acceleration (BA)

This service records the accelerations at the building basement and computes indicators (mean and range) based on the standard deviation of the three components. The goal is that of detecting the ground motion or events of significant impact for the stability of the building structure.

During the monitored period between January 11th and March 31st (it has to be noted that sensor #1 for unknown reasons stopped working from March, 1st up until March 20th) no significant events have been recorded. The mean standard deviations for the three components reported values around 2 mg during the whole period, while the ranges oscillated between 0.2 and 0.4 mg, with a few peaks above 0.6 mg.

In the below picture, we report the values recorded during 3 consecutive days (the data for the rest of the monitored period are fairly similar). No significant day-night variations are observed.

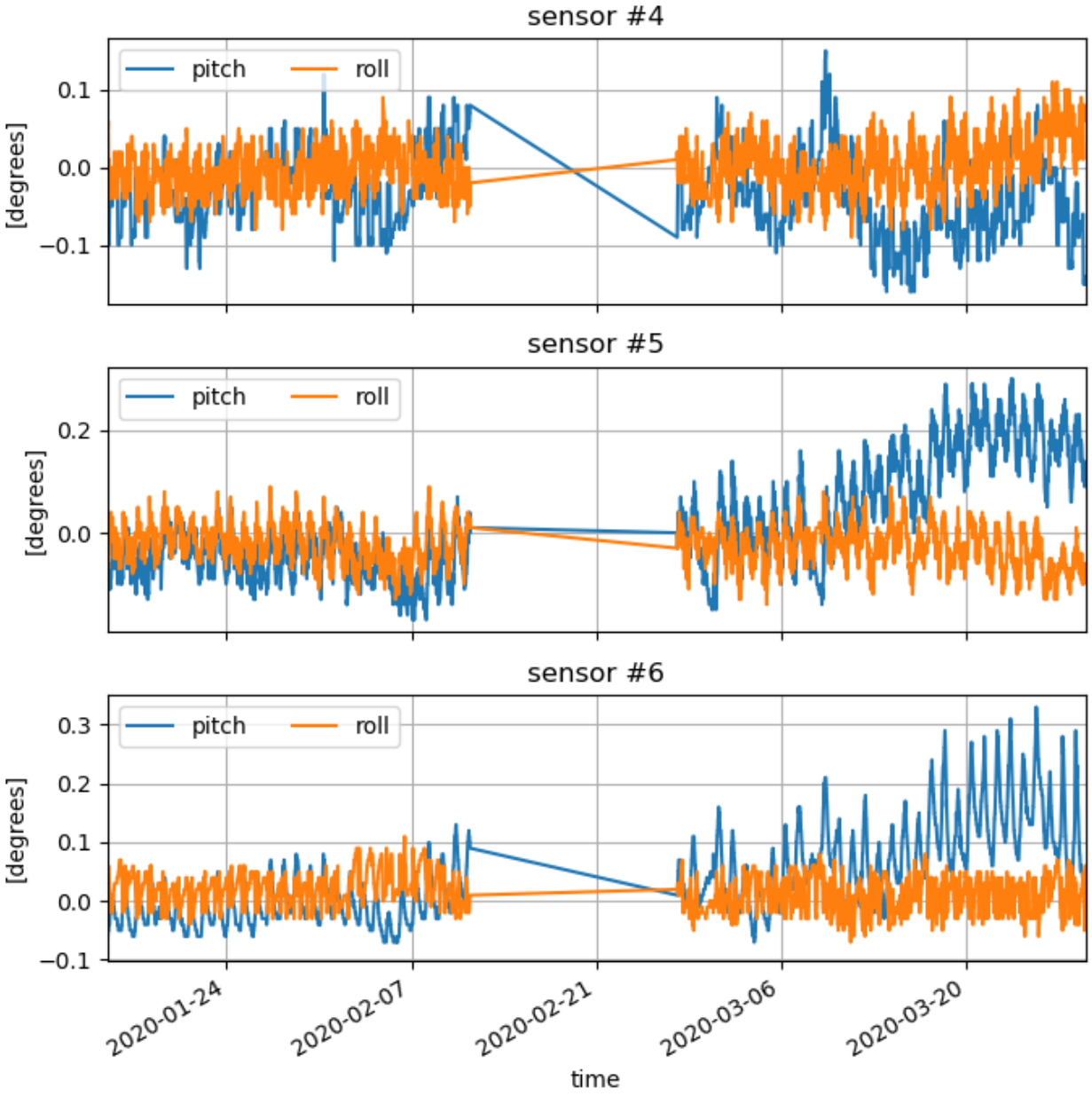
It has to however be noted that on March 5th it has been recorded by CEIC (China Seismological Network Center) a seismic movement of 2.1 magnitude on the Richter scale at 15Km deep at north of Beijing (north of the Olympic Green). Unfortunately, given the void recording period no detection was possible.

10 Service: Building Drift (BD)

This service checks the natural "movement" of the building and highlights anomalies, especially if progressive or due to extraordinary events such as earthquakes, violent weather, etc. The service is based on the measurement of the angles of rotation for evaluating variations in the verticality of the building.

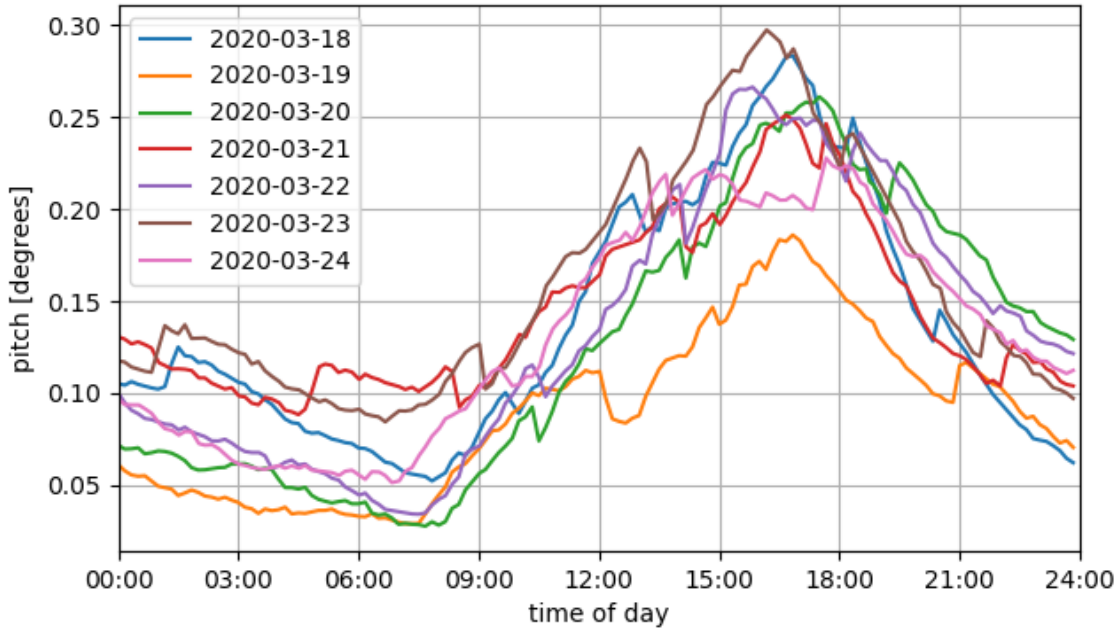
The measured quantities are the pitch and roll rotation angles. In the below picture we report the evolution of such two quantities during the monitored period, for the three uppermost

sensors (sensor #4, #5 and #6).



During the first observation windows (11/01/2020 - 11/02/2020) both the pitch and roll angles reported controlled oscillations, without showing any drift or significant deviation from the verticality (i.e. 0 degrees). During the second window (27/02/2020 - 31/03/2020), the pitch rotation angle reported a small drift for the two uppermost sensors. However, the magnitude of such phenomenon is not worrisome, since the peak is smaller than 0.3 degrees.

The oscillations that can be observed in the above picture are due to the natural day-night fluctuations of thermal nature. Indeed, in the picture below we report the daily-basis evolution of the pitch angle for the uppermost sensor (sensor #6), by superimposing the traces of 7 consecutive days.



By superimposing the traces, the effect of the natural thermic fluctuations is apparent from the picture.

In the table below we report the average temperatures in Beijing in the same period:

Day	18/03/2020	19/03/2020	20/03/2020	21/03/2020	22/03/2020	23/03/2020	24/03/2020
Average temperature	13.4 °C	11.1 °C	16.1 °C	11.2 °C	12.2 °C	12.7 °C	13.7 °C

(source: NOAA, National Oceanic and Atmospheric Administration, https://www.noaa.gov/)

We notice that the days for which we recorded the smallest gaps between the minimum and the maximum pitch angle correspond to the days with lowest average temperature (19/03 and 21/03), and hence to the smallest thermic expansion of the building components. On the other hand, the warmest day (20/03) is the one for which we recorded the largest gap between minimum and maximum pitch angle. These observations provide a validation for the recorded data.

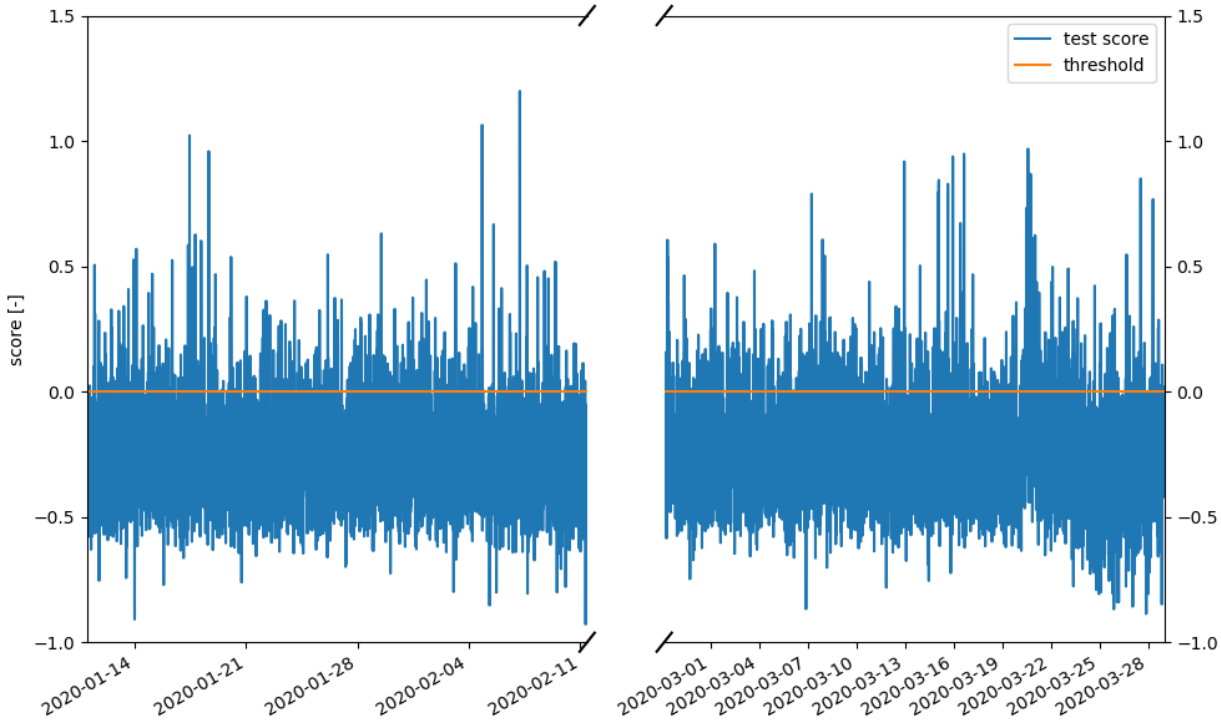
11

12 Service: Building Health (BH)

This service provides information of "non-structural" damage that could be potentially dangerous both for the building stability (in the medium to long term) and for the loss of value of the building itself. The survey is carried out on the basis of UNI 9916 (vibration effects on buildings) and ISO 4866 via measurement using accelerometers mounted on the lift guide rails.

The service provides a score, based on the UNI 9916 guidelines. A positive value of the score means that the recorded vibrations are above the threshold, a negative value entails that the vibrations are under control.

We remark that the UNI 9916 is referred to concrete buildings, while for buildings of different classes the threshold values should be accordingly adapted. Hence, it is not of concern that, as shown in the picture below, the score recorded in the monitored period often exceeded the threshold value.


test score
threshold
score [-]
1.5
1.0
0.5
0.0
−0.5
−1.0
2020-01-14
2020-01-21
2020-01-28
2020-02-04
2020-02-11
2020-03-01
2020-03-04
2020-03-07
2020-03-10
2020-03-13
2020-03-16
2020-03-19
2020-03-22
2020-03-25
2020-03-28

13 Service: Guide Rails Monitoring (GM)

This service provides real-time information on the status of the lift guide rails via measurement using sensors mounted on the same. This Service can also be regarded as **LIFT SAFE FOR USE**.

GM detects:

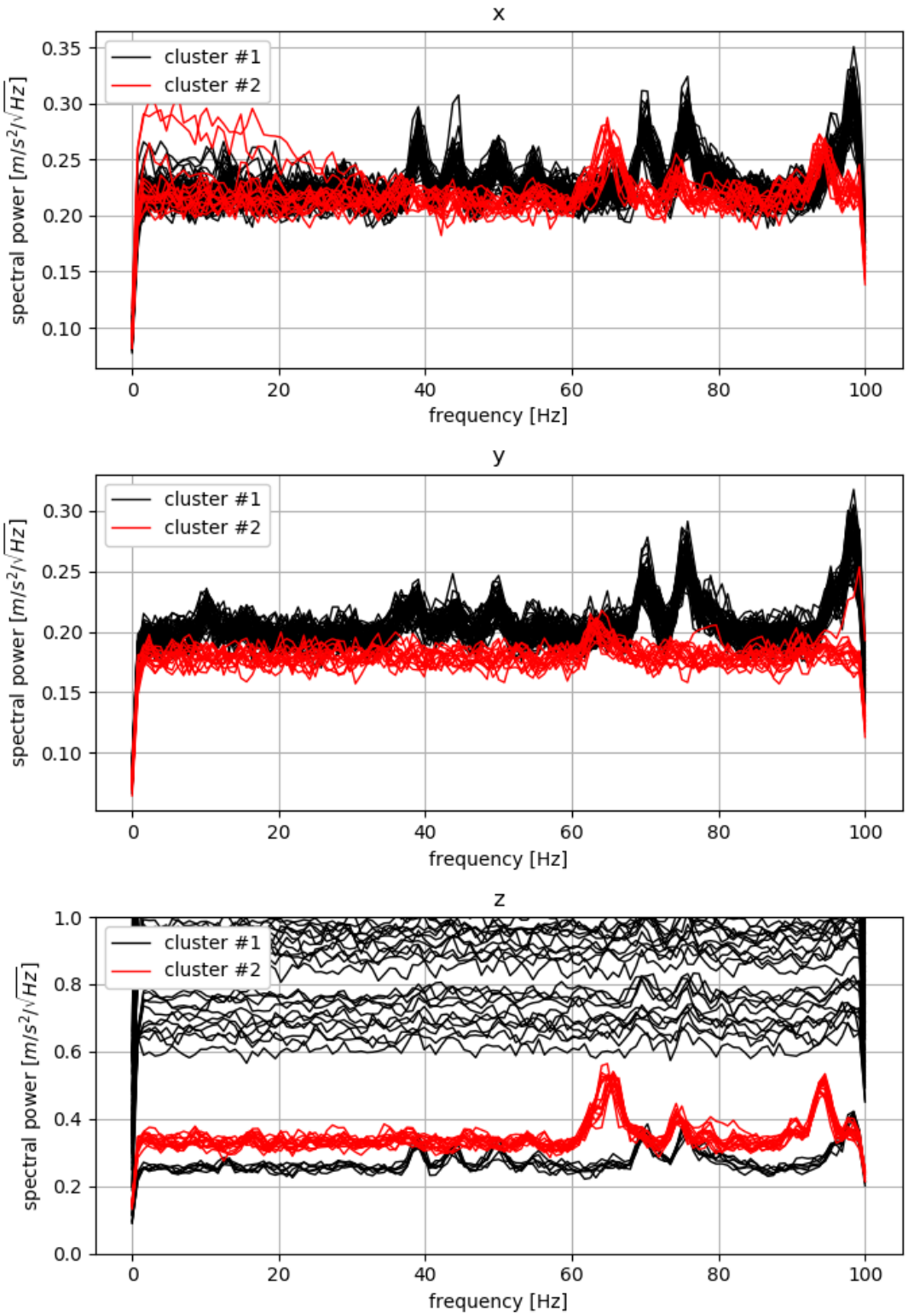
1. Abnormal inclinations
2. Guide rails fixing issues
3. Sling guide shoes or rollers issues via vibration Frequency Analysis (note: this analysis is to be compared with crossed readings obtain with a further sensor applied on the Lift Sling/Cabin).

