



UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

**DOTTORATO DI RICERCA IN  
Ingegneria Industriale curriculum Affidabilità**

CICLO XXXIII


COORDINATORE Prof. Tucci Mario

Analisi ed applicazione di modelli diagnostici e prognostici per guasti e  
prestazioni di componenti di impianti industriali nell'era I4.0

Settore Scientifico Disciplinare ING-IND/17

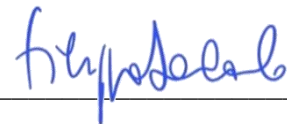
**Dottorando**

Dott. Navicelli Andrea

  
\_\_\_\_\_

**Tutore**

Prof. De Carlo Filippo

  
\_\_\_\_\_

**Coordinatore**

Prof. Tucci Mario

  
\_\_\_\_\_

Anni 2017/2020







# Sommario

Il ruolo fondamentale che la manutenzione gioca nei costi di esercizio e nella produttività degli impianti industriali ha portato le aziende e i ricercatori a spostare il loro interesse su questo tema. L'ultima frontiera dell'innovazione in campo manutentivo, resa possibile anche dall'avvento della quarta rivoluzione industriale che promuove la sensorizzazione e l'interconnessione di tutti i macchinari di impianto, è la manutenzione predittiva. Essa mira ad ottenere una previsione accurata della vita utile dei componenti degli impianti industriali al fine di ottimizzare la schedulazione degli interventi sul campo.

Lo studio parte da una accurata revisione della letteratura scientifica di settore riguardante le tecniche diagnostiche e prognostiche applicate a componenti di impianti industriali, necessaria alla comprensione dei diversi modelli sviluppati in funzione della tipologia di componente e modo di guasto in analisi. Successivamente ho spostato l'attenzione sul concetto di manutenzione 4.0 al fine di mappare tutte le caratteristiche associate al paradigma dell'Industria 4.0 e le loro possibili applicazioni alla manutenzione.

Lo studio condotto ha portato poi alla progettazione, sviluppo e validazione delle metodologie necessarie all'applicazione in real-time di modelli diagnostici e prognostici avanzati, sia statistici che machine learning, necessari all'implementazione sul campo di un sistema di manutenzione predittiva.

Grazie all'applicazione delle metodologie proposte ad un caso studio è stato possibile non solo validare i modelli proposti ma anche definire l'architettura informatica necessaria alla loro corretta implementazione sul sistema distribuito di controllo (Distributed Control System - DCS) di impianto in funzione della tipologia del componente e del guasto in analisi.

I modelli testati e validati hanno mostrato elevate prestazioni diagnostiche soprattutto per quanto riguarda i modelli ML che sfruttano le Support Vector Machine (SVM).

In definitiva, questo lavoro di tesi mostra nel dettaglio tutti i passaggi necessari allo sviluppo di un sistema di manutenzione predittiva efficace in impianto: partendo dall'analisi dei modi di guasto e dalla sensorizzazione dei componenti, passando poi allo sviluppo dei modelli diagnostici e prognostici real-time fino alla costruzione dell'interfaccia di visualizzazione dei risultati delle analisi svolte, analizzando anche l'architettura informatica necessaria al suo corretto funzionamento.



# Abstract

The fundamental role that maintenance plays in the operating costs and productivity of industrial plants has led companies and researchers to shift their interest in this issue. The last frontier of innovation in the maintenance field, made possible also by the advent of the fourth industrial revolution which promotes the sensorisation and interconnection of all plant machinery, is predictive maintenance. It aims to obtain an accurate forecast of the useful life of the industrial plants' components in order to optimise the scheduling of interventions in the field.

The study starts from an accurate review of the scientific literature concerning the diagnostic and prognostic techniques applied to industrial plant components, necessary to understand the different models developed according to the type of component and failure mode under analysis. Subsequently I shifted the focus to the maintenance 4.0 concept in order to map all the characteristics associated with the Industry 4.0 paradigm and their possible applications to maintenance operations.

The study then led to the design, development and validation of the methodologies necessary for the real-time application of advanced diagnostic and prognostic models, both statistical and machine learning, necessary for the field implementation of a predictive maintenance system.

Thanks to the application of the proposed methodologies to a case study, it was possible not only to validate the proposed models but also to define the IT architecture necessary for their correct implementation on the plant's Distributed Control System (DCS) according to the type of component and the fault under analysis.

The tested and validated models showed high diagnostic performance, especially regarding the Support Vector Machine (SVM) Machine Learning models.

Ultimately, this thesis shows in detail all the steps necessary for the development of an effective predictive maintenance system in the plant: starting from the analysis of failure modes and component sensorisation, then moving on to the development of real-time diagnostic and prognostic models up to the build-up of the interface for visualising the results of the analyses carried out, also analysing the IT architecture necessary for its correct operation.





---

# Indice

<b>1 - Introduzione.....</b>	<b>29</b>
1.1 - La gestione della manutenzione .....	30
1.1.1 - Storia della manutenzione .....	30
1.1.2 - Principi di analisi dell'affidabilità .....	31
1.1.3 - Politiche manutentive .....	38
1.1.4 - Costi di manutenzione.....	41
1.2 - Stato dell'arte delle tecniche di manutenzione predittiva.....	45
1.2.1 - Raccolta dei dati.....	46
1.2.2 - Analisi delle aree geografiche degli articoli .....	59
1.2.3 - Analisi dei settori industriali degli articoli selezionati.....	60
1.2.4 - Analisi dei componenti trattati negli articoli selezionati .....	61
1.2.5 - Analisi delle tecniche di stima della RUL .....	62
1.2.6 - Conclusioni sull'analisi preliminare della letteratura .....	109
1.3 - La manutenzione nell'era dell'industria 4.0.....	111
1.3.1 - Rivoluzioni industriali .....	112
1.3.2 - Vantaggi e sfide .....	114
1.3.3 - Tecnologie chiave.....	115
1.3.4 - Applicazione delle tecnologie 4.0 nella gestione della manutenzione .....	124
1.3.5 - DCS: Distributed Control System.....	126

---

1.4 - Domande di ricerca .....	129
<b>2 - Metodi.....</b>	<b>130</b>
2.1 - Approccio alla prognostica .....	130
2.2 - Modello prognostico .....	131
2.3 - Modello di stima della RUL in assenza di dati di guasto .....	134
2.4 - Architettura informatica .....	135
2.5 - Modelli utilizzati nel caso studio .....	138
2.5.1 - Validazione delle misure acquisite dai sensori .....	138
2.5.2 - Modello diagnostico SVM .....	142
2.5.3 - Modello diagnostico - Riconoscitore di forme d'onda .....	143
2.5.4 - Analisi di serie temporali ARIMA.....	145
2.5.5 - Modello di state detection – Carte di controllo multivariate.....	155
2.5.6 - Modello di state detection – One class SVM .....	156
<b>3 - Caso studio .....</b>	<b>159</b>
3.1 - Gli impianti geotermici.....	161
3.1.1 - Classificazione dei sistemi geotermici.....	162
3.1.2 -- Tecnologie di generazione geotermoelettriche .....	163
3.1.3 -- Ciclo a condensazione a singolo flash .....	164
3.2 - DCS di impianto: PI System .....	169
3.2.1 - Architettura informatica per modelli soft real-time .....	172

3.2.2 - Architettura informatica per modelli hard real-time.....	174
3.2.3 - La logica object-oriented.....	170
3.3 - Implementazione del sistema di validazione delle misure.....	175
3.3.1 - Diagnostica sui sensori ridondati del condensatore.....	175
3.3.2 - Analisi per i sensori ridondati.....	176
3.3.3 - Efficacia del metodo di diagnostica predittiva.....	183
3.4 - Analisi delle problematiche di impianto.....	186
3.4.1 - Analisi FMECA.....	188
3.4.2 - Analisi di letteratura delle tecniche prognostiche per i componenti critici.....	198
3.5 - Sensorizzazione dei componenti di interesse.....	218
3.5.1 - Rockwell – Allen-Bradly Dynamix 1444.....	224
3.5.2 - Dewesoft.....	225
3.6 - Riflessione sui Big data.....	216
3.7 - Applicazione di modelli diagnostici al compressore.....	227
3.7.1 - Modello diagnostico – SVM.....	227
3.7.2 - -Modello diagnostico – riconoscimento di forme d’onda (ShapeFind).....	257
3.7.3 - Rilascio dei modelli su PLC Siemens SIMATIC STEP 7.....	271
3.8 - Modello di stima della RUL in assenza di dati di guasto.....	278
3.8.1 - Applicazione delle carte di controllo multivariate.....	279
3.8.2 - Applicazione del One-class SVM.....	280

3.8.3 - Applicazione del modello ARIMA.....	281
3.9 - Interfaccia modelli prognostici soft real-time .....	284
3.10 - Prototipo di applicativo per prognostica soft real-time.....	291
<b>4 - Discussione .....</b>	<b>297</b>

# Elenco delle Figure

Figura 1.1	Andamenti tipici del tasso di guasto per componenti meccanici .....	36
Figura 1.2	Le politiche manutentive secondo le UNI EN 13306 e UNI 10147 .....	39
Figura 1.3	Grafico dei costi di manutenzione preventiva e correttiva .....	42
Figura 1.4	Schema a blocchi dei costi della manutenzione .....	43
Figura 1.5	Numero di articoli pubblicati per anno (Keywords: Predictive Maintenance, Limits: Articles Fonte: Scopus.com) .....	45
Figura 1.6	numero di articoli per Keywords Scopus.....	48
Figura 1.7	numero di articoli per caso IEEE .....	50
Figura 1.8	Percentuale per anno degli articoli selezionati.....	59
Figura 1.9	Paesi finanziatori degli articoli selezionati .....	60
Figura 1.10	Settori industriali degli articoli selezionati .....	61
Figura 1.11	Componenti trattati negli articoli selezionati .....	62
Figura 1.12	Classificazione modelli prognostici (Liao and Köttig, 2014) .....	64
Figura 1.13	Classificazione modelli prognostici (Lee et al., 2019).....	64
Figura 1.14	Modelli prognostici negli articoli selezionati .....	65
Figura 1.15	Variabili considerate per la stima della RUL.....	65
Figura 1.16	Percentuale dei metodi di stima della RUL utilizzati per il componente cuscinetto .....	67
Figura 1.17	Tecniche di stime della RUL degli articoli selezionati .....	67
Figura 1.18	Tecniche statistiche di stime della RUL degli articoli selezionati .....	68
Figura 1.19	Esempio di catena markoviana del prim'ordine .....	69
Figura 1.20	Diagramma di stato per componenti non riparabili .....	70

Figura 1.21 Diagramma di stato per componenti riparabili .....	70
Figura 1.22 Tipica struttura di Markov .....	71
Figura 1.23 Schema del modello HSMM .....	75
Figura 1.24 Modello MB-HSMM di diagnostica e prognostica.....	75
Figura 1.25 Metodologia utilizzata nell'articolo.....	78
Figura 1.26 Metodologia utilizzata per il modello regressivo .....	87
Figura 1.27 Metodologia proposta per il modello regressivo .....	91
Figura 1.28 Support Vectors .....	95
Figura 1.29 Iperpiano di separazione tra le classi .....	96
Figura 1.30 Struttura dell'algoritmo di Support Vector.....	97
Figura 1.31 Support Vector Regression .....	98
Figura 1.32 Support Vector Regression-funzioni di rilassamento .....	99
Figura 1.33 trasformazione lineare dello spazio dei dati.....	100
Figura 1.34 Metodologia di applicazione delle SVR .....	100
Figura 1.35 Schema rete neurale.....	102
Figura 1.36 Schema di funzionamento di una rete neurale .....	103
Figura 1.37 Forward and Backward propagation (Gullo, 2018) .....	104
Figura 1.38 Metodologia utilizzata .....	105
Figura 1.39 Struttura Restricted Boltzmann Machines .....	106
Figura 1.40 Metodologia di ricerca dei dati .....	110
Figura 1.41 Metodologia generale di stima della RUL.....	111
Figura 1.42 La quarta rivoluzione industriale .....	112

Figura 1.43 Rappresentazione dei Sistemi Cyber-Fisici e dell'Internet of Things .....	116
Figura 1.44 Tecnologie abilitanti ed elementi dell'industria 4.0 (Ahuett-Garza and Kurfess, 2018)....	117
Figura 1.45 Tecnologie abilitanti per IoT (Li et al., 2015).....	118
Figura 1.46 Una tassonomia multidimensionale dell'apprendimento automatico (Zhou et al., 2017). .....	120
Figura 1.47 Applicazione della Realtà Aumentata nel contesto industriale (Alcácer and Cruz-Machado, 2019).....	122
Figura 1.48 Struttura DCS. ....	127
Figura 2.1 Approccio alla prognostica.....	130
Figura 2.2 Framework del modello prognostico real-time. ....	132
Figura 2.3 Framework del modello di stima della RUL in assenza di dati di guasto. ....	135
Figura 2.4 Logica dell'algoritmo ARIMA training. ....	148
Figura 2.5 Logica dell'algoritmo ARIMA init. ....	151
Figura 2.6 Logica dell'algoritmo ARIMA real-time.....	153
Figura 3.1 Struttura del progetto SmartGEO.....	160
Figura 3.2 percentuale di produzione di energia elettrica da fonti geotermiche in Europa nel 2015.	161
Figura 3.3 percentuale di produzione di energia elettrica da fonti geotermiche nel mondo nel 2015. .....	162
Figura 3.4 sorgente idrotermale. ....	163
Figura 3.5 cicli di generazione di energia elettrica per i sistemi idrotermali. ....	164
Figura 3.6 tipico schema di funzionamento del ciclo a condensazione a singolo flash i. ....	164
Figura 3.7 separatore ciclonico. ....	166
Figura 3.8 condensatore.....	167