In this regard, advanced analysis based on **anomaly detection algorithms** are under development. Such algorithms detect the **typical patterns** contained in the spectrum of the accelerometric signals measured on the lift guide rails, that represent the characteristic signature of the lift components. Failure of lift components can be detected by alterations in the patterns detected in the spectrum. Hence, by a comparative analysis of the measured spectra, sBB will allow to detect a malfunctioning before it produces consequences detectable by the final users (that are typically of more serious kind). Specifically, when the algorithm detects significant deviations from the baseline patterns, it raises a warning signal.

We have performed two pilot analysis on the spectra collected during the monitored period. The first one concerns the signals collected from the sensor located close to the building basement, while the second concerns signals collected close to the driving motor.

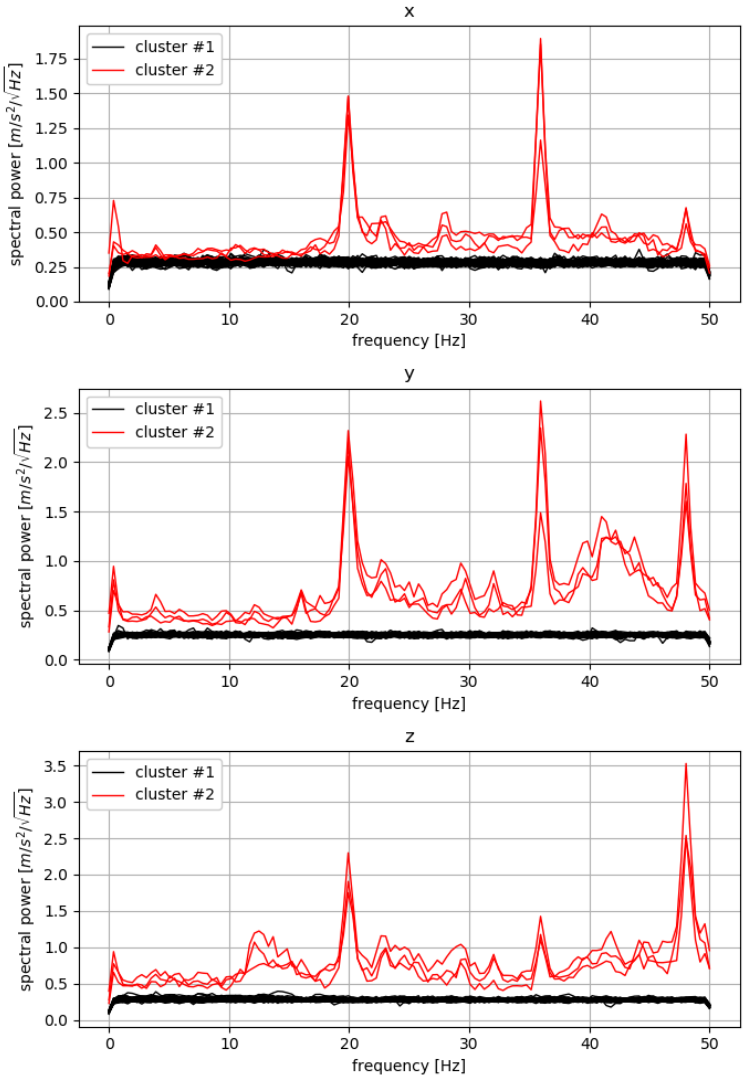
In the next page we report the spectra measured along the three axes, at different times, close to the building basement. We detected two **clusters**, highlighted in black and red (cluster #1 and cluster #2, respectively). The two clusters correspond to two different regimes of the lift (that may correspond to the times when the lift is moving and the times when it is not).

Being the sensor located close to the building basement, the energy contained in the two regimes is not significantly different. Nonetheless, we remark that (except for the z axis, for which the spectra are widespread due to the higher energy content associated with the vertical direction) the patterns are fairly auto-similar, which makes the application of anomaly detection algorithms promising.


x
cluster #1
cluster #2
spectral power [$m/s^2/\sqrt{Hz}$]
frequency [Hz]
y
cluster #1
cluster #2
spectral power [$m/s^2/\sqrt{Hz}$]
frequency [Hz]
z
cluster #1
cluster #2
spectral power [$m/s^2/\sqrt{Hz}$]
frequency [Hz]

In the next page, conversely, we report the spectra computed from measurement taken from the uppermost sensor (the one that is closest to the driving motor). Also in this case we detected two well-separated **clusters**. The two clusters correspond to the times when the lift is moving (red cluster) and the times when it is not (black cluster). In the second case, the energy involved is remarkably higher than the first case (and also higher than that of the spectra measured close to the basement), thanks to the closedness to the driving motor. As a matter of fact, the red cluster is associated to an energy level that allows to easily distinguish their spectra from the spectrum of the background noise. This allows to effectively detect the moments when the lift is moving.

Moreover, we remark that the spectra associated to the red clusters are auto-similar, and the characteristic peaks are located always in same positions. Those peaks correspond to the **proper frequencies of the main components of the lift and of the driving motors**. This key-result testifies that our analysis would allow to detect failures of such components that would yield to shifts in those peaks. Even if we do not dispose, at the moment, of enough data to show the effectiveness of our approach, those results are promising in this regard.



Documentazione:

Vedi attività svolta.

14 A.2.2. (RI) Riordino e connessione dei dati in una struttura organica

I dati raccolti nella prima fase sono in generale scollegati ed eterogenei, per cui verranno collegati e uniformati in funzione del tipo di analisi che si vuole effettuare in seguito. La seconda fase consiste quindi nella riorganizzazione dei dati e nell'estrazione delle caratteristiche principali, necessarie per le fasi successive. I dati verranno inoltre pre-filtrati e puliti da anomalie ed errori di misura.

Attività Svolta:

Sviluppo firmware sensori STWIN. Individuazione gateway e sviluppo firmware per l'invio dei dati su cloud. Analisi dati raccolti, validazione e corrispondenza dati ed evento. Strutturazione DB e infrastruttura cloud.

ATTIVITA' A) Sviluppo architettura sensori-gateway-cloud

Sensori

Firmware sviluppato in linguaggio C con le seguenti funzionalità:

• Lettura dei sensori e archiviazione dei dati in buffer pool circolari

• Ricerca di connettività Wi-Fi e connessione gateway

• Trasmissione dati al gateway

• Interprete dei comandi per eseguire le funzioni richieste dal gateway:

- richieste di aggiornamento del firmware

- aggiornamento dei parametri di lettura dei sensori e attivazione dei sensori

- richiesta trasmissione dati

• Watchdog per riavviare il sensore in caso di malfunzionamento

• Ripristinare il firmware alla funzionalità minima nel caso in cui non vengano ripristinate le normali condizioni di funzionamento

Portale

Sul gateway sono attivi diversi processi con esecuzione parallela, schedulati tramite PM2.

• Servizio di ascolto su coda MQTT per ricezione comandi da cloud, tra cui:

- Scaricare e installare l'aggiornamento del firmware del sensore
- Modificare i parametri di acquisizione del sensore
- Aggiornamenti dell'algoritmo
- Aggiunta/cancellazione di algoritmi
- Riavvio catena sensori
- Controllo dello stato di acquisizione del sensore
- Apertura tunnel SSH per abilitare l'accesso remoto (solo per supporto tecnico)

• Software di comunicazione con sensore, che:

- Invia aggiornamenti ai sensori (firmware, modifica parametri, ecc.)
- Riceve le misurazioni e le scrive in un DB Redis

• Algoritmi

• Servizio che invia i risultati dell'algoritmo al cloud tramite la coda MQTT

Portale tecnico

• Istanziazione di nuovi gateway

• Apertura tunnel SSH per il controllo diretto dei gateway

• Invia comandi a un singolo gateway

• Invia comandi massivi a una serie di gateway in modalità asincrona

• Verifica dell'esito dei comandi inviati

• Registri delle operazioni eseguite

• Comandi disponibili:

- Programmazione degli algoritmi

- Modifica la pianificazione degli algoritmi

- Attivazione sensori

- Modificare i parametri di lettura dei sensori

- Reimpostazione dei gateway

- Reset sensori

• Visualizzazione stato gateway e sensori

• Visualizzazione degli avvisi

• Configurazione dell'app installatori

Servizio web e servizi cloud

• Gestione della comunicazione tra portale e gateway

• Servizio per l'invio di comandi in modalità asincrona

• Servizio di gestione delle code MQTT, che:

- elabora i segnali ricevuti

- ricevere segnali di allerta ed esito dei comandi inviati

App installatori

• Segnalazioni di interventi da eseguire

• Registrazione struttura su database

• Configurazione del gateway

• Attivazione del gateway

• Attivazione dei sensori

• Test dei sensori

Portale utenti

• Visualizzazione delle strutture e del loro stato

• Visualizzazione dei dati storici dei risultati degli algoritmi

ATTIVITA' B) Technical Specifications Algorithms & Sensors

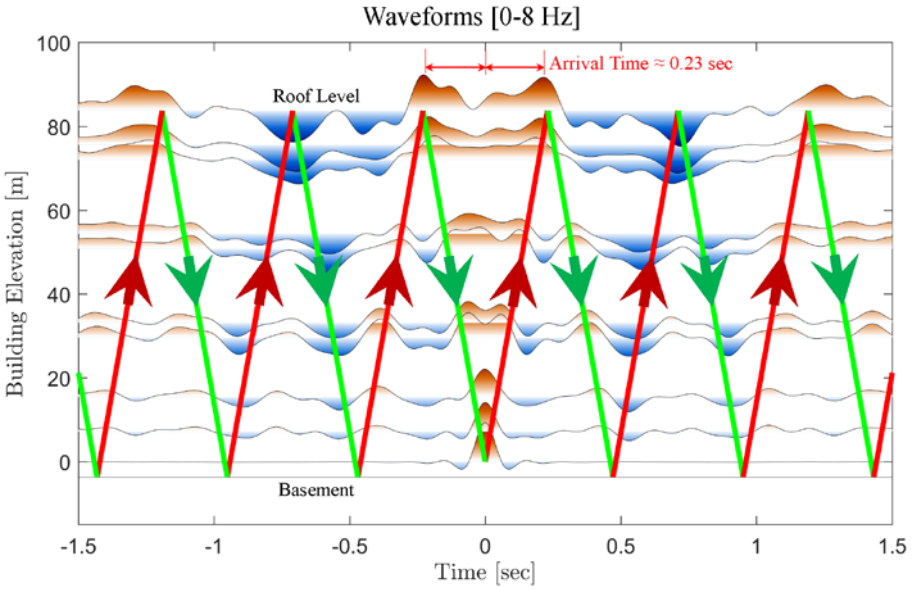
15 1. Sincronizzazione Sensori

La trasmissione sincrona RS485, a differenza di quella Wifi, permette di sincronizzare i clock dei sensori con accuratezza superiore.

Vantaggi di una sincronizzazione cablata

Gli algoritmi previsti fino ad oggi non necessitano di una sincronizzazione particolarmente accurata (anche 1 secondo di margine può essere tollerabile). La sincronizzazione è richiesta ad esempio nel momento in cui il sensore di cabina deve informare se l'ascensore è fermo o in movimento, perché l'analisi spettrale effettuata sulle guide deve confrontare i valori misurati con soglie diverse a seconda dei due casi. Tuttavia, 1 secondo di margine è tollerabile in questo caso.

Avere sensori sincronizzati con alta precisione permetterebbe si sviluppare altri algoritmi, quali algoritmi che analizzano come un segnale si propaga in verticale lungo l'edificio (ad esempio riconoscendo forme d'onda a diversi piani, e calcolando il ritardo fra un piano e l'altro). Un esempio è riportato nell'immagine seguente, riportante un'analisi effettuata nel laboratorio LISS (MIT).



Fonte: http://web.mit.edu/liss/research-sensing.html

Decisione: WiFi (per i nostri servizi non è necessaria una sincronizzazione eccessivamente accurata)

16 2. Valutazione del livellamento della cabina al piano

Le seguenti tecniche per la valutazione del livellamento al piano possono essere considerate:

- Utilizzo di **magnetometri**.
- Utilizzo di **sensori a effetto Hall**.

- Utilizzo di **sensori ottici**. Una problematica che può sorgere è quella del l'accumulo di sporco sul tag di lettura. Possibile introdurre una "spazzola" che pulisce il tag ad ogni passaggio dell'ascensore?

17 3. Valutazione della posizione della cabina

Si riportano sotto 4 strategie differenti per valutare il piano a cui si trova la cabina. Le strategie sotto riportate possono essere combinate fra di loro, per migliorare l'accuratezza della stima.

17.1.1 Valutazione quota tramite barometro

Un barometro è spesso utilizzato nel monitoraggio ascensori per ricavare la quota dell'ascensore (vedi sistema Schindler).

~~La misurazione della quota tramite barometro può essere soggetta a variazioni meteo. Per compensare, occorrerà posizionare un barometro anche nel sensore di fossa per poter valutare la differenza di pressione fra fossa e cabina.~~

Usare un solo barometro e ragionare per delta.

Sensori necessari: 1 barometro (+ tecnica per resettare la quota periodicamente)

Accuratezza: 2 piani circa (Fonte: Schindler. Confermato da A. Cigada)

17.1.2 Integrazione dell'accelerazione dell'accelerometro di cabina.

Integrando due volte l'accelerazione è possibile avere una stima dello spostamento della cabina. Nota l'altezza di ciascun piano, è quindi possibile stimare il piano a cui la cabina si trova.

In questo caso si renderebbe necessario il reset periodico della quota (altrimenti il drift porterebbe a errori che si accumulano nel tempo).

Sensori necessari: 1 accelerometro in cabina (+ tecnica per resettare la quota periodicamente)

Accuratezza: Da valutare

Confronto delle intensità delle accelerazioni lungo le guide.

In linea di principio, i sensori più vicini alla cabina riportano accelerazioni superiori. Questo permetterebbe di individuare il piano a cui si trova la cabina, quando questa si muove. Tuttavia, le guide trasmettono molto bene le accelerazioni, e quindi questo può dare solo una stima del piano a cui si trova la cabina.

Sensori necessari: accelerometri lungo le guide.