Figura 3.9 schema dell'impianto AMIS. ....	168
Figura 3.10 pompa PAE. ....	168
Figura 3.11 torri di raffreddamento. ....	169
Figura 3.12 Architettura di un PI System su cui si basano anche gli impianti di produzione di energia elettrica di Enel. ....	170
Figura 3.13 Integrazione Matlab Production Server (MPS) e Pi System 2018. ....	173
Figura 3.14 Struttura informatica Hard real-time. ....	174
Figura 3.15 Nel progetto vi è l'idea di mantenere in stretto collegamento il sistema di diagnostica con una rappresentazione fedele, secondo classi, dell'impianto di Rancia2. Tale metodo, poi, deve essere esteso a tutti gli altri siti del parco geotermico. ....	172
Figura 3.16 Analisi che confronta istante per istante la differenza a coppie tra i sensori e ne restituisce il valore al momento stesso della valutazione. ....	177
Figura 3.17 Sono rappresentate tutte le espressioni dell'analisi che serve per addestrare il sistema al test di diagnostica. ....	178
Figura 3.18 Nell'immagine si nota che l'attributo "TobeTrainedIn" è figlio dell'attributo "TobeTrained", con la proprietà aggiuntiva di essere non visibile ("hidden"). ....	179
Figura 3.19 Si vede in figura l'espressione che permette di ottenere il valore finale di output, una volta che vengono valutate tutte le possibilità presenti nella tabella della verità. ....	181
Figura 3.20 Le funzioni che compaiono in figura, servono a calcolare la qualità e il punteggio del valore di output. Il risultato tiene conto di tutti i casi ottenuti dalla tabella della verità. ....	182
Figura 3.21 E' qui rappresentata l'analisi completa che porta al calcolo del valore finale, della sua qualità e del suo punteggio. Il risultato ottenuto deriva dall'aver prima valutato tutti i casi legati ai superamenti di soglia. ....	183
Figura 3.22 La figura rappresenta l'andamento della differenza tra A e C, nel periodo di tempo selezionato. La riga rossa orizzontale, invece, indica il valore di UCL_AC che è stato usato per il test diagnostico. ....	184



Figura 3.23 La figura rappresenta l'andamento della differenza tra A e B, nel periodo di tempo selezionato. La riga rossa orizzontale, invece, indica il valore di UCL_AB che è stato usato per il test diagnostico. ....	184
Figura 3.24 La figura rappresenta l'andamento della differenza tra B e C, nel periodo di tempo selezionato. La riga rossa orizzontale, invece, indica il valore di UCL_BC che è stato usato per il test diagnostico. ....	185
Figura 3.25 tabella FMEA di prodotto.....	188
Figura 3.26 tabella FMEA di processo. Sono state riportate solo le prime colonne in quanto la differenza rispetto alla tabella vista in precedenza consiste solo in queste. ....	192
Figura 3.27 grafici riportanti esempi di derive dei segnali di temperatura o vibrazione in caso di guasto di una valvola [25]. ....	200
Figura 3.28 deriva dei misuratori di coppia e posizione in casi di guasti alle valvole.....	201
Figura 3.29 esempio di test di calcolo della RUL di una valvola .....	202
Figura 3.30 rappresentazione degli effetti del "Time Synchronus Averaging" di un segnale vibrazionale. Più campioni di segnale si estraggono mediante il trigger e più il campione finale, ottenuto come media dei campioni di partenza e privato del rumore, sarà pulito. ....	204
Figura 3.31 illustrazione degli effetti sul rumore di fondo della tecnica TSA tratti. Dai grafici in figura (estratti da lavori della Crystal Instruments) è possibile notare come un segnale non sincrono anomalo possa essere immediatamente riconosciuto mediante tecnica TSA.....	204
Figura 3.32 illustrazioni di come la tecnica basata sulle Wavelet Transform è in grado di rilevare i guasti. In questo caso gli autori hanno utilizzato una Discrete Wavelet Transform con Kaiser windows .....	205
Figura 3.33 Decision Tree elaborato da Sakthivel .....	206
Figura 3.34 Confronto tra <i>wavelet</i> applicate a segnali di una pompa sana ed una difettosa .....	207
Figura 3.35 confronto tra pompa sana (sx) e pompa guasta (dx) in una simulazione con utilizzo delle wavelet .....	207
Figura 3.36 immagini di cuscinetti interni al motore elettrico che riportano danni alla pista esterna (a sinistra) e a quella interna (a destra) .....	209

Figura 3.37 spettro di frequenza normalizzata della line current (analisi MCSA) in un motore sano (in alto) e in motori con criticità di danno al rotore crescente (immagini successive procedendo dall'alto verso il basso) .....	210
Figura 3.38 analisi dello spettro della line current normalizzata su un motore sano (in alto) e su un motore con disallineamenti (in basso) intorno alla frequenza fondamentale di funzionamento. ....	211
Figura 3.39 segnali di esperimenti con tecnica MCSA che dimostrano, in questo caso analizzando la corrente storica, delle differenze che si notano tra motore sano (in alto) e motore danneggiato (in basso) .....	211
Figura 3.40 caratteristica di funzionamento del compressore in stallo graduale, stallo improvviso e pompaggio .....	213
Figura 3.41 nascita e sviluppo dello stallo rotante .....	214
Figura 3.42 funzionamento stabile e instabile in termini energetici .....	215
Figura 3.43 Geolocalizzazione della Pompa PAE e dei Ventilatori all'interno dell'impianto pilota.....	219
Figura 3.44 Posizionamento quadri per punti di misura .....	221
Figura 3.45 Esempio installazione trasduttori su Pompa PAE. (A) Motore, (B) Corpo pompa .....	222
Figura 3.46 Esempio installazione trasduttori su ventilatore.....	222
Figura 3.47 Schematizzazione monitoraggio Pompa PAE e Ventilatori .....	223
Figura 3.48 Schema generale architettura Rockwell per Pompa PAE e Ventilatori .....	224
Figura 3.49 Schema di dettaglio architettura Rockwell per Pompa PAE e Ventilatori.....	225
Figura 3.50 Schema di dettaglio architettura Dewesoft per Pompa PAE e Ventilatori .....	225
Figura 3.51 Architettura ridondante sistemi di acquisizione dati .....	226
Figura 3.52 dati forniti da Enel Green Power S.p.a .....	228
Figura 3.53 celle e sensori corrispondenti .....	229
Figura 3.54 contenuto delle celle .....	230
Figura 3.55 gran parte dei sensori a cui fanno riferimento i dati (evidenziati) .....	231

Figura 3.56 sensori sincronizzati all'interno di un file .....	233
Figura 3.57 valori medi delle ridondanze .....	234
Figura 3.58 esempio di andamento delle misure rilevate dal sensore RA2_GS_DT_404. ....	235
Figura 3.59 esempio di andamento delle misure rilevate dal sensore RA2_GS_PT_450. ....	235
Figura 3.60 malfunzionamento del sensore RA2_GS_DT_403.....	236
Figura 3.61 malfunzionamento del sensore RA2_GS_DT_401 nel file RA2_COMPRESSORE_POMPAGGIO_160620171300_160620171500.mat.....	236
Figura 3.62 filtro delle misure rilevate dal sensore RA2_GS_PT_401 con limite di inizio del pompaggio. .....	238
Figura 3.63 filtro delle misure rilevate dal sensore RA2_EL_W_801 con limite di inizio del pompaggio. .....	238
Figura 3.64 filtro delle misure rilevate dal sensore RA2_GS_DT_401 con limite di inizio del pompaggio. .....	239
Figura 3.65 filtro delle misure rilevate dal sensore RA2_GS_DT_402 con limite di inizio del pompaggio. .....	239
Figura 3.66 filtro delle misure rilevate dal sensore RA2_GS_PT_402 con limite di inizio del pompaggio. .....	240
Figura 3.67 filtro delle misure rilevate dal sensore RA2_GS_PT_404 con limite di inizio del pompaggio. .....	240
Figura 3.68 filtro delle misure rilevate dal sensore RA2_GS_PT_405 con limite di inizio del pompaggio. .....	241
Figura 3.69 filtro delle misure rilevate dal sensore RA2_GS_PT_406 con limite di inizio del pompaggio. .....	241
Figura 3.70 filtro delle misure rilevate dal sensore RA2_GS_PT_452 con limite di inizio del pompaggio. .....	242
Figura 3.71 filtro delle misure rilevate dal sensore RA2_GS_PT_301 con limite di inizio del pompaggio. .....	242

Figura 3.72 risultato dell'unione. ....	244
Figura 3.73 Dataset_pompaggio.mat. ....	244
Figura 3.74 codice di estrazione casuale. ....	245
Figura 3.75 risultato dell'estrazione. ....	245
Figura 3.76 Dataset_regime.mat. ....	246
Figura 3.77 codice di NCA. ....	247
Figura 3.78 Risultato NCA: pesi delle variabili considerate. ....	247
Figura 3.79 risultati dell'addestramento del modello con kernel lineare sul dataset composto da 7 variabili. ....	250
Figura 3.80 risultati dell'addestramento del modello con kernel gaussiano medio sul dataset composto da 7 variabili. ....	251
Figura 3.81 risultati dell'addestramento del modello con kernel gaussiano fine sul dataset composto da 7 variabili. ....	251
Figura 3.82 Accuratezza previsionale sui singoli eventi di pompaggio. ....	254
Figura 3.83 tempi di permanenza nella classe del pompaggio di dati relativi alle condizioni di funzionamento a regime. ....	256
Figura 3.84 forma d'onda pseudo-periodica in condizioni di pompaggio. ....	258
Figura 3.85 forma d'onda forma d'onda sporadica in condizioni di incipiente pompaggio. ....	258
Figura 3.86 cicli giornalieri di pressione in condizioni di regime. ....	259
Figura 3.87 esempio di anomalia. ....	259
Figura 3.88 esempio di anomalia dovuta al blocco della centrale. ....	260
Figura 3.89 Firma tipica. ....	260
Figura 3.90 Firma bicornio. ....	261
Figura 3.91 Firma inversa. ....	261

Figura 3.92 Firma discendente esponenziale. ....	262
Figura 3.93 Anomalia 1: presenza di segnali di innesco pompaggio. ....	269
Figura 3.94 Anomalia 7: presenza di segnali di innesco pompaggio. ....	269
Figura 3.95 Anomalia 9: presenza di segnali di innesco pompaggio. ....	270
Figura 3.96 Pompaggio 28: Il pompaggio non viene riconosciuto. ....	270
Figura 3.97 Regime 43: presenza di segnali di innesco pompaggio. ....	271
Figura 3.98 Organizzazione generale del progetto Simulink per PLC Siemens. ....	273
Figura 3.99 Sottostruttura "Compressore", identificazione delle parti di progetto dedicate agli algoritmi sviluppati. ....	274
Figura 3.100 Struttura dell'algoritmo SVM. ....	275
Figura 3.101 Struttura dell'algoritmo ShapeFind. ....	277
Figura 3.102 Esempio di divisione delle condizioni operative del macchinario. ....	279
Figura 3.103 Evoluzione nel tempo della media mobile della variabile di controllo $T^2$ . ....	280
Figura 3.104 Evoluzione nel tempo della media mobile della variabile di controllo Likelihood. ....	281
Figura 3.105 Prima previsione ARIMA della RUL. ....	283
Figura 3.106 Servizi PI Web API. ....	285
Figura 3.107 Landing page del sito SmartGeoDiana. ....	286
Figura 3.108 Pagina multistep del sito SmartGeoDiana (Lista analisi). ....	287
Figura 3.109 Pagina multistep del sito SmartGeoDiana (Lista event frame). ....	288
Figura 3.110 Pagina multistep del sito SmartGeoDiana (Definizione tipo analisi). ....	288
Figura 3.111 Pagina multistep del sito SmartGeoDiana (Selezione tag PI) ....	289
Figura 3.112 – Pagina multistep del sito SmartGeoDiana (Selezione range temporale) ....	289

Figura 3.113 – Pagina multistep del sito SmartGeoDiana (Configurazione dati advanced) .....	290
Figura 3.114 – Pagina grafico scatter dei risultati.....	290
Figura 3.115 – Schermata di addestramento.....	293
Figura 3.116 – Finestra “Impostazioni” della schermata di previsione realtime della RUL.....	294
Figura 3.117 – Finestra “Risultati” della schermata di previsione realtime della RUL.....	295







# Elenco delle Tabelle

Tabella 1.1	Keywords e numero di articoli Scopus.....	47
Tabella 1.2	Keywords e numero di articoli IEEE caso 1.....	49
Tabella 1.3	Keywords e numero di articoli IEEE caso 2.....	49
Tabella 1.4	Keywords e numero di articoli IEEE caso 3.....	49
Tabella 1.5	Keywords e numero di articoli IEEE caso 4.....	49
Tabella 1.6	Parametri degli articoli scelti su Scopus .....	51
Tabella 1.7	Parametri degli articoli scelti su IEEE .....	56
Tabella 1.8	Valor medio dei parametri degli articoli scelti su Scopus .....	58
Tabella 1.9	Valor medio dei parametri degli articoli scelti su IEEE.....	58
Tabella 1.10	Analisi componente analizzato – variabili selezionate.....	65
Tabella 1.11	Analisi componente analizzato – metodo scelto .....	66
Tabella 1.12	Differenze fondamentali di transizione all'industria 4.0 dalle precedenti rivoluzioni industriali (Popkova et al., 2019). .....	114
Tabella 1.13	Panoramica dell'implementazione dell'Industria 4.0 nella gestione della manutenzione. 124	124
Tabella 2.1	.Livelli di rilascio dei modelli e di analisi on-line dei dati .....	137
Tabella 2.2	La tabella della verità mostra 4 categorie di situazioni degradate, che si hanno in base a quante coppie di sensori superano la soglia. Per ogni caso, è riportato il valore da preferire come finale e lo stato di qualità di tale valore.....	140
Tabella 2.3	esempio degli intervalli di confidenza in funzione del parametro nsigma .....	154
Tabella 3.1	descrizione dei campi presenti nella tabella FMECA di prodotto. ....	189
Tabella 3.2	descrizione dei campi presenti nella tabella FMECA di processo. ....	192