Accuratezza: 2-3 piani

17.1.3 Sensori di prossimità (vedi sezione sulla valutazione del livellamento).

Posizionando un magnetometro (o altri sensori di prossimità) sul piano della cabina e una tacca magnetica ad ogni piano, è possibile valutare il piano "contando" il numero di tacche attraversate. Almeno una delle tacche dovrà avere un'intensità di campo magnetico differente,

per poter resettare il sistema ogni tanto (in caso si perda una rilevazione). Per gli altri piani, per valutare se l'ascensore sta salendo o scendendo, è sufficiente guardare il segno dell'accelerazione verticale dell'accelerometro di cabina. Se si utilizzano sensori ottici, invece, ciascun piano può avere un tag differente.

Sensori necessari: 1 magnetometro + tacche magnetiche a ogni piano (possibilmente differenziate)

Oppure

1 sensore a effetto Hall

Oppure

1 lettore ottico

Accuratezza: 1 piano

18 4. Utilizzo di giroscopi

Un tema da approfondire è quello dell'utilizzo di giroscopi. In certi contesti, i giroscopi possono dare informazioni che gli accelerometri non possono dare.

Gli algoritmi attualmente sviluppati utilizzano accelerometri come inclinometri (mediando le accelerazioni su finestre di 1 minuto). Questo fornisce una stima dell'inclinazione dell'edificio con una buona accuratezza (si è dimostrato che si riescono a cogliere le fluttuazioni termiche con buona precisione, e quindi è presumibile che si riescano a misurare anche le inclinazioni dovuti a cedimenti strutturali).

Un giroscopio si rivela vantaggioso quando si vogliono misurare le inclinazioni in campo dinamico, anziché in campo statico. Questo può essere utile soprattutto in caso di **evento sismico**. Quanto questa informazione sia di valore per la valutazione del danno o in generale nell'ambito dell'analisi sismica, è una questione da indirizzare a un esperto di sismica.

P. Riva: conferma che il giroscopio è utile solo durante l'evento sismico

Rotazioni angolari di picco ottenute con simulazioni FEM su telaio in calcestruzzo (S. Castelli 2020-07-10):

- Terremoto forte (L'Aquila): 175 mrad/s
- Terremoto medio: 35 – 40 mrad/s
- Vibrazioni ambientali: 0.36 mrad/s o meno

19 5. Utilizzo di Termometri

L'aggiunta di termometri può rendersi utile a tre scopi differenti:

- Correzione della calibrazione degli altri strumenti
- Aggiunta delle misurazioni della temperatura alla dashboard
- Posizionando un termometro in prossimità del motore, è possibile valutarne lo sforzo oppure individuare eventuali surriscaldamenti anomali?

Cigada: la dipendenza dalla temperatura dipende molto da pezzo a pezzo per i MEMS, quindi questo vanifica ogni sforzo

Cigada: attenzione alla posizione (vicino a una scheda può leggere il surriscaldamento della scheda e non la temperatura ambiente)

20 6. Utilizzo di Igrometri

Come per i termometri, gli igrometri possono servire a:

- Correzione della calibrazione degli altri strumenti (forse meno importante che i termometri)
- Aggiunta delle misurazioni di umidità alla dashboard

21 7. Aleatorietà nella frequenza di campionamento dei sensori

L'utilizzo dei sensori STEVAL nell'installazione pilota ha evidenziato una criticità, legata a una frequenza di campionamento effettiva, che si discosta da quella nominale. In certi casi, sensori con frequenza nominale di 200 Hz si trovano a leggere a 189 Hz. Inoltre, ciascun sensore ha un a frequenza di campionamento che può variare nel tempo.

Al momento si stanno mettendo in atto strategie per limitare l'impatto di questo aspetto, quali la riassegnazione dei timestamp in modo uniforme su intervalli di 4 secondi, e una successiva regolarizzazione su intervalli di 1 minuto.

Si noti che alcuni algoritmi possono venire impattati da questa problematica, mentre altri no. In particolare, vengono impattati tutti gli algoritmi che coinvolgono la variabile temporale (analisi modale, integrazione di segnali) mentre non vengono impattati (se non in minima parte) quelli per cui importano soltanto i valori assoluti delle misure (come quelli basati su RMS o sulla valutazione dell'inclinazione).

In particolare, gli algoritmi potenzialmente impattati sono:

- Building Integrity
- Buiding Health
- People Disturbance
- Guide Rails monitoring (se fatta analisi spettrale)
- Doors monitoring (se fatta analisi spettrale)
- Usage statistics (se il piano viene ricavato con doppia integrazione dell'accelerazione)

22 8. Specifiche sensori

Accelerometro

Fondo scala	± 2 g
Risoluzione	1 mg, ma sarebbe meglio 0.1 mg
Banda	400 Hz

NB: queste specifiche sono compatibili con il sensore ST IIS2DH, ora in uso nell'impianto installato presso R. Zappa.

Giroscopio

Fondo scala	±100 dps
Risoluzione	al più 5 mdps
Banda	400 Hz

NB: Rispetto al sensore ST ISM330DHCX, pensato soprattutto per utilizzi di moto di oggetti o di macchine, qui siamo interessati a range di velocità angolari e a frequenza più basse. Queste specifiche sono infatti vicine all'estremo inferiore di quelle configurabili con il sensore ST (il fondo scala va da ± 125 dps a ± 4000 dps, la banda fino a 6 kHz). Non avendo noi bisogno di misurare movimenti così rapidi, è preferibile tenere un fondo scala non oltre i 100 dps e guadagnare in risoluzione.

Punti aperti

- Sarebbe opportuno (qualora fosse possibile caratterizzate almeno in modo approssimato la risposta dei sensori alla temperatura) dotare i sensori di termometro per effettuare la compensazione.
- Certificazione dei sensori (uno a uno).

NB: per le applicazioni building, non occorre differenziare i sensori in base alla collocazione.

Nel seguito si riportano valori di riferimento delle grandezze fisiche in gioco durante eventi sismici e a fronte dei soli rumori ambientali.

23 Eventi sismici

"ShakeMap Scientific Background. Rapid Instrumental Intensity Maps". *Earthquake Hazards Program*. U. S. Geological Survey. Archived from the original on 23 June 2011.

Instrumental Intensity (Mercalli)	Acceleration (g)	Velocity (cm/s)	Perceived shaking	Potential damage
I	**< 0.0017**	< 0.1	Not felt	None
II–III	**0.0017 – 0.014**	0.1 – 1.1	Weak	None
IV	**0.014 – 0.039**	1.1 – 3.4	Light	None

V	**0.039 – 0.092**	3.4 – 8.1	Moderate	Very light
VI	**0.092 – 0.18**	**8.1 – 16**	Strong	Light
VII	0.18 – 0.34	**16 – 31**	Very strong	Moderate
VIII	0.34 – 0.65	**31 – 60**	Severe	Moderate to heavy
IX	0.65 – 1.24	**60 – 116**	Violent	Heavy
X+	> 1.24	> **116**	Extreme	Very heavy

Rotazioni angolari di picco ottenute con simulazioni FEM su telaio in calcestruzzo (S. Castelli 2020-07-10):

- Terremoto forte (L'Aquila): 175 mrad/s = 10 dps
- Terremoto medio: 35 – 40 mrad/s = 2 – 2.3 dps
- Vibrazioni ambientali: 0.36 mrad/s = 0,02 dps (più o meno)

NB: 1 mrad/s = 0.001*180/pi = 0.0573 dps (degrees per second)

24 Frequenza proprie edifici

0.5 - 2 Hz (fonte: P.Riva)

25 Drift

Picco (fonte: P. Riva):

- Cemento armato: 1 %
- Muratura: 0.5 %

26 UNI 9916

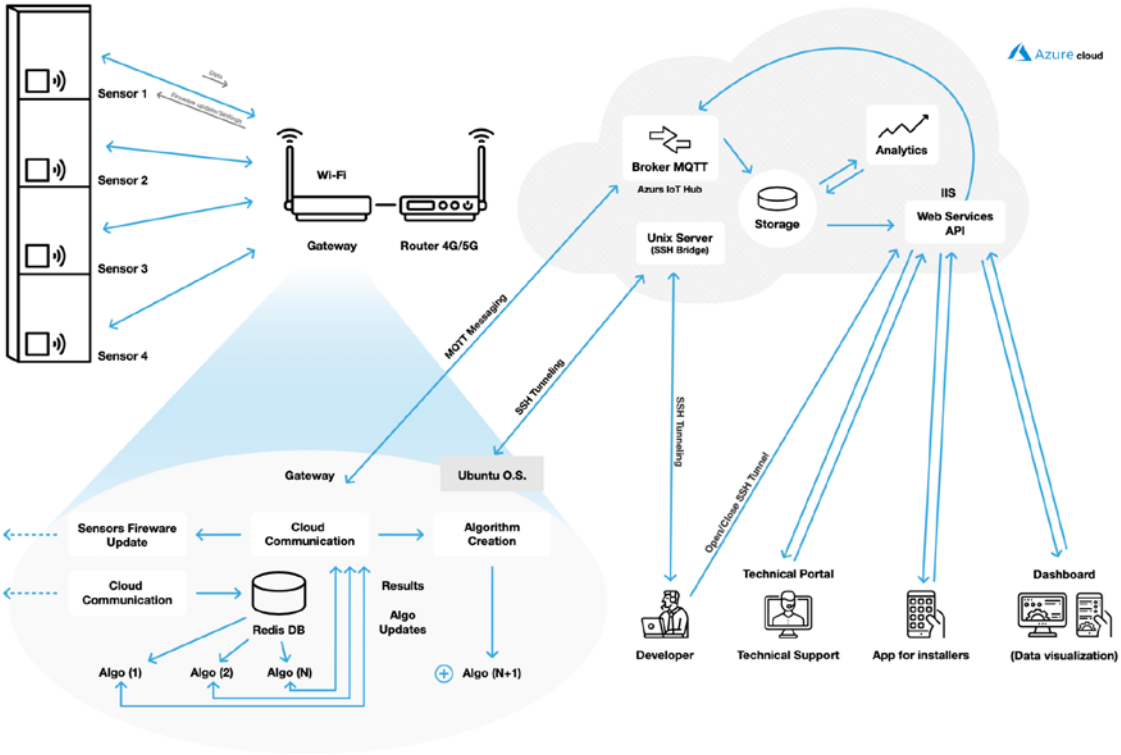
Soglie p.c.p.v. (**mm/s**)

		fondazioni			**Piano alto**	**Solai (comp. Verticale)**
		1-10 Hz	**10-50 Hz**	**50-100 Hz**	**tutte**	**tutte**
1	Costruzioni industriali, edifici industriali e costruzioni strutturalmente simili	20	20 – 40	40 – 50	40	20

2	Edifici residenziali e costruzioni simili	5	5 – 15	15 – 20	15	20
3	Costruzioni che non ricadono nelle classi 1 e 2 e che non sono degne di essere tutelate	3	3 – 8	8 – 10	8	3 - 4

Documentazione:

A) Mappa Infrastruttura cloud e organizzazione dell'ambiente di test e di produzione:



B) Specifiche Sensori

27 **Posizionamento sensori: La catena sBB è dotata di un numero customizzabile di sensorboard, in base al numero di piani dell'edificio. Una sensorboard è sempre presente in fossa, mentre altre sensorboard sono disposte lungo le guide dell'ascensore, a diverse altezze. Una sensorboard può essere posizionata anche a bordo della cabina ascensore.**

Descrizione sensorboard: Ciascuna sensorboard è dotata di un buffer circolare, suddiviso in due slot che possono contenere i dati raw per una durata di circa 4 secondi. Quando uno slot è pieno, la sensorboard inizia a scrivere nell'altro slot, mentre i dati del primo slot vengono inviati al gateway. Questa procedura si ripete ciclicamente. La comunicazione con il gateway avviene tramite:

- WiFi o, da valutare, Powerline (sensore a bordo cabina) TO CHECK

- RS485 (sensori lungo guide e in fossa) TO CHECK

All'interno del circuito sarà incluso un convertitore AC-DC per permettere l'alimentazione diretta da 90 a 230V, 50 o 60 Hz.

28 **Specifiche tecniche:** **La sensorboard sarà dotata dei seguenti sensori – o altri con rapporto segnale disturbo adeguato ai servizi proposti.**

Accelerometro triassiale (ST ISM330DHCX)

Fondo scala: settabile fra ±2/±4/±8/±16 g (nostro setting: ±2 g)
Risoluzione: 16 bit (nostro setting: 0.06 mg)
Banda: 12-6667 Hz (nostro setting: 200 Hz)
Densità di rumore: 60 µg/√Hz

Giroscopio 3D (ST ISM330DHCX)

Fondo scala: settabile fra ±125/±250/±500/±1000/±2000/±4000 dps (nostro setting: ±125 dps)
Risoluzione: 16 bit (nostro setting: 3.8 mdps)
Banda: 12-6667 Hz (nostro setting: 200 Hz)
RMS rumore: 70 mdps

Termometro (ST ISM330DHCX)

Range: –40 °C / +125 °C (–20 °C / +90 °C è sufficiente)
Risoluzione: 0.5 °C

Igrometro (HTS221)

Range: [0, 100%]
Accuratezza: ±3.5 (fra 20% e 80%) e ±5% al di fuori

Magnetometro triassiale (ST IIS2MDC)

Range: ± 50 gauss

Sensitività: 1.5 mgauss
Banda: 10-100 Hz

Barometro (ST LPS22HH)

Range: 260 - 1260 hPa
Accuratezza: 0.5 hPa
Rumore (a bassa pressione): 0.65 Pa
Banda: 1 – 200 Hz

29 A.2.3. (RI) Definizione degli algoritmi di analisi descrittiva e cognitiva e dei relativi dati di input

Si utilizzeranno algoritmi come Data Mining, Graph Analytics, Deep Learning, oltre alle più classiche simulazioni scientifiche, per processare i dati in funzione delle quantità che si vogliono calcolare e, in particolare, delle domande alle quali si vuole rispondere nell'ultima fase. Gli algoritmi utilizzati in questa fase sono solo in parte classici, in quanto è necessario un discreto livello di personalizzazione in base al tipo di dato e tipo di analisi per rispondere alle esigenze specifiche del problema. Fra i numerosi aspetti di ricerca, uno dei più interessanti e unici offerto dal progetto riguarda l'opportunità di accoppiare classiche tecniche di Data Analytics, legate al largo volume di dati generato dal problema in esame, alle più classiche simulazioni numeriche ad elementi finiti: queste ultime possono essere viste sia come elemento di arricchimento, generando ulteriori dati ove non vi fosse la possibilità di installare i sensori, oppure come metodo diretto di analisi, per prevedere effetti e conseguenze nel presente e nel futuro a partire dalle misurazioni dei sensori stessi. Questo accoppiamento è un primo grande esempio verso una nuova era di Big Data Modeling & Simulation.

Attività Svolta:

1 Introduzione

Nella soluzione sBB, il movimento e le vibrazioni delle componenti dell'ascensore e dell'edificio sono misurati per tramite di sensori accelerometrici. Pertanto, la grandezza direttamente disponibile è l'**accelerazione**. Tuttavia, molte analisi richiedono di valutare piuttosto le **velocità** o, addirittura, gli **spostamenti** cui va incontro l'ascensore piuttosto che l'edificio. Ciò richiede di effettuare l'integrazione in tempo dei segnali in accelerazione al fine di ricostruire la velocità e, qualora si renda necessaria, l'integrazione in tempo dei segnali in velocità così ottenuti allo scopo di ricostruire lo spostamento (*displacement*).

2 Artefatti numerici

L'integrazione di segnali ottenuti da misure sperimentali è tuttavia un'operazione critica e che nasconde parecchie difficoltà. Senza opportuni accorgimenti, infatti, i risultati possono essere affetti da importanti artefatti numerici ed essere privi di significato fisico. Nel seguito si elencano le principali criticità a questo riguardo.