Tabella 3.3	serie di tabelle nelle quali sono riportate le scale utilizzate per la definizione degli indici. Menzione particolare per la scala relativa all'impatto sulla sicurezza del personale per la quale abbiamo tenuto conto non solo dei danni a cui gli operatori possono incorrere, ma anche della probabilità che il danno possa essere di quella entità, in modo da soppesare ad esempio un danno grave o irreversibile ma con probabilità di accadimento raro allo stesso modo di un danno lieve con alta probabilità di accadimento. ....	194
Tabella 3.4	tabella dei questionari AHP per il livello 1.....	196
Tabella 3.5	tabella dei questionari AHP per il livello 4.....	197
Tabella 3.6	sigla dei sensori e grandezze misurate. ....	231
Tabella 3.7	Tabella delle ridondanze. ....	233
Tabella 3.8	grandezze oscillanti. ....	237
Tabella 3.9	parametri ottimali per il codice di filtro. ....	237
Tabella 3.10	pesi delle variabili. ....	247
Tabella 3.11	variabili considerate nel primo tentativo di addestramento SVM.....	249
Tabella 3.12	Performance dei modelli SVM testati.....	252
Tabella 3.13	Accuratezza previsionale sui singoli eventi di pompaggio. ....	253
Tabella 3.14	Variabili necessarie per la prognostica del compressore. ....	255
Tabella 3.15	Periodi di regime o di anomalia, insieme di parametri "Due Firme".....	264
Tabella 3.16	Periodi di pompaggio, insieme di parametri "Due Firme". ....	264
Tabella 3.17	Periodi di regime o di anomalia, insieme di parametri "Firma inversa".....	265
Tabella 3.18	Periodi di pompaggio, insieme di parametri "Firma inversa". ....	265
Tabella 3.19	Periodi di regime o di anomalia, insieme di parametri "Firma discendente". ....	266
Tabella 3.20	Periodi di pompaggio, insieme di parametri "Firma discendente".....	266
Tabella 3.21	Periodi di regime o di anomalia, insieme di parametri "Firma discendente, distanza ridotta".	267

Tabella 3.22	Periodi di pompaggio, insieme di parametri “Firma discendente, distanza ridotta”. 267
Tabella 3.23	Performance dei modelli di state detection e previsione della RUL. .... 283



# 1 - Introduzione

I costi di manutenzione contribuiscono per gran parte del totale dei costi operativi di tutti gli impianti manifatturieri o di produzione. Tali costi variano mediamente, in funzione della tipologia di industry, tra il 15 ed il 60 per cento del costo del prodotto finito (Mobley, 2002).

Lo scopo dello studio è quello di sviluppare ed implementare modelli predittivi per guasti e prestazioni di componenti di impianti industriali al fine di valutarne la vita utile residua (RUL - Remaining Useful Life) ed ottimizzare la schedulazione e la gestione degli eventi manutentivi, oltre ad ampliare al massimo la disponibilità dei macchinari incrementando in ultima analisi la produttività e riducendo i costi manutentivi.

La manutenzione predittiva, una delle 3 principali tecniche manutentive insieme alla manutenzione a guasto e alla preventiva, parte dalla premessa che un monitoraggio continuo delle attuali condizioni meccaniche, di efficienza ed altri indicatori delle condizioni operative del treno macchine e dei sistemi di processo, forniscano i dati necessari a garantire il massimo intervallo tra interventi manutentivi, minimizzando il numero ed il costo di interruzioni di produzione non schedulati (Hashemian, 2010).

Per ottenere un modello predittivo è quindi necessaria l'applicazione di una gamma di sensori ai macchinari e l'implementazione di una forte infrastruttura tecnologica ed informatica (ICT – Information and Communication Technology) che permetta il loro monitoraggio continuo e la loro analisi sia on-line che off-line (Lee et al., 2004).

Questa tematica, che si inserisce a pieno all'interno della quarta rivoluzione industriale, la cosiddetta Industria 4.0, risulta un argomento particolarmente attuale e di interesse per le aziende dati anche gli ingenti finanziamenti statali stanziati per il suo sviluppo (Tiraboschi and Seghezzi, 2016).

Nei prossimi paragrafi verrà illustrata la teoria della gestione della manutenzione e dell'analisi di affidabilità, l'analisi della letteratura riguardo le più innovative tecniche diagnostiche e prognostiche e il concetto di manutenzione 4.0 definendo poi le domande di ricerca alle quali questa tesi intende rispondere.

## 1.1 - La gestione della manutenzione

“La manutenzione è una “scienza” antica quanto l’invenzione della macchina stessa, poiché qualsiasi attrezzatura presenta la naturale tendenza ad usurarsi e conseguentemente a rompersi. La manutenzione, infatti, è quella combinazione di tecniche amministrative, incluse azioni di supervisione, volte a mantenere o a riportare una entità in uno stato in cui possa eseguire la funzione richiesta” (UNI EN 13306).

Il settore della gestione della manutenzione negli anni ha avuto un’importanza sempre più centrale in campo industriale poiché attraverso una corretta gestione della manutenzione è possibile ridurre drasticamente i costi di produzione e quindi ottenere un sostanzioso vantaggio competitivo.

In questo capitolo si vuole inquadrare quella che è la funzione della manutenzione negli impianti industriali e far emergere l’importanza di una corretta gestione della manutenzione per ottenere il miglior trade-off tra costi connessi a tale attività e produttività dei macchinari.

### 1.1.1 - Storia della manutenzione

Negli ultimi decenni il concetto di manutenzione si è evoluto e rinnovato profondamente, la sua storia è fortemente legata ai cambiamenti della produzione industriale e dei modi di produrre. Questi cambiamenti hanno portato la manutenzione ad avere un ruolo sempre più complesso e multidisciplinare.

Oggi la manutenzione è un’attività strategica che permette di aumentare le prestazioni di un’azienda in modo da poter competere in mercati sempre più competitivi e fortemente globalizzati.

Prima dello sviluppo della produzione di tipo industriale i modi di produrre e di fare manutenzione erano attività di tipo artigianale. Molto spesso era lo stesso utilizzatore ad eseguire la manutenzione degli attrezzi.

All’inizio del XX secolo si sviluppa, soprattutto negli Stati Uniti grazie alla spinta nel settore automobilistico di Henry Ford, un approccio alla produzione di tipo industriale orientato alla produzione di massa. La produzione di massa determina per la manutenzione un aumento del livello di specializzazione e fa sì che diventi un ruolo professionale a tutti gli effetti. Il risultato di questo sviluppo è una suddivisione del personale manutentivo in una serie di gruppi operativi specializzati.

Il secondo grande cambiamento, che influirà in maniera significativa l'approccio alla manutenzione, avviene alla fine della Seconda Guerra Mondiale in Giappone, grazie allo sviluppo del concetto di produzione snella. I precursori di questa rivoluzione industriale furono i due ingegneri Eiji Toyoda e Taiichi Ohno, che svilupparono in Toyota un sistema di produzione basato sulla flessibilità produttiva e sulla diminuzione della difettosità. Viene introdotto il concetto di Kaizen, ossia di miglioramento continuo del processo produttivo e si realizza una forte integrazione tra ingegneria di prodotto, di processo e industriale. Con il lavoro di squadra si vuole accrescere la partecipazione e la creatività del personale e si vuole eliminare l'eccesso di specializzazioni e di livelli che spesso determinavano un punto di debolezza nella produzione di massa dal punto di vista del coordinamento. L'approccio snello alla produzione si è esteso anche alla manutenzione ed è basato su tre principi:

- Integrazione tra le funzioni produzione e manutenzione nelle responsabilità manutentive di tipo operativo.
- Creazione della figura dell'operatore di processo, come esecutore della manutenzione e controllore dell'andamento delle macchine o dell'impianto di cui è conduttore.
- Sviluppo dell'ingegneria di manutenzione come servizio trasversale alle responsabilità produttive.

L'organizzazione snella della manutenzione è basata su un modello organizzativo incentrato su figure operative multifunzionali, in stretta collaborazione con la parte produttiva, e da strutture ingegneristiche centrali che progettano, controllano e migliorano la manutenzione.

La corretta scelta delle politiche di gestione della manutenzione non può prescindere da un'analisi dell'affidabilità dei componenti di impianto.(Furlanetto et al., 2006)

### **1.1.2 - Principi di analisi dell'affidabilità**

In teoria dei sistemi, probabilità e Ingegneria dell'affidabilità, definendo come "qualità" di un assieme o di un sistema più o meno complesso, la sua rispondenza ai criteri di specifica tecnica di funzionamento, si definisce affidabilità la capacità di rispettare le specifiche tecniche di funzionamento nel tempo.

Si definisce quindi affidabilità di un componente la probabilità che esso funzioni senza guastarsi per un certo periodo di tempo  $t$  con riferimento a ben precisate condizioni ambientali.

Al fine di determinare l'affidabilità di un componente o di un sistema di componenti è necessario avere definito i seguenti elementi:

- Un criterio univoco per giudicare se un elemento è funzionante o meno
- Esatta definizione delle condizioni ambientali e d'impiego
- Determinazione dell'intervallo di tempo  $t$

Una volta fissate le prime due condizioni l'affidabilità di un elemento diviene funzione solo del tempo e la forma di tale funzione dipende dalla legge probabilistica con cui le condizioni di cattivo funzionamento o di cedimento possono verificarsi nel tempo.

Esistono tre approcci per valutare l'affidabilità di una macchina:

- a. Utilizzare le informazioni che provengono per un lungo periodo di tempo da molte macchine uguali nelle stesse condizioni di funzionamento;
- b. Utilizzare le informazioni che provengono dal funzionamento per un breve periodo di tempo di poche macchine. I dati possono fornire una stima del comportamento avente un certo grado di confidenza, ovvero una certa probabilità di risultare vera;
- c. Utilizzare la conoscenza, se esiste, dell'affidabilità dei componenti per fare previsioni sull'affidabilità dell'intera macchina.

Ciascun componente di un sistema ha una precisa funzione da assolvere. La specifica di un componente contiene numerosi dati tra cui la descrizione della funzione da assolvere, le interazioni con altri componenti e le condizioni ambientali nel quale si trova ad operare.

Il comportamento di un componente influenza il funzionamento complessivo del sistema a vari livelli:

- Assolve la sua funzione correttamente;
- Non assolve la sua funzione correttamente;
- Assolve solo parzialmente la sua funzione;
- Durante l'attività disturba un'altra funzione o un altro componente.

Il periodo di regolare funzionamento di un dispositivo si conclude quando un qualsiasi fenomeno fisico-chimico (guasto) prodottosi in una o più delle sue parti determina una variazione delle prestazioni nominali tali da ritenere inaccettabile il comportamento del dispositivo stesso. Il dispositivo passa dallo stato di funzionamento a quello di avaria.

Le principali cause di guasto sono le seguenti:



- a. Sollecitazioni, urto e fatica: Sono funzione della distribuzione temporale e spaziale delle condizioni di carico e della risposta del materiale. In questo caso assumono un ruolo importante le caratteristiche strutturali del componente, da valutarsi nella forma più ampia possibile, inglobando anche possibili errori progettuali, realizzativi, difetti del materiale, ecc.
- b. Temperatura: È una variabile operativa che influisce prevalentemente in funzione delle caratteristiche specifiche del materiale (inerzia termica), nonché della distribuzione spaziale e temporale delle sorgenti di calore.
- c. Usura: Si tratta di uno stato di degradazione fisica del componente; si manifesta in seguito a fenomeni di invecchiamento che si accompagnano alla normale attività (attrito fra materiali, esposizione ad agenti dannosi, ecc.)
- d. Corrosione: È un fenomeno che dipende dalle caratteristiche dell'ambiente in cui il componente si trova ad operare. Queste condizioni possono portare il materiale a processi di degradazione fisica e chimica capaci di rendere il componente non più idoneo.

Ognuna delle cause di guasto appena descritte possono avvenire durante le differenti fasi della vita di un componente o di un sistema di componenti. Una differente classificazione dei tipi di guasto è la seguente:

- a. Guasti infantili o prematuri: si manifestano nella prima fase di esercizio del sistema. Le cause sono spesso riconducibili ad una carenza strutturale, progettuale oppure a difetti di installazione. In termini di affidabilità un impianto sottoposto alla manifestazione di guasti infantili migliora il proprio stato con l'andare del tempo.
- b. Guasti accidentali o casuali: si hanno in condizioni di esercizio non nominali che mettono a dura prova i componenti, producendo delle inevitabili alterazioni con conseguente perdita delle capacità operative. Questo tipo di guasto si manifesta durante la vita utile dell'impianto e corrisponde a situazioni imprevedibili. La probabilità di manifestarsi di un guasto è indipendente dal periodo di esercizio accumulato.
- c. Guasti per usura: alterazioni del componente per invecchiamento strutturale e materiale. L'inizio del periodo di usura è individuato da una crescita della frequenza dei guasti fino al raggiungimento del valore massimo, per poi decrescere per effetto della scomparsa della popolazione. I guasti per usura si presentano intorno all'età media di funzionamento; l'unico modo di evitare questa tipologia di guasto è quello di operare un ricambio preventivo della popolazione.

Dopo aver definito l'affidabilità e le tipologie di guasto, passiamo alla definizione degli elementi di teoria dell'affidabilità.

Considerando come variabile il tempo di guasto di un elemento intercorrente tra l'istante iniziale e l'istante di guasto. Si definisce la **densità di probabilità di guasto** come la funzione  $f(t)$  tale che la probabilità infinitesima che l'elemento si rompa al tempo  $t$  o in un suo intorno infinitesimo  $dt$  sia proprio  $f(t)dt$ . L'area sottesa alla  $f(t)$  è uguale all'unità se il limite temporale superiore è posto all'infinito, dato che un elemento tende prima o poi a guastarsi, ovvero:

$$\int_0^{\infty} f(t) dt = 1$$

Di conseguenza la probabilità che l'elemento funzionante all'istante iniziale si guasti nel tempo  $t$  è data da:

$$F(t) = \int_0^t f(\tau) d\tau$$

E l'**affidabilità**, cioè la probabilità di sopravvivenza o di funzionamento corretto che dir si voglia al tempo  $t$  sarà:

$$R(t) = \int_t^{\infty} f(\tau) d\tau = 1 - F(t)$$

Il tempo medio tra due guasti consecutivi (nell'ipotesi che il componente sia riparabile) è il **MTBF Mean Time Between Failures** che viene calcolato come:

$$MTBF = \int_0^{\infty} \tau \cdot f(\tau) d\tau$$

Di conseguenza, la conoscenza della densità di probabilità di guasto,  $f(t)$ , permette di valutare l'affidabilità del componente.