2.1 Mancata conoscenza della costante di integrazione

Integrando l'accelerazione in un dato intervallo temporale, si ottiene la variazione di velocità in quell'intervallo (il cosiddetto "delta"), piuttosto che il valore assoluto della velocità. Per ottenere quest'ultimo è infatti necessaria, in linea di principio, la conoscenza del suo valore iniziale (la **costante di integrazione**):

$$v(t) = v(t_0) + \int_{t_0}^{t} a(s)\ ds$$

Questo aspetto si rivela ancor più problematico quando il segnale in velocità deve essere integrato anch'esso per ottenere lo spostamento. Infatti, eventuali errori nella stima della costante di integrazione $v(t_0)$ si riflettono in **trend fittizi** nel segnale ricostruito dello spostamento (vedi punto successivo).

Quando si ricostruisce lo spostamento integrando il segnale in velocità, si ripropone il medesimo problema:

$$d(t) = d(t_0) + \int_{t_0}^{t} v(s)\ ds$$

Tuttavia, il problema di stimare la costante $d(t_0)$ è di secondaria importanza rispetto a quello di stimare $v(t_0)$. Infatti, la quantità di interesse non è tanto il valore assoluto dello spostamento $d(t)$, ma piuttosto la sua variazione $\Delta d(t)$, che è indipendente dal valore della costante di integrazione. In altre parole, variare $d(t_0)$ è equivalente a cambiare sistema di riferimento muovendo l'origine degli assi, operazione che non ha alcun effetto sull'interpretazione dei risultati.

La strategia per ovviare al problema della mancata conoscenza di $v(t_0)$ dipende dal tipo di condizioni nelle quali la misurazione è effettuata. Si considerano ad esempio le seguenti situazioni.

2.1.1 Misurazione di un corpo vibrante

Quando l'accelerazione si riferisce a un corpo che vibra, è ragionevole supporre che lo spostamento totale dall'inizio alla fine dell'intervallo di misurazione sia trascurabile in confronto all'ampiezza della vibrazione. Pertanto, siccome il delta dello spostamento è dato dall'integrale della velocità, è ragionevole supporre che la velocità avvia media nulla nell'intervallo di misurazione. Questa condizione definisce in modo implicito la costante di integrazione $v(t_0)$.

2.1.2 Misurazione di un corpo in movimento

Qualora l'accelerazione si riferisca a un corpo in movimento (quale la cabina di un ascensore), la strategia sopra riportata porterebbe chiaramente a conclusioni fallaci. Infatti, lo spostamento totale è in generale non nullo; al contrario, il suo valore rappresenta un output di interesse dell'analisi.

Tuttavia, se al momento in cui la misurazione ha inizio il corpo non è ancora in movimento, potrebbe essere lecito assumere che la velocità all'istante t_0 sia molto piccola in confronto alle velocità raggiunte durante la misurazione. Questo suggerisce di porre $v(t_0) = 0$.

Nel seguito si analizza l'errore introdotto da tale approssimazione. La velocità così stimata è data da:

$$\tilde{v}(t) = \int_{t_0}^{t} a(s)\ ds$$

L'errore relativo fra la velocità reale e quella stimata, rispetto alla norma infinito, è dato da:

$$\frac{|\tilde{v}(t) - v(t)|}{|v(t)|} = \frac{|v(t_0)|}{v_{max}}$$

Pertanto, se la velocità al tempo iniziale della misurazione è trascurabile rispetto alla velocità massima (velocità di picco), quest'ultima è stimata con una buona approssimazione.

Analizziamo ora l'errore commesso nella stima degli spostamenti. A tal scopo, definiamo:

$$\Delta d(t) = \int_{t_0}^{t} v(s)\ ds \quad \Delta\tilde{d}(t) = \int_{t_0}^{t} \tilde{v}(s)\ ds$$

Si ha che:

$$\Delta\tilde{d}(t) = \Delta d(t) - v(t_0)\,(t - t_0)$$

Pertanto, l'errore relativo nella valutazione dello spostamento è dato da:

$$\frac{\left|\Delta\tilde{d}(t) - \Delta d(t)\right|}{|\Delta d(t)|} = \frac{|v(t_0)|}{\left|\frac{\Delta d(t)}{t - t_0}\right|} = \frac{|v(t_0)|}{\left|\frac{1}{t - t_0}\int_{t_0}^{t} v(s)\ ds\right|}$$

L'errore che si commette nella valutazione dello spostamento dipende quindi dal rapporto fra la velocità al tempo iniziale e la velocità media che si ha fino al momento di interesse.

In conclusione, l'approssimazione di porre $v(t_0) = 0$ porta ad errori trascurabili quando vi è una ragionevole certezza nell'assumere che all'istante di inizio della misurazione la velocità sia piccola rispetto alle velocità raggiunte durante il moto.

Nel caso di interesse (il moto di una cabina di ascensore), tale approssimazione è lecita purché si soddisfino le seguenti condizioni:

- L'operazione di integrazione viene avviata in condizioni di cabina ferma. Per questa ragione si rende necessario la definizione di un criterio per controllare tale condizione (basato, ad esempio, sull'analisi dell'RMS dell'accelerazione).
- Le vibrazioni della cabina ferma non siano caratterizzate da **velocità istantanee** elevate. Infatti, un oggetto considerato *fermo* (nel senso di privo di deriva netta) può avere velocità istantanee (dovute a vibrazioni) non trascurabili. Qualora questa ipotesi non sia verificata, è opportuno sostituire l'ipotesi $v(t_0) = 0$ con l'ipotesi che la velocità abbia **media nulla in un breve periodo situato all'inizio della misurazione** (da t_0 a t_ε). A tal proposito si veda la sezione precedente. Questa strategia è più robusta della precedente, in quanto l'eventuale velocità istantanea non nulla non comporterebbe il propagarsi di errori.

Nota: tutte le valutazioni riportate in questa sezione, riferite all'ipotesi che il corpo sia fermo all'istante iniziale della misurazione, possono essere replicate analogamente sotto l'ipotesi che il corpo sia fermo all'istante finale, o in un qualunque istante intermedio.

2.2 Errori sulle componenti a bassa frequenza

Le analisi riportate nella sezione precedente sono effettuate sotto l'ipotesi di mancanza di errori di misura. L'oggetto di questa sezione è lo studio dell'impatto degli errori di misurazione sul segnale integrato, e delle strategie messe in atto per correggere gli artefatti così generati.

Quando si integra un segnale, una particolare attenzione deve essere adottata nei confronti degli errori associati alle frequenze più basse che, una volta integrati, porterebbero a significativi trend fittizi nel segnale ricostruito. Ad esempio, un errore sistematico nell'accelerazione dato da una costante si riflette in un trend lineare nella velocità. Questo rende necessario l'utilizzo di un **filtro passa alto**.

La normativa UNI 9916 a tal proposito raccomanda:

> *"L'uso di un filtro passa alto che permetta di eliminare il rumore (di natura elettrica o altro) della catena di misura […]. L'operazione di filtraggio è delicata e deve essere*

condotta adottando un tipo di filtro che non introduca esso errori e distorsioni; in particolare è importante che

a) *il filtro abbia una **frequenza di taglio congruente con il contenuto in frequenza del rumore** che si vuole eliminare*
b) *e soprattutto che abbia una **risposta in frequenza lineare per ciò che riguarda la fase**, visto che i criteri di valutazione degli effetti delle vibrazioni fanno riferimento ai valori di picco.*

*Un filtro che risponde appieno a tale esigenza è un **filtro di tipo FIR**. Possono essere comunque adottati altri tipi di filtri […]: in particolare, per mantenere piena confrontabilità con i risultati forniti da strumentazione oggi largamente utilizzata, si suggerisce il ricorso a **filtri IIR del tipo Butterworth di ordine 6**.*

*La **frequenza di taglio preferenziale è 1 Hz**; tuttavia in funzione del caso specifico, si possono adottare altre frequenza di taglio, scelta fra multipli e sottomultipli di 1 Hz con fattore 2.*"

(p. 12; grassetto nostro)

Si noti come questo aspetto (la rimozione delle componenti spurie a bassa frequenza) sia interlacciato con il precedente (la determinazione della costante di integrazione). Infatti, se la modalità utilizzata per determinare la costante di integrazione della velocità è quella di richiedere che il segnale in velocità abbia media nulla in un certo intervallo di tempo, la presenza di un trend fittizio dovuto all'integrazione di una componente spuria a bassa frequenza porterebbe a una valutazione errata.

3 Metodi di integrazione numerica

I metodi di integrazione numerica possono essere divisi in due classi:

- Metodi nel **dominio del tempo** (come metodo dei trapezi. Si noti che altri metodi non portano a risultati significativamente diversi.)
- Metodi nel **dominio delle frequenze** (noti in alcuni contesti come metodi di "Omega aritmetica")

La normativa UNI 9916 si esprime chiaramente in favore dei primi. Per essere *compliant* con la normativa infatti:

> *"È invece sempre proibita qualunque procedura d'integrazione che operi nel dominio delle frequenze"*
>
> *(p. 12)*

> *"A causa della natura sostanzialmente temporale dell'analisi richiesta per il confronto con i limiti, ai fini dell'applicazione della presente norma non possono essere applicati algoritmi di filtraggio ed integrazione nel dominio delle frequenze, tipicamente basati sulla trasformata di Fourier (DFT o FFT), a cause degli* ***errori di fase*** *che introducono e, quindi, della variazione sul valore istantaneo."*
>
> *(p. 25; grassetto nostro)*

Tuttavia, come si mostra in seguito, la nostra analisi effettuati su dati raccolti tramite sensori MEMS non ha riportato le criticità attribuite dalla UNI 9916 ai metodi operanti nel dominio delle frequenze.

4 Risultati su segnali raccolti con tavola vibrante

Al fine di confrontare e validare le diverse tecniche di filtraggio e integrazione, ovvero di calibrare i valori ottimali dei parametri quali la frequenza di taglio (*cutoff*), in questa sezione si riportano i risultati ottenuti applicando tali tecniche a misure sperimentali.

I dati sperimentali utilizzati in questa analisi sono stati raccolti presso il laboratorio di prove materiali dell'Università degli studi di Bergamo in data 03/03/2020, tramite tavola vibrante. L'obiettivo è di caratterizzare le performance degli algoritmi in condizioni sia di vibrazioni ambientali che di input sismico. Durante i test sono posti a confronto quattro diverse tipologie di sensori:

- Sensore piezoelettrico
- Sensore MEMS Unibg
- Sensore MEMS prima generazione sBB (d'ora in poi: sBB)
- Sensore MEMS in dotazione STWIN (d'ora in poi: ST)

I sensori MEMS sono stati posizionati sia sulla tavola che su un modello a un grado di libertà, mentre il sensore piezoelettrico solo sulla tavola (vedi Fig. 1). Pertanto, in questo documento si analizzano soltanto i segnali raccolti su tavola.

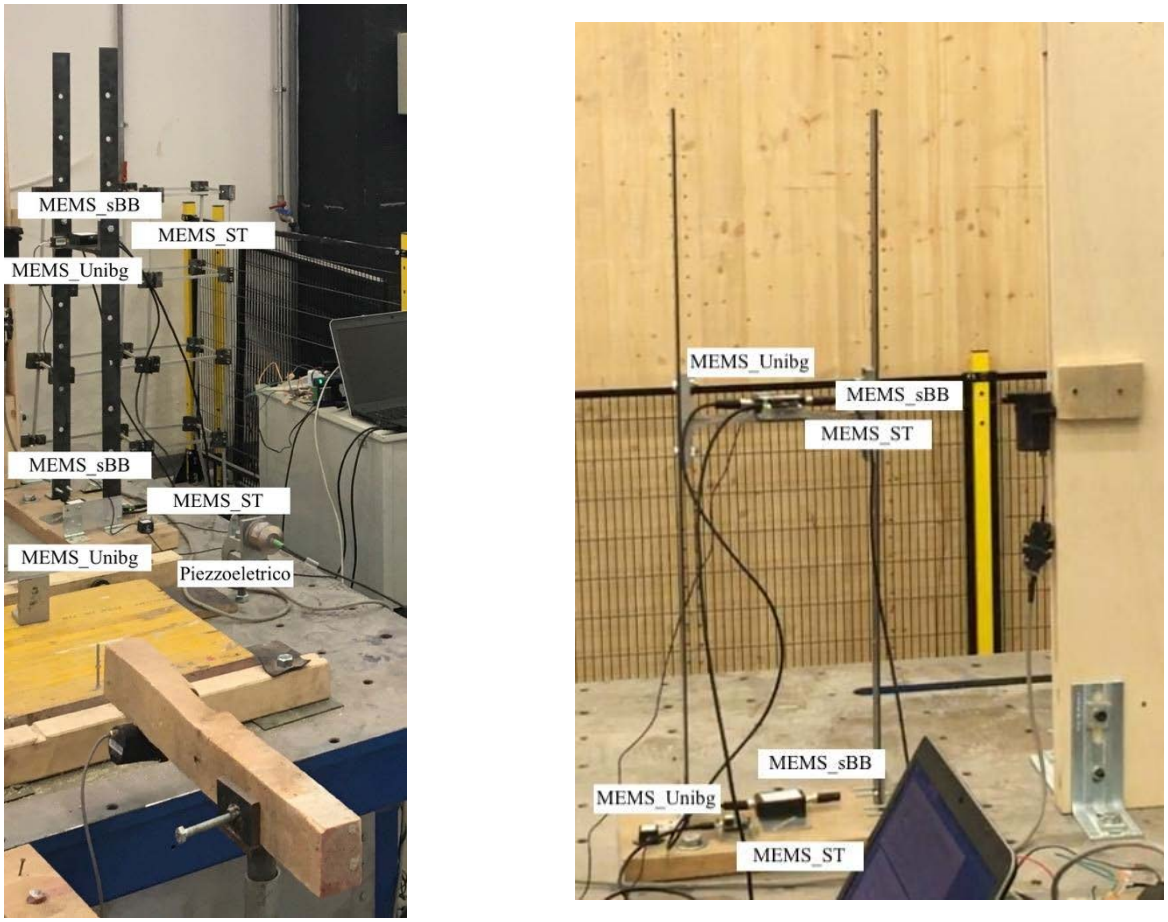


Figura 1 *Setup di prova (fonte: Report prove su tavola v3.0 – Unibg)*

È inoltre disponibile una misurazione dello spostamento della tavola. Tale misurazione è utilizzata come strumento di validazione dei diversi algoritmi considerati, nonché delle misurazioni ottenute con i diversi sensori. Nello specifico, si valuta in quale misura il segnale in accelerazione doppiamente integrato permette di ricostruire lo spostamento misurato.

La frequenza di acquisizione per tutti i sensori è di 200 Hz.

4.1 Trattamento dei dati

In questa analisi si parte dai dati in formato *raw*. I dati vengono successivamente allineate mediante massimizzazione della cross-correlation (vedi anche: *Report prove su tavola v3.0 – Unibg*).

Ogni successiva elaborazione è descritta caso per caso.

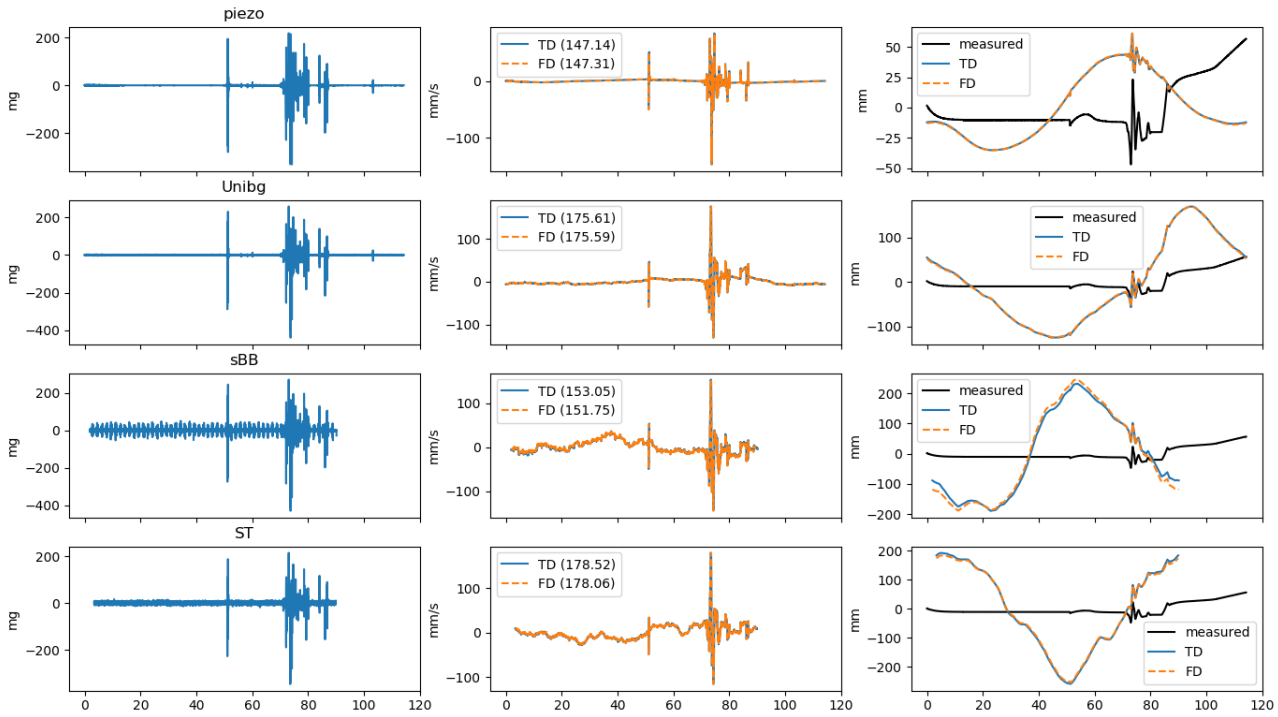
4.2 Risultati in assenza di filtraggio

Come prima elaborazione effettuiamo i seguenti passaggi:

1. Rimozione del valor medio all'accelerazione. Questa operazione rappresenta un primo "filtro passa alto" (in quanto rimuove la frequenza zero) e serve a rimuovere trend dal segnale in velocità dovuti a errori sistematici nella misurazione dell'accelerazione (oltre alla costante data dall'accelerazione di gravità).
2. Analisi nel dominio del tempo (TD):
    - Integrazione tramite metodo dei trapezi dell'accelerazione.
    - Rimozione del valor medio alla velocità (assumendo l'ipotesi che la velocità abbia media nulla; vedi sezione "Misurazione di un corpo vibrante").
    - Integrazione tramite metodo dei trapezi della velocità.
    - Rimozione del valor medio allo spostamento (unicamente per fini di visualizzazione).
3. Analisi nel dominio delle frequenze (FD):
    - Integrazione dell'accelerazione nel dominio delle frequenze (i.e. divisione per $i\omega$) per ottenere la velocità.
    - Doppia-integrazione dell'accelerazione del dominio delle frequenze (i.e. divisione per $-\omega^2$) per ottenere lo spostamento.

Nella figura sotto si riportano i risultati ottenuti.

- Nella prima colonna è mostrata l'accelerazione. Si noti come il MEMS sBB durante la prova ha riportato un disturbo periodico. È probabile che un simile disturbo sia legato allo specifico esemplare (altri sensori analoghi non hanno riportato simili disturbi). Ciononostante, si è scelto di non eliminare tale disturbo, allo scopo di valutare la robustezza degli algoritmi qui considerati rispetto a simili fenomeni.
- Nella seconda colonna è mostrata la velocità ricostruita. Nella legenda, fra parentesi tonde è indicata il massimo del valore assoluto della velocità (*peak component particle velocity*, *pcpv*), un indicatore utilizzato per alcuni test, fra cui quelli definiti dalla UNI 9916.
- Nell'ultima colonna è mostrato lo spostamento ricostruito. La linea nera riporta lo spostamento misurato, mentre quelle colorate lo spostamento ricostruito. Si noti come la quantità di interesse non è tanto il valor assoluto dello spostamento, ma le sue variazioni (vedi sopra).

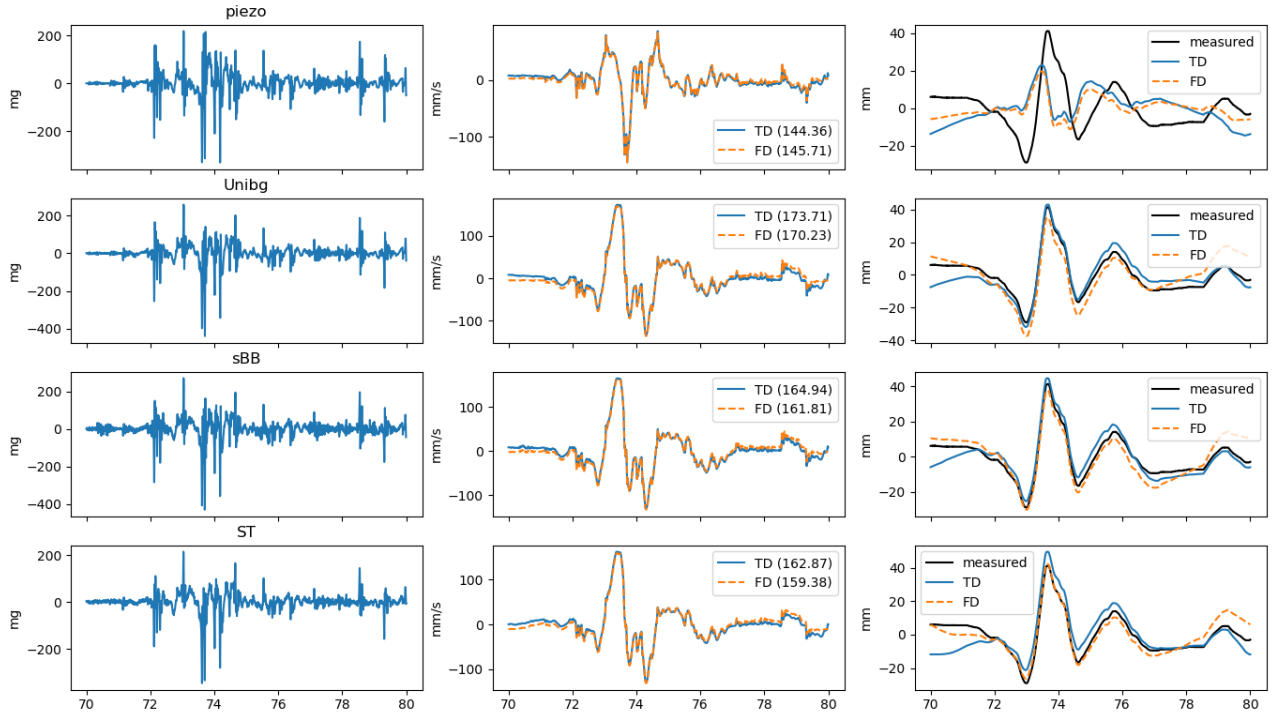


Osservazioni:

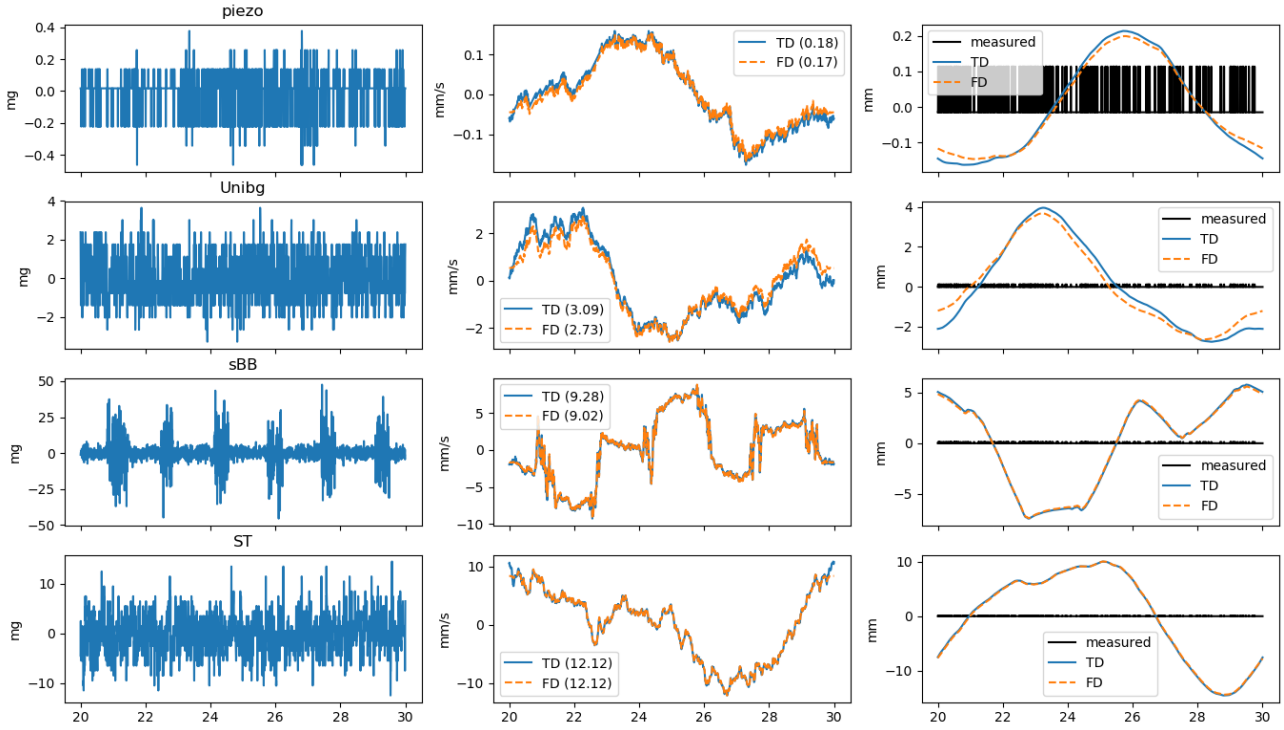
- Le due tecniche TD e FD sono sostanzialmente in accordo nei risultati. Si discostano leggermente per il sensore sBB (probabilmente a causa dei disturbi), ma in misura molto limitata.
- Lo spostamento ricostruito in questo modo riporta degli evidenti artefatti, legati a componenti spurie a bassa frequenza. Ciò è confermato dal fatto che tali artefatti sono riproducibili fra FD e TD, mentre sono differenti nei quattro sensori utilizzati. Si noti, d'altra parte, che nei momenti di maggior scuotimento (da 70 a 80 secondi circa) lo spostamento "corretto" viene ricostruito correttamente, anche se sovrapposto alle oscillazioni fittizie dovute ai modi spuri a bassa frequenza.

Allo scopo di valutare l'effetto dell'ampiezza dell'intervallo di integrazione sul risultato ottenuto, nelle seguenti due immagini mostriamo il risultato della medesima analisi di cui sopra restringendo la finestra temporale:

- all'intervallo 70 s – 80 s (il momento di maggior scuotimento)



- all'intervallo 20 s – 30 s (durante il quale la tavola vibrante è a riposo, quindi è presente solo il rumore ambientale).



Osservazioni:

- Quando le vibrazioni sono elevate (alto rapporto segnale-rumore), lo spostamento ricostruito riporta un ottimo match con quello misurato per tutti e tre i MEMS e per entrambi gli approcci (TD, FD). Questo porta a concludere che anche le velocità siano state ricostruite con un buon livello di fedeltà (la pcpv è coerente).
- D'altro canto, lo spostamento ricostruito utilizzando il dato dell'acceleratore piezoelettrico risulta significativamente diverso (vi è uno sfasamento in fase di circa mezzo periodo e anche un mismatch sulla massima ampiezza). Il fatto che il risultato ottenuto in TD e FD sia pressoché coerente, sembrerebbe suggerire un **vantaggio dei sensori MEMS rispetto al piezoelettrico**.
- Nel caso di solo rumore ambientale, il rumore di fondo porta ad avere artefatti nella ricostruzione dello spostamento. Tali artefatti sono trascurabili per il sensore piezo (ordine di 0.1 mm), mentre meno trascurabili per i MEMS (5 mm per Unibg e sBB, 10 mm per ST, il quale infatti è il sensore che riporta un maggior rumore di fondo). La pcpv stimata dal piezoelettrico è di circa 0.2 mm/s, mentre quella ricostruita dai MEMS è di circa 2 mm/s (Unibg) o 10 mm/s (sBB e ST). Siccome le velocità di interesse che si intendono rilevare sono di ordine comparabile (per certe classi di edifici, secondo la UNI 9916, il limite può essere addirittura inferiore a 5 mm/s), **la correzione di tali artefatti si rende necessaria**.

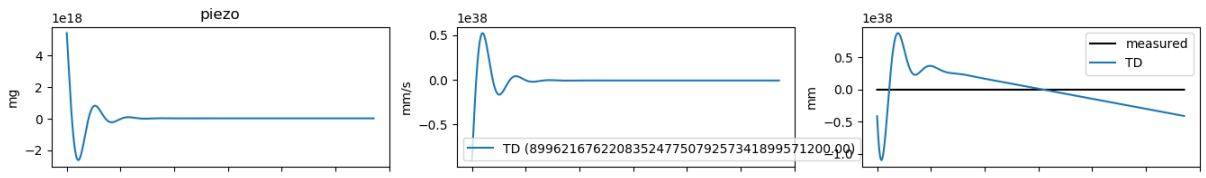
Nelle successive sezioni vengono descritte alcune tecniche di filtraggio e integrazione nel dominio del tempo e delle frequenze, che verranno poi messe a confronto.

29.1

4.3 Filtraggio e integrazione nel dominio del tempo

Consideriamo innanzitutto un filtro di tipo Butterworth di ordine 6, in due diverse implementazioni: implementazione "base" e implementazione come serie di sezioni del second'ordine (*series of second order sections, sos*).

Le due implementazioni portano a risultati sostanzialmente equivalenti, tranne che per basse frequenze di *cutoff*, per le quali l'implementazione base riporta seri problemi instabilità numeriche, come riportato nella figura seguente (cutoff = 0.1 Hz). Pertanto, nel seguito si considererà unicamente l'implementazione "sos".



Nei risultati mostrati in seguito vengono filtrati sia il segnale in accelerazione che quello in velocità (prima di essere integrati), in quanto integrando soltanto uno dei due gli artefatti sopra riportati permangono.