Si definisce inoltre il **tasso di guasto** come la funzione  $\lambda(t)$  tale che la probabilità infinitesima che l'elemento si rompa al tempo  $t$  o in un suo intorno infinitesimo  $dt$  sia  $\lambda(t)dt$ , nell'ipotesi che in  $t$  sia ancora funzionante. Ovvero:

$$\lambda(t) = \lim_{\Delta t \rightarrow 0} \frac{P(t < \bar{t} < t + \Delta t | \bar{t} > t)}{\Delta(t)}$$

Il tasso di guasto ha le dimensioni dell'inverso di un tempo e può essere interpretato come il "numero di guasti nell'unità di tempo", ovvero come una misura della velocità di verificarsi del guasto.

Esiste una relazione tra densità di probabilità di guasto  $f(t)$  e tasso di guasto  $\lambda(t)$ , in particolare, definito:

- $NC(0)$ : numero di componenti tutti funzionanti al tempo  $t=0$ ;
- $NC(t)$ : numero di componenti ancora funzionanti al tempo  $t$
- $NC(t)=NC(0) \cdot R(t)$

Dalla definizione di affidabilità si ha:

$$R(t) = \frac{NC(t)}{NC(0)}$$

Dalla definizione di tasso di guasto si ha invece:

$$\lambda(t) = \frac{f(t) \cdot NC(0)}{NC(t)} = \frac{f(t)}{R(t)} \text{ e quindi } \lambda(t) \cdot R(t) = f(t)$$

Ricordando che:

$$f(t) = \frac{dF(t)}{d(t)} = -\frac{dR(t)}{d(t)}$$

Si ha:

$$dR(t) = -R(t)\lambda(t)dt \rightarrow \frac{dR(t)}{R(t)} = -\lambda(t)dt$$

Da cui:

$$\ln R(t) - \ln R(0) = -\int_0^t \lambda(\tau)d\tau \rightarrow \ln R(t) - \ln(1) = -\int_0^t \lambda(\tau)d\tau \rightarrow \ln R(t) = -\int_0^t \lambda(\tau)d\tau$$

Di conseguenza le relazioni generali che definiscono gli elementi della teoria dell'affidabilità sono:

$$R(t) = e^{-\int_0^t \lambda(\tau)d\tau}$$

$$f(t) = \lambda(t) \cdot R(t) = \lambda(t) \cdot e^{-\int_0^t \lambda(\tau)d\tau}$$

$$F(t) = 1 - R(t) = 1 - e^{-\int_0^t \lambda(\tau) d\tau}$$

Ogni componente o sistema sotto analisi, durante le sue diverse fasi della vita modifica le sue caratteristiche di funzionamento. In particolare, analizzando la vita dei componenti utilizzando il tasso di guasto otteniamo un andamento come quello in figura:

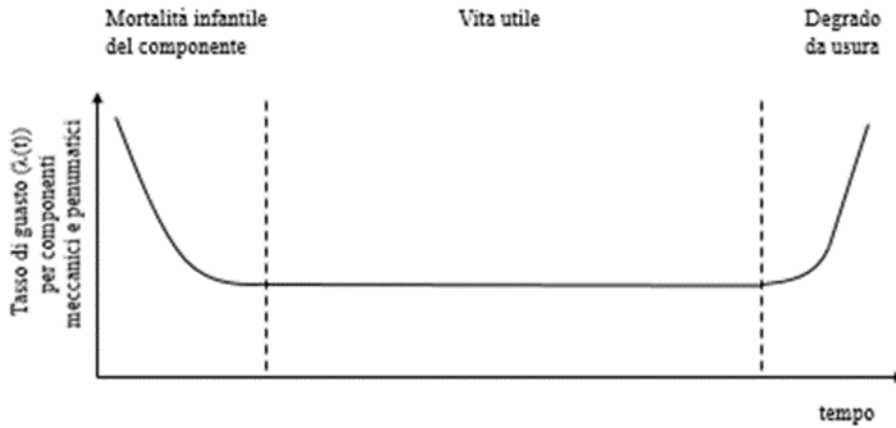


Figura 1.1 Andamenti tipici del tasso di guasto per componenti meccanici

Una buona progettazione del componente fa sì che la vita utile del componente sia maggiore della vita di progetto del componente stesso.

Nella Figura 1.1, sono ben evidenti le 3 fasi di vita del componente:

1. Mortalità infantile del componente: La fase di vita iniziale della macchina viene descritta con una distribuzione di Weibull. La percentuale di popolazione che cede al tempo  $t$  risulta essere espressa nella forma:

$$F(t) = 1 - e^{-\left(\frac{t}{\alpha}\right)^\beta}$$

Con i parametri  $\alpha$  e  $\beta$  positivi.  $\alpha$  corrisponde alla vita caratteristica del componente ed ha la dimensione di un tempo;  $\beta$  rappresenta il parametro di forma della funzione di Weibull. Questo valore varia generalmente tra 0,5 e 5; se  $\beta < 1$  il tasso di guasto è decrescente, ovvero descrive la fase iniziale della vita del componente. Altrimenti il tasso di guasto è crescente e può essere utilizzata per descrivere la fase finale di degrado o usura.

2. Vita Utile: è una fase della vita caratterizzata da un tasso di guasto sostanzialmente costante che caratterizza l'intera vita dei componenti elettronici ed è inoltre una buona approssimazione dei componenti meccanici ed elettromeccanici in questa fase di vita. Questa fase si descrive con la distribuzione esponenziale negativa corrispondente ad un tasso di guasto costante. Si dice che in

questo periodo la macchina “non ha memoria” nel senso che il suo comportamento è lo stesso qualunque sia stata la sua storia precedente. Sono guasti che risultano dalla combinazione di numerosi eventi di natura puramente casuale (detti alla Poisson). Data la natura costante del tasso di guasto intervenire con una sostituzione o riparazione preventiva del componente durante questa fase della vita non è assolutamente conveniente; infatti come visibile dal grafico, in questa fase il tasso di guasto e di conseguenza la probabilità che questo non si guasti in un determinato intervallo di tempo è al minimo e quindi non è migliorabile con un intervento manutentivo. Per questo come vedremo in seguito la miglior tecnica manutentiva applicabile in questo caso è la manutenzione correttiva che consiglia di intervenire sul componente solo se si è verificato il guasto. In questa fase a tasso di guasto costante le relazioni generali si semplificano e diventano le seguenti:

$$R(t) = e^{-\lambda t}$$

$$F(t) = 1 - R(t) = 1 - e^{-\lambda t}$$

$$f(t) = \lambda(t) \cdot R(t) = \lambda \cdot e^{-\lambda t}$$

$$MTBF = \int_0^{\infty} \tau \cdot f(\tau) d\tau = \int_0^{\infty} R(\tau) d\tau = \int_0^{\infty} e^{-\lambda \tau} d\tau = \frac{1}{\lambda}$$

3. Degrado da usura: è l'ultima fase della vita del componente che lo porta ad un incremento esponenziale della probabilità che si verifichi un guasto. I componenti meccanici arrivano in questa fase a seguito dell'invecchiamento dei materiali, al fenomeno dell'usura e della corrosione chimica. Questa fase della vita può essere descritta da una funzione di Weibull con  $\beta > 1$  ma tipicamente viene studiata tramite una funzione di densità di probabilità normale, caratterizzata da un caratteristico valor medio  $\mu$  e da una deviazione standard  $\sigma$ .