4.4 Filtraggio e integrazione nel dominio delle frequenze

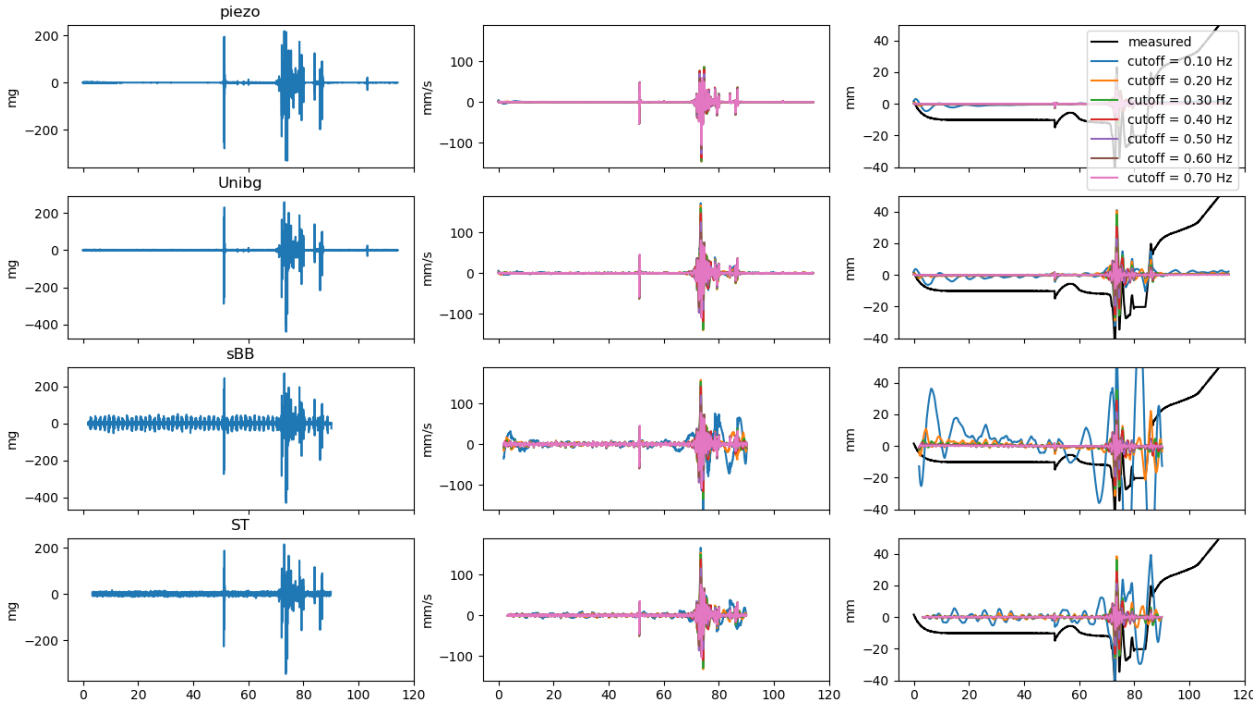
Si considera ora la seguente tecnica di filtraggio, coniugata con l'algoritmo di integrazione nel dominio delle frequenze. Dopo aver trasformato il segnale tramite algoritmo FFT e integrato tramite opportuna divisione (per $i\omega$ o $-\omega^2$, in caso di coppia integrazione) a esso viene applicata una maschera, che taglia i modi associati a frequenza inferiori alla frequenza di cutoff. Infine, il segnale in tempo viene ricostruito tramite FFT inversa.

4.5 Confronto fra le tecniche

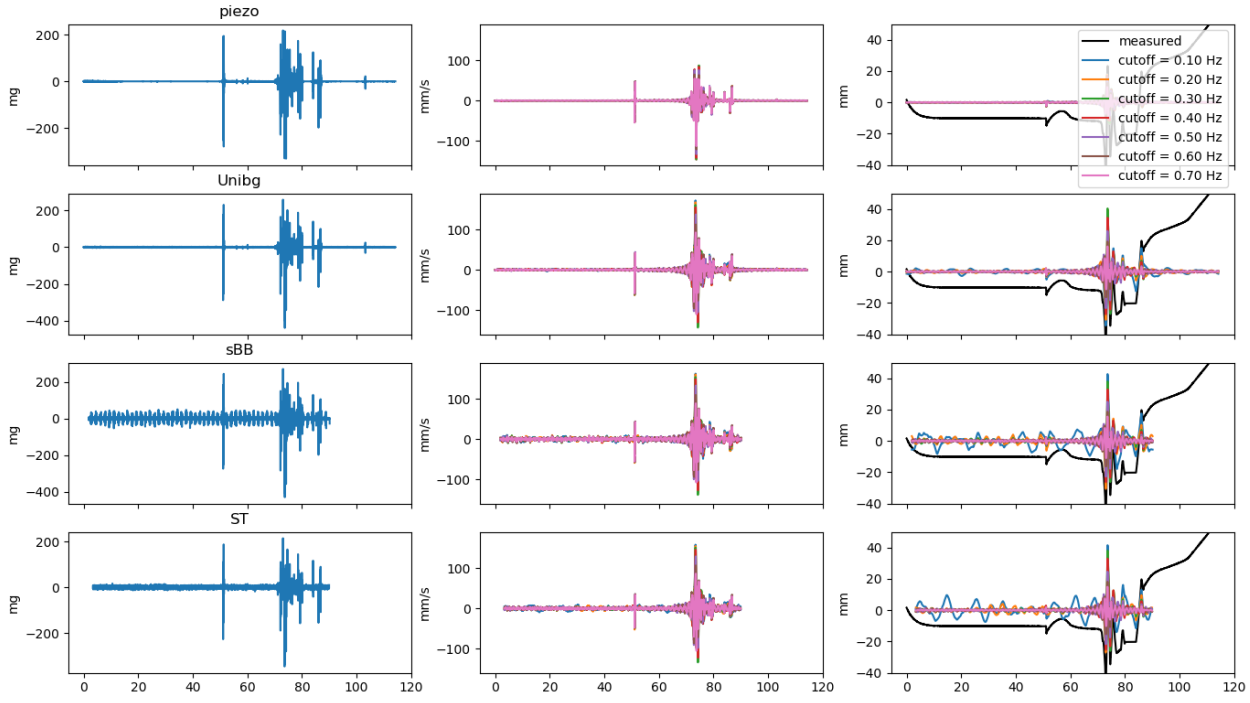
Nel seguito si pongono a confronto i risultati ottenuti con le due tecniche, per diversi valori di frequenza di cutoff, sulle tre finestre temporali considerate precedentemente.

4.5.1 Finestra temporale completa

Dominio del tempo (TD)

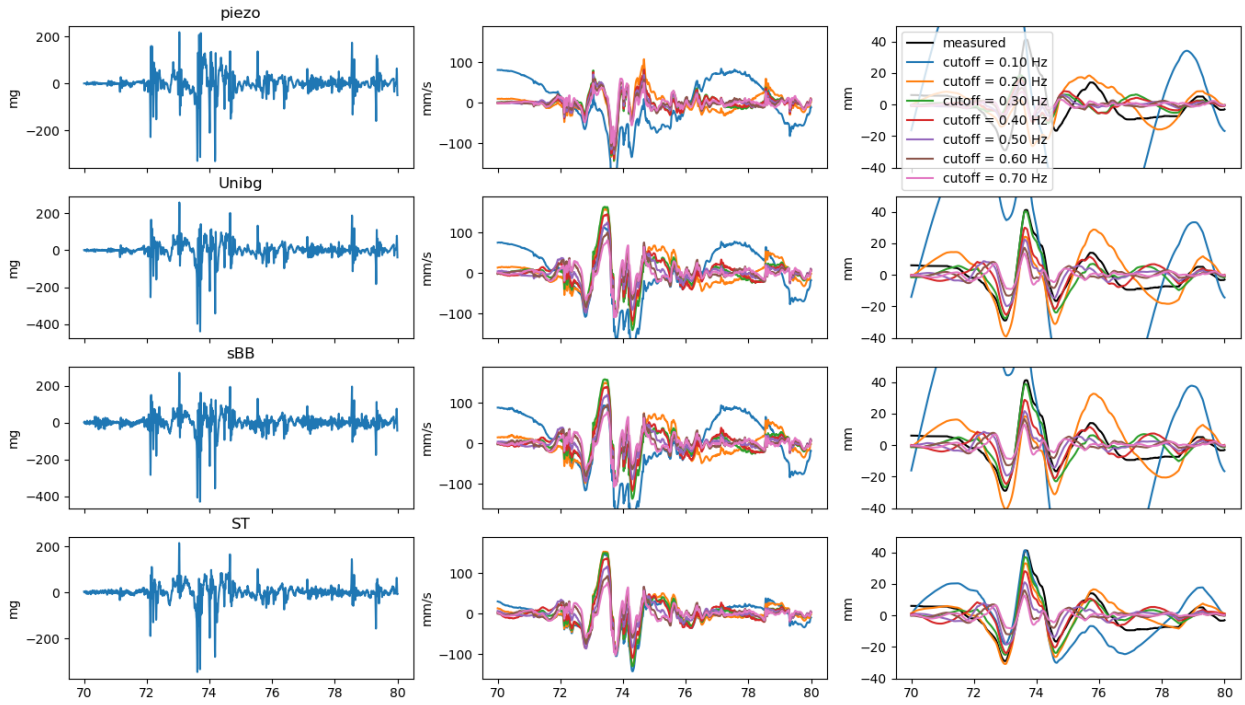


Dominio delle frequenze (FD)

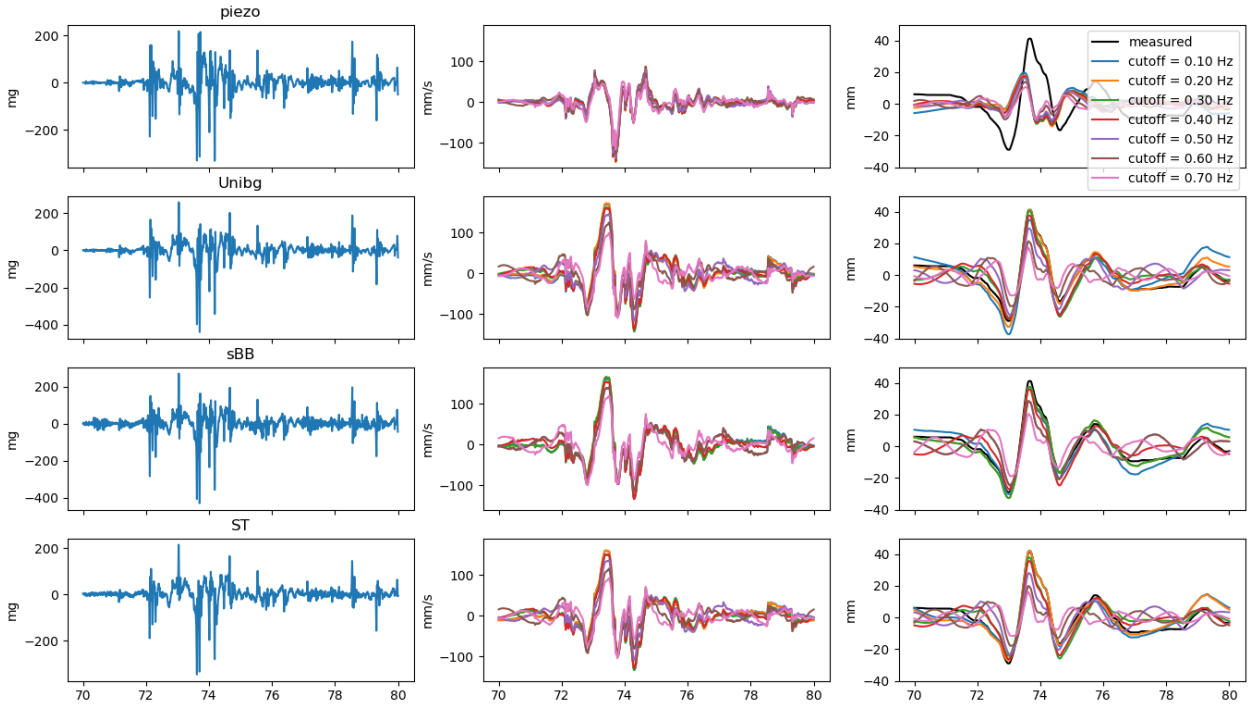


4.5.2 Intervallo di maggior scuotimento (70 s – 80 s)

Dominio del tempo (TD)

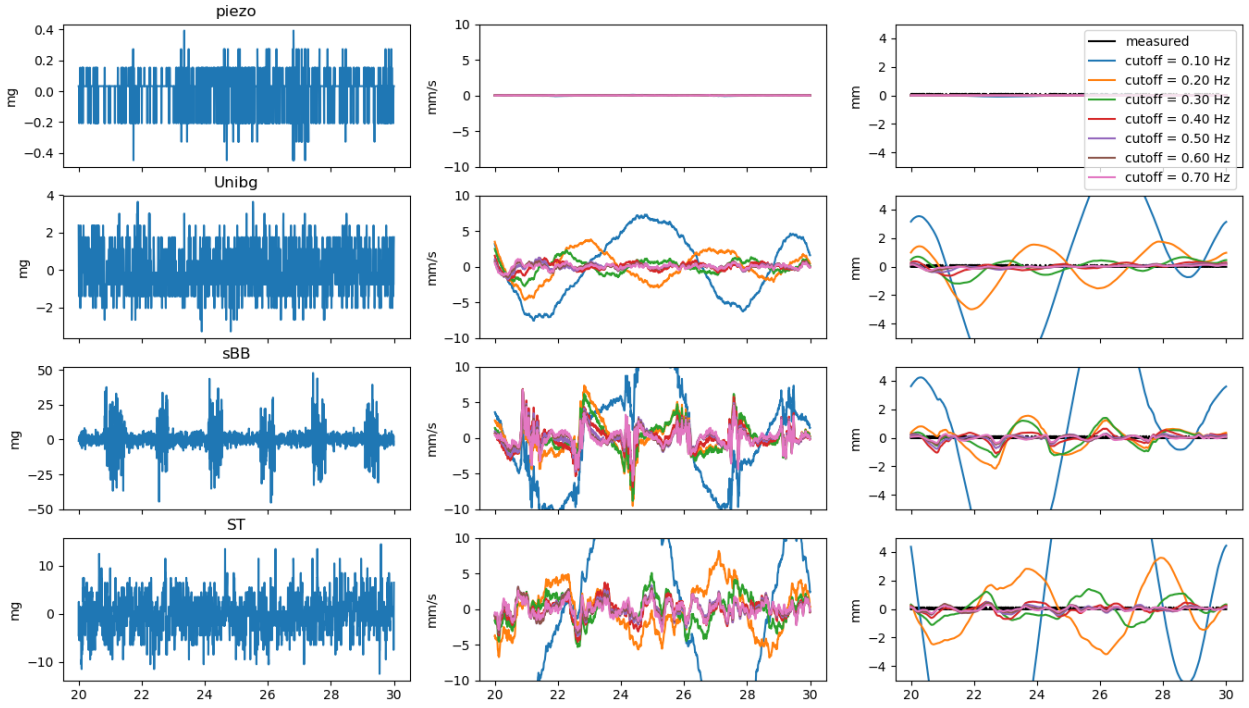


Dominio delle frequenze (FD)

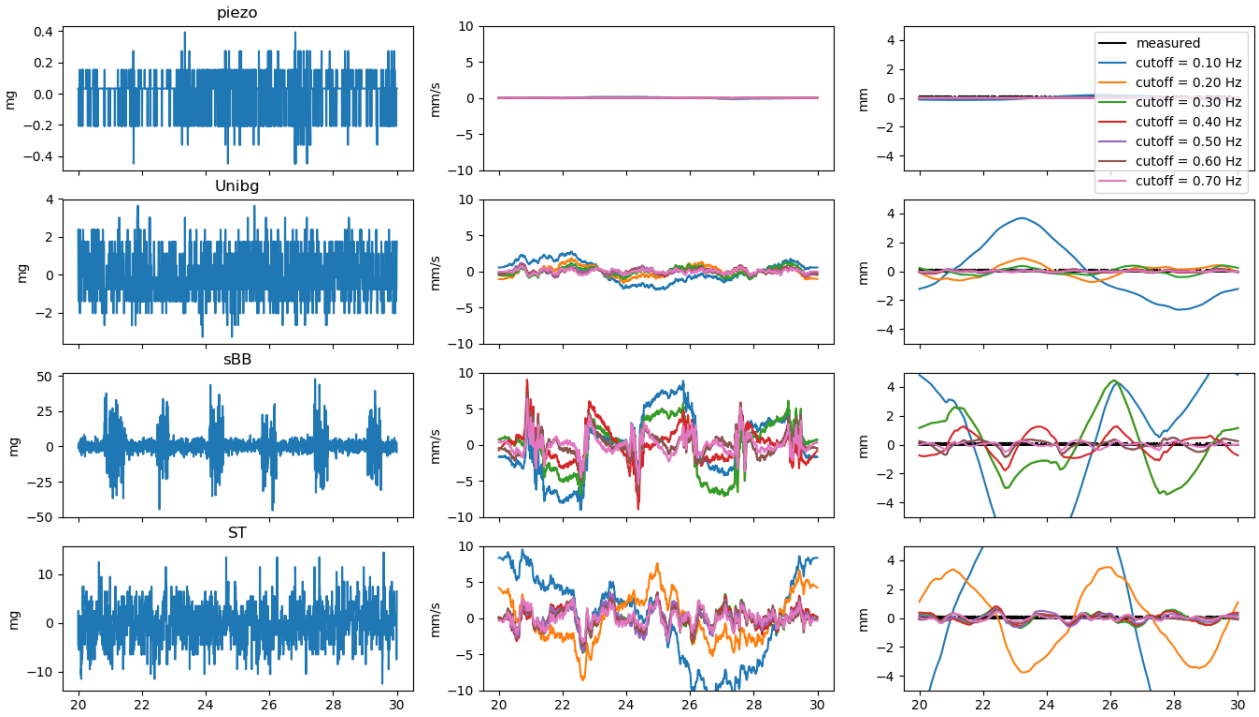


4.5.3 Intervallo con solo rumore ambientale (20 s – 30 s)

Dominio del tempo (TD)



Dominio delle frequenze (FD)



4.5.4 Commenti

In generale, entrambi metodi mostrano sensibilità alla frequenza di cutoff, sia per il segnale in velocità ricostruito che per quello in spostamento. Il valore ottimale che bilancia il *trade-off* fra i due target contrastanti, ossia la capacità

- da una parte, di cogliere l'ampiezza delle oscillazioni nell'intervallo di massimo scuotimento
- dall'altra, di eliminare oscillazioni fittizie durante l'intervallo in cui la tavola vibrante è ferma

risulta essere attorno a 0.4 Hz.

Si noti come l'approccio FD sia **più stabile** rispetto al valore del cutoff rispetto all'approccio TD. In altre parole, una piccola variazione del *cutoff* porta a una variazione più sensibile nei risultati nel caso TD che nel caso FD. Ciò rappresenta un aspetto di fondamentale importanza, in quanto una soluzione cercata non dovrebbe richiedere *tuning* manuale di alcun parametro.

In conclusione, nonostante le indicazioni contenute nella UNI 9916, l'approccio che si dimostra più promettente è quello di **filtraggio e integrazione nel dominio delle frequenze**, con cutoff di circa 0.4 Hz. Infatti, tale approccio porta ai seguenti risultati:

- Spostamento ricostruito con buona precisione.
- Significativa riduzione del rumore ambientale, sia nella ricostruzione della velocità che dello spostamento.
- Lo spostamento ricostruito è poco sensibile al *cutoff*, in un intervallo prossimo a 0.4 Hz.
- La pcpv ricostruita è poco sensibile al *cutoff*, in un intervallo prossimo a 0.4 Hz.

5 Conclusioni

Il presente studio ha evidenziato le seguenti principali conclusioni:

1. I sensori MEMS permettono di ricostruire lo spostamento (e quindi, presumibilmente, le velocità, anche se non si ha riscontro su questo, in assenza di misure dirette) meglio del sensore piezoelettrico. Ciò potrebbe essere alla bassa risposta del piezoelettrico alle basse frequenze.
2. Il rapporto segnale-rumore dei MEMS è d'altra parte tale da rendere necessario un filtraggio del segnale per ricostruire correttamente velocità e spostamento (l'errore ottenuto con il segnale raw è dello stesso ordine delle soglie date, in certe condizioni, dalla UNI 9916).
3. Un confronto fra tecniche di integrazione e filtraggio nel dominio del tempo e nel dominio delle frequenze ha evidenziato dei vantaggi per un algoritmo appartenente alla seconda categoria (contrariamente alla criticità espresse nella UNI 9916, che non sono emerse nella nostra implementazione).
4. Applicando tale algoritmo siamo stati in grado di:
    a. ricostruire lo spostamento misurato sperimentalmente con buona precisione;
    b. ridurre significativamente il rumore ambientale (di oltre un fattore 10);

c. ottenere risultati relativamente stabili rispetto alla scelta della frequenza di cutoff.

Documentazione:

Output dei servizi. Elenco algoritmi sviluppati:

BUILDING'S SERVICES LIST

- BUILDING INTEGRITY (BI)
- BUILDING HEALTH (BH)
- BUILDING DRIFT (BD)
- BASEMENT ACCELERATION (BA)
- PEOPLE DISTURBANCE (PD)
- BUILDING LOGBOOK (BL)

LIFT'S SERVICES LIST

- DOORS MONITORING (DM)
- FLOOR LEVELING (FL)
- LIFT USAGE (LU)
- USER EXPERIENCE (UX)
- POWER OUTAGES (PO)
- LIFT LOGBOOK (LL)

30 A.2.4. (RI) Definizione dei modelli dati e degli algoritmi adattativi

L'ultima fase consiste nell'adattare la risposta algoritmica prodotta dai modelli e trasformarla in un'informazione utile per l'utilizzatore finale, come ad esempio suggerire un'ispezione in una determinata sezione di un viadotto in base a stime di carico che indicano possibili danni alla struttura, non ancora gravi ma potenzialmente in fase peggiorativa.

Attività Svolta:

Individuazione lista algoritmi e sensori utilizzati

BUILDING'S SERVICES LIST	
	BUILDING INTEGRITY (BI): The triaxial accelerometers of the **SAFETY BLUE BOX** sensors detect vibrations that can **induce cosmetic damage** to the building such as cracks, fissures, plaster detachment. In this way the **non-structural damage** threshold is monitored on the basis of the references given by **UNI9916:2014** and **ISO4866:2010**. The **peak velocity** along each direction is reconstructed starting from the accelerometric signal and the associated frequencies carrying the highest energy content are found by **spectral analysis**. The combined knowledge of these values provides an indication on the hazard level of architectural damage to the building. In order to filter out potential disturbances due to the lift movements or to the doors opening/closing, the algorithm works in conjunction with a classifier, that allows to select time windows of idle lift, on the basis of the vibrations measured through the cabin sensor. Then, the algorithm aggregates the results on a 10 minutes basis, accounting for the potentially differences in the amount of time of idle lift.

	STATUS: developed and validated. **PATENTABLE:** NO, as code will translate the Norm (delta would be on continuous monitoring vs on spot) **THRESHOLD:** these are defined in the standard for different building (workout a specific table). Any other sensor to placed along the shaft? **SENSORS:** accelerometers (non synchronized)
	**BUILDING HEALTH (BH): SAFETY BLUE BOX** checks the **structural response** of a building to internal or external events. Detects the anomalies of a typical behavior of the single building in its own environment of reference in case of explosions, seismic swarms or building works in the vicinity, constantly monitoring the stability conditions thereof. This algorithm performs a **dynamical identification** of the building characteristic response, in terms of proper frequencies of vibration and damping properties. After a substantial event, sBB compares the new characteristic response of the building with the one recorded before the event. Deviations from a previous state can be indicative of building damage, so facilitating the level of intervention. Moreover, this algorithm also monitors possible gradual deviations of building characteristic response, thus detecting a potential degradation of the building stability. **STATUS:** R&D phase – forecasted release spring 2021 (UNIBG dependant) **PATENTABLE:** YES (to be checked w/PRiva based on UNIBG developments) **THRESHOLD:** to be checked w/PRiva based on UNIBG developments **SENSORS:** accelerometers (must be synchronized ??? Better via RS485)
	**BUILDING DRIFT (BD): SAFETY BLUE BOX** computes the maximum and residual inclination of a building generated by external agents inducing oscillations. By analyzing the data received from sensors is possible to assess a risks associated to a potential structural failure. Specifically, by comparing the pitch and roll angles measured at different floors, it is possible to detect the overall **residual inclination** of a building and the **reciprocal drift** between floors. **STATUS:** developed and validated. **PATENTABLE:** NO, as we are measuring angles (to be included with other patents) **THRESHOLD:** value defined by PRiva

	SENSORS: accelerometers (non synchronized) **SENSORS that could be used:** gyroscopes (for dynamical analysis)
	BASEMENT ACCELERATION (BA): SAFETY BLUE BOX monitors the effects of earthquakes and nearby natural or human induced vibrations to the building basement. Motions that typically propagate in the ground. By monitoring the energy content associated with the vibrations measured at the building basement, it is possible to monitor repetitive or exceptional **sources of vibrations**. **STATUS:** developed and validated. **PATENTABLE:** NO, innovation is to have the sensor in the pit for any kind of vibration measure. Valuable for capillary measurement vs discrete (to be included with other patents) **THRESHOLD:** legare ai valori definiti per le rilevazioni sismiche **SENSORS:** accelerometer at the basement
	PEOPLE DISTURBANCE (PD): SAFETY BLUE BOX measures vibrations in the three axial dimensions, even those produced by the human activity (operation of industrial machines, road and traffic in general, building sites activity) and is used as "indicator for potential disturbance to people". By measuring the intensity and frequency of sources of noise, is possible to check deviations of set thresholds with reference to **UNI9614: 2017** and **ISO2631/1:1997**. **STATUS:** R&D phase – development time XXXXXXXXXX TBD **PATENTABLE:** TBD **THRESHOLD:** rif. 55dB average **SENSORS:** accelerometer on the cabin roof + microphone?

BUILDING'S ADMINISTRATIVE SERVICE	
	BUILDING LOGBOOK (BL): SAFETY BLUE BOX can provide periodic reports or real time notifications available on PC and/or smartphones. The data supplied provide an objective evidence of the presence of elements of disturbance or risk together with valuable information which can be used to perform building maintenance works. Tracks the history of the building's structural response when subject to ordinary strains or extraordinary stresses. It is a real black box of the building which can also facilitate and accelerate the verification of post-event building usability.

	STATUS: Under development – check with TEKNET

LIFT'S SERVICES LIST	
	**GUIDE RAILS MONITORING (GM):** This service provides real-time information on the status of the lift guide rails fixings or deformations using sensors mounted on the guide rails themselves. This Service can also be regarded as a mean to indicate **LIFT SAFE FOR USE**. By using the spectral analysis of the accelerometric signals, the algorithm identifies the **signature characteristic** of a specific guide rail. By comparing signatures over time, **SAFETY BLUE BOX** identifies gradual or abrupt deviations, which are symptoms of potential failure. First, during a training phase, the natural variability of the spectrum is identified. Then, the measured spectra are compared with the mean spectrum measured during the training phase, and a statistical hypothesis test is performed to provide a p-value to the hypothesis that a deviation has occurred. **STATUS:** R&D phase – development time 4+ month **PATENTABLE:** YES, mathematical analysis – literature to be checked **THRESHOLD:** comparative analysis over time and measure of deviation **SENSORS:** accelerometers (non synchronized)
	**DOORS MONITORING (DM):** This service provides information on doors malfunctions while these are opening and closing, via measurement using sensors mounted on the cabin. This service can measure the presence of people on the car roof. This algorithm measures the vibrations during the doors opening and closing lifting an alert whenever **anomalous vibrations** are detected. **STATUS:** R&D phase – development time 4+ month **PATENTABLE:** YES, Wittur offer a solution based on a comparison between Current absorption and Encoder retroaction analysis **THRESHOLD:** comparative analysis over time and measure of deviation **SENSORS:** accelerometer on the cabin

	**FLOOR LEVELING (FL):** This service provides information on the misalignment between the car and the floor threshold value (> 10 mm) via measurement using sensors mounted on the cabin (leveling and re-leveling accuracy). This service provides an approximate car position in the shaft and car movement sudden interruptions. **STATUS:** R&D phase – development time 1 month (once technology is defined) **PATENTABLE:** NO (nel brevetto Sacertis con magnetometro, da includere in altro deposito (?)) **THRESHOLD:** measures deviation over time +/- 10 mm **SENSORS:** magnetometers, Hall effect sensors, optical sensors (**to be defined**)
	**LIFT USAGE (LU):** Provides a first-level statistical analysis of Lift Travel Data, Measures the number of trips made, provides the count of major destinations, the mean travelling time, measures the capacity utilization over time, measure the total trips distance on a monthly basis. Provides an estimation of the usage of the lift system for re-planning of basic maintenance services (e.g. cleaning services). This service supports people traffic management analysis. **STATUS:** R&D phase – development time 4 month (we can start with partial service) **PATENTABLE:** YES (?) se non viene fatto da quadro di manovra **THRESHOLD:** not applicable **SENSORS:** magnetometers, Hall effect sensors, optical sensors, accelerometers, barometers, … (to be defined)
	**USER EXPERIENCE (UX):** with the **SAFETY BLUE BOX**, by merging data resulting from different sensor's measurements it is possible to evaluate the exceeding of a set threshold, in accordance with defined set quality of service. Level of user comfort during the travel experience is measured. Provides a measure of the quality of motion (accelerations and stops). **STATUS:** R&D phase – development time 3 month (we can start with partial service) NOTE: Linked to PD development **PATENTABLE:** YES (?) – continuous measure vs on spot **THRESHOLD:** sum-up of different service readings to define the quality of trips **SENSORS:** accelerometer + microphone (?) on the cabin.
	**POWER OUTAGES (PO):** this service measures the number of time power has gone OFF during a period of time.

	STATUS: TBD **PATENTABLE:** NO, can be obtained with other means **THRESHOLD:** ON/OFF **SENSORS:**

LIFT'S ADMINISTRATIVE SERVICE	
	**LIFT LOGBOOK (LL): SAFETY BLUE BOX** can provide periodic reports or real time notifications available on PC and/or smartphones. The data supplied provide an objective evidence of the presence of elements of disturbance or risk together with valuable information which can be used to perform extraordinary maintenance works. Tracks the history of the Lift performance response when subject to ordinary strains or extraordinary stresses. It is a real black box of the building which can also facilitate and accelerate the verification of post-event Lift usability. **STATUS:** Under development – check with TEKNET

Documentazione:

Vedi File “presentazione dei risultati.pptx” cartella “documentazione IOTSAFE”

Benefici della proposta IOTSAFE

- Gestire (mitigare) il rischio
- Aumentare la sicurezza delle persone
- Disporre di dati indipendenti utili ad esempio per contenziosi e analisi per confronto prima e dopo un evento
- Essere in grado di fare le scelte giuste al momento giusto
- Ottimizzare i costi operativi e straordinari.
- Gestire meglio situazioni critiche
- Tutelare il valore nel tempo degli edifici

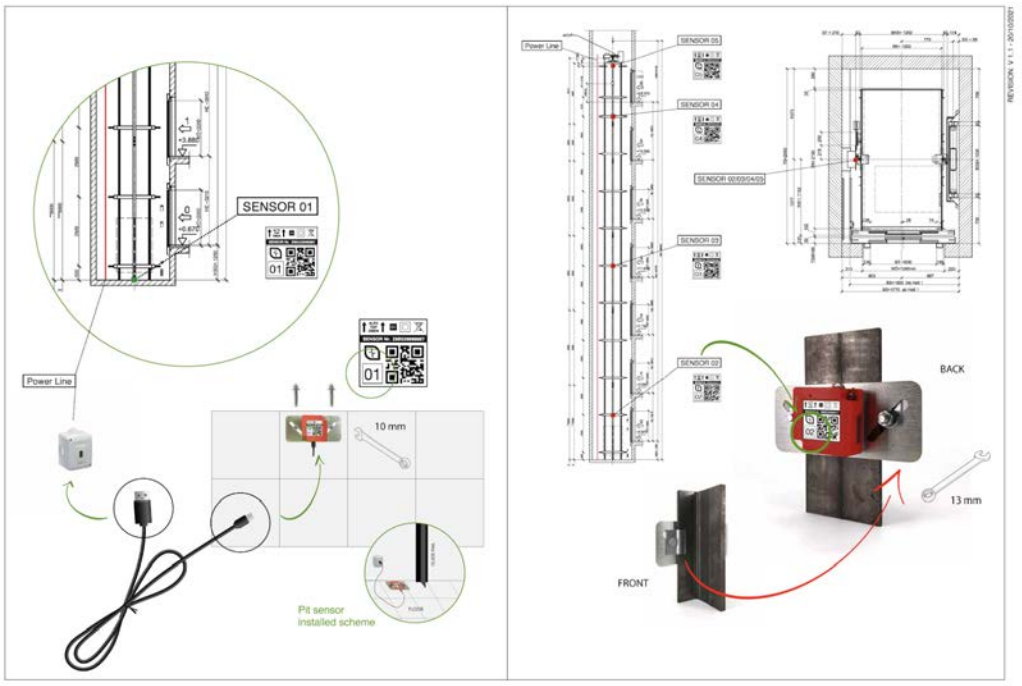
Rischi degli edifici che possono essere monitorati dalla soluzione IOTSAFE

Invecchiamento

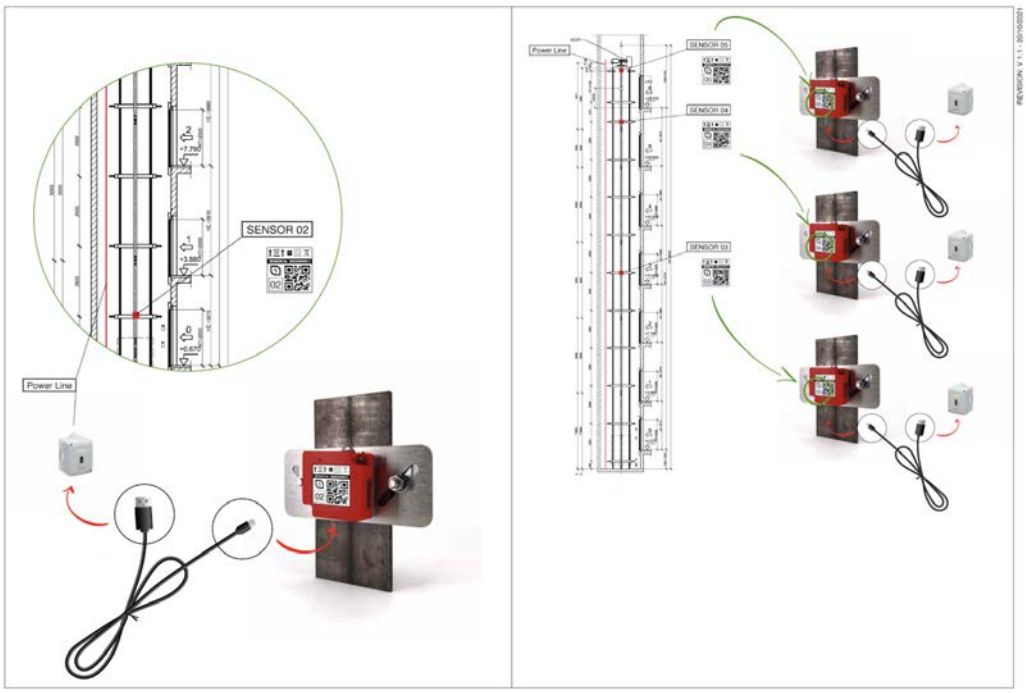
Eventi naturali

Azioni umane

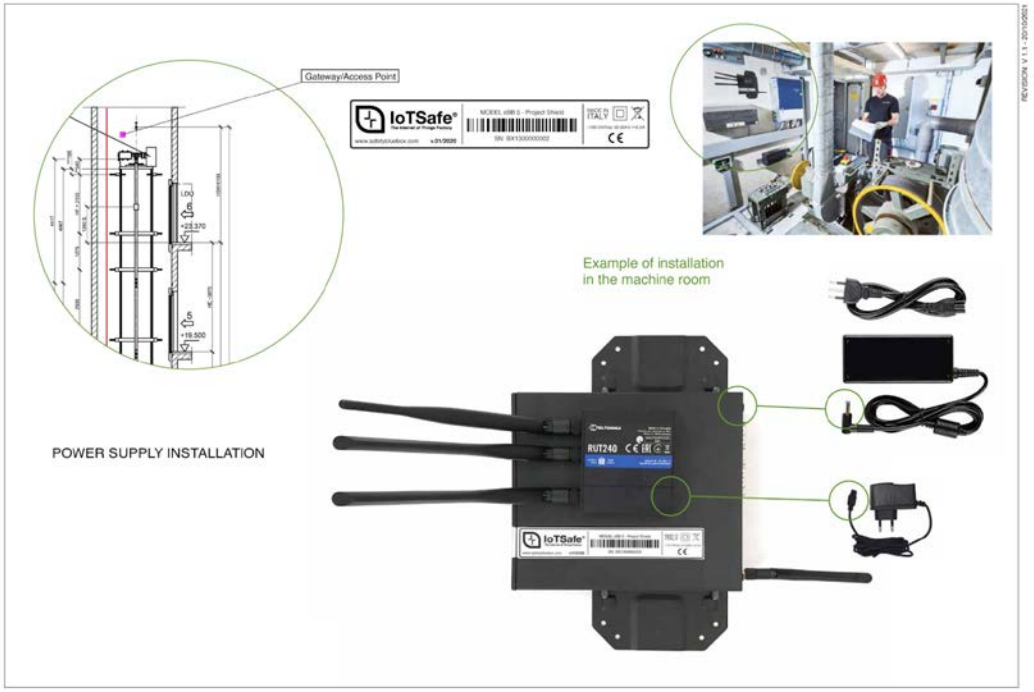
+ Installazione sensore di fossa e sulle guide



+ Alimentazione dei sensori



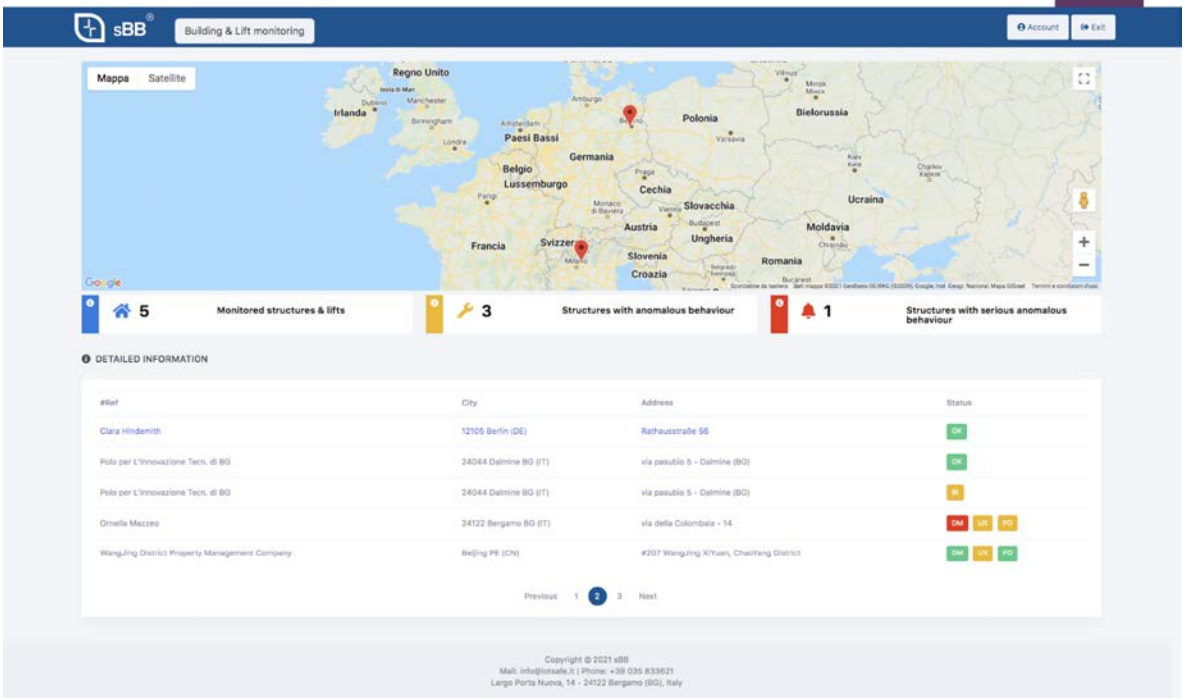
Installazione del gateway



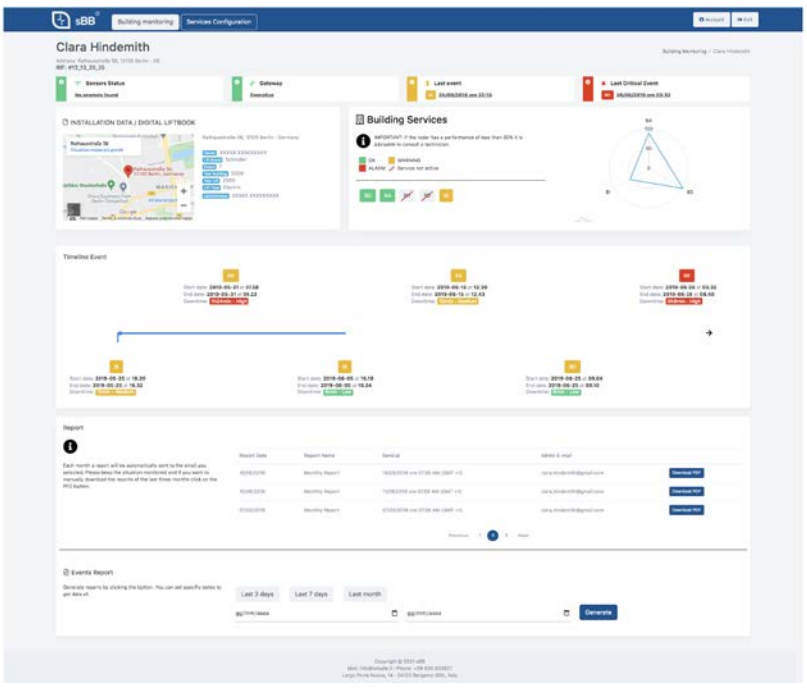
Dashboard

Visualizzazione dei dati

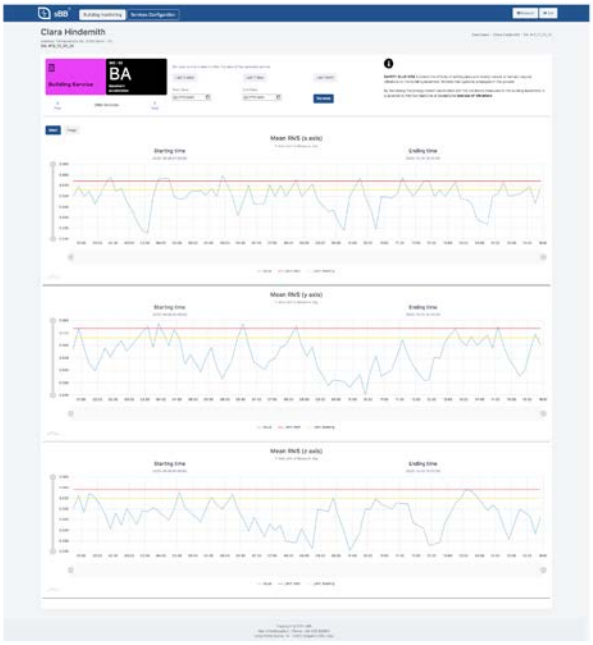
Dashboard – Entry page



Impianto pilota



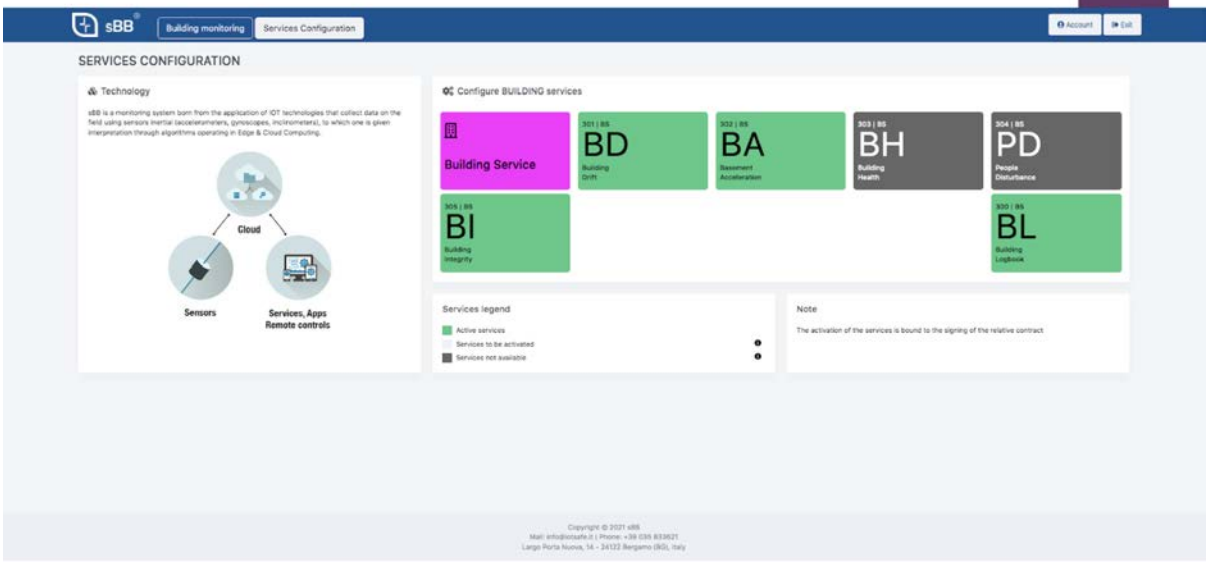
+ Servizio BA



+ Servizio BI

+ Servizio BD

+ Pagina di configurazione



31 IMPATTO ATTESO

L'impatto atteso dalle attività ed i risultati ottenuti per questo OR2 riguardano un aspetto molto importante dell'intero progetto INSIST, ovvero quello della raccolta dei dati e la loro analisi.

L'approccio seguito è stato quello di ricercare strumenti di analisi avanzata (Big Data Analytics e Machine Learning) da applicare al monitoraggio strutturale, fornendo anche una possibile soluzione per quel che riguarda il data storage su cloud.

32 CONCLUSIONI

Nonostante durante le fasi cruciali del progetto ci siano state, purtroppo, delle difficoltà create dall'emergenza COVID-19, tutti i partner hanno mostrato la massima disponibilità a rimodulare le proprie attività e a far sì che tutti i punti dell'OR2 fossero coperti al massimo dalla migliore risposta in termini di qualità e quantità dei risultati ottenuti, confermando un'elevata professionalità nella gestione e implementazione delle attività.