La funzione distribuzione di probabilità in questa fase della vita assume la ben nota forma:

$$f(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2}$$

Questa è la fase della vita del componente in cui agire con tecniche manutentive preventive, al fine di ripristinare con manutenzione perfetta e completo ripristino del componente (As Good As New - AGAN) o con manutenzione imperfetta e ripristino parziale, che, in entrambi i casi consente di abbassare il tasso di guasto e quindi la probabilità che in un certo intervallo di tempo il componente risulti guasto. (Furlanetto et al., 2006)

Nel prossimo capitolo verranno trattate nel dettaglio tutte le politiche manutentive da poter implementare per gestire la manutenzione dei componenti di impianti industriali.

### 1.1.3 - Politiche manutentive

L'interesse principale della manutenzione è quello di garantire un alto livello di disponibilità operativa e contenere i costi di manutenzione di macchine ed impianti durante il loro ciclo di vita.

Per conseguire gli obiettivi prefissati, la manutenzione deve essere progettata in modo da identificare quali siano le politiche manutentive più efficaci da mettere in atto. Identificare una politica manutentiva significa scegliere con che modalità andare a realizzare le attività di manutenzione sull'impianto o sul macchinario in oggetto, basandosi su scelte razionali che derivano dalla conoscenza approfondita dell'impianto o macchinario, dall'analisi dei guasti e da valutazioni di carattere economico.

Nello specifico si tratta di sviluppare dei ragionamenti su quando è più opportuno operare con interventi di riparazione solo in seguito ad un guasto, quando invece è preferibile prevenire i guasti effettuando interventi preventivi o quali possono essere le attività di manutenzione che possono determinare un miglioramento della situazione esistente.

In base alle normative UNI EN 13306 e UNI 10147 si identificano tre politiche di manutenzione principali:

- Manutenzione correttiva o a guasto;
- Manutenzione preventiva
- Manutenzione migliorativa

La manutenzione preventiva a sua volta può essere suddivisa in tre sotto politiche:

- Manutenzione preventiva ciclica;
- Manutenzione su condizione;
- Manutenzione predittiva.

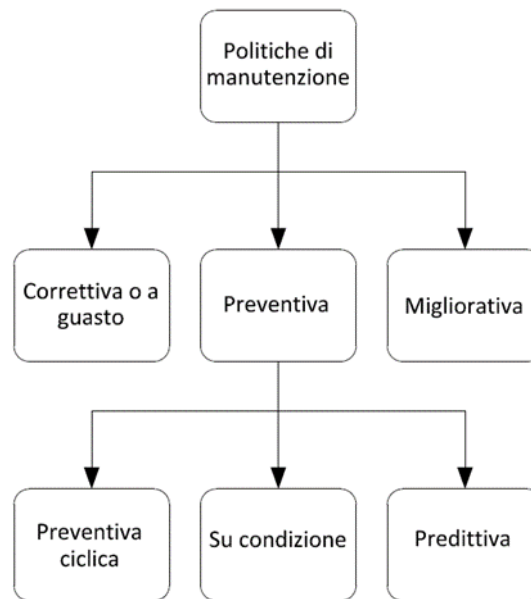


Figura 1.2 Le politiche manutentive secondo le UNI EN 13306 e UNI 10147

### 1.1.3.1 Manutenzione correttiva

La manutenzione correttiva è definita come “l’azione correttiva effettuata a causa di guasti o carenze rilevati durante la manutenzione preventiva, per ripristinare un componente/macchinario al suo stato operativo” (Dhillon, 2002). Generalmente, la manutenzione correttiva è un’azione non programmata, basata sul verificarsi di eventi specifici in un determinato momento. Rappresenta ancora una parte sostanziale dello sforzo di manutenzione generale.

Comprende le seguenti categorie di azioni:

- Manutenzione guasti;
- Manutenzione di emergenza;
- Manutenzione correttiva;
- Riparazioni;
- Interventi di ricostruzione.

Uno svantaggio di questo tipo di politica di manutenzione è dovuto a interruzioni impreviste della produzione.

Al fine di ridurre l’impatto sulle prestazioni del sistema con questo tipo di scelta nella politica manutentiva, potrebbero essere implementati sistemi ridondanti. Inoltre, volendo ridurre i tempi di

fermo per componenti con un alto tasso di guasti e lunghi tempi di consegna, i pezzi di ricambio devono essere gestiti consapevolmente.

Per ovviare agli svantaggi legati a questa politica, è necessario intercettare i fenomeni di degrado prima che possano portare a guasti totali e quindi essere in grado di correggere i difetti con il minor sforzo possibile.

Nonostante l'inefficienza insita in questa politica, la manutenzione correttiva è talvolta inevitabilmente necessaria poiché non è possibile prevedere tutti i tipi di guasto. In particolare, come già accennato, quei fenomeni di guasto dovuti ad eventi casuali (o alla Poisson) non possono per loro natura essere previsti e devono quindi essere gestiti con manutenzione correttiva.

### 1.1.3.2 Manutenzione preventiva

Il termine manutenzione preventiva si riferisce all'esecuzione di attività di manutenzione prima del verificarsi di un guasto, consentendo così una riduzione dei tempi di fermo non programmati e danni prematuri alle apparecchiature.

Si possono individuare tre differenti categorie di "prevenzione":

- **Predittiva:** La manutenzione predittiva può essere definita come "la manutenzione basata sulle condizioni, effettuata a seguito di una previsione derivata da analisi ripetute o caratteristiche note e valutazione dei parametri significativi del degrado del componente" (European Committee for Standardization (CEN, 2018). Di solito comprende il rilevamento e la misurazione di uno o più parametri e l'extrapolazione, secondo i modelli appropriati, del tempo rimanente prima del guasto (UNI 10147). Pertanto, le attività di manutenzione predittiva hanno lo scopo di identificare le condizioni tecniche di un componente al fine di prevederne la vita utile residua. In analogia alla manutenzione basata sulle condizioni, i segnali vengono misurati strumentalmente.
- **Su condizione:** È una politica per la quale viene effettuata la valutazione della necessità di intervento attraverso il monitoraggio delle condizioni effettive di un bene. Si concentra sulla misurazione di un parametro che indica un deterioramento o un degrado delle prestazioni funzionali dell'apparecchiatura (ad es. vibrazioni, temperatura di esercizio, amperaggio, contaminanti di olio lubrificante, livello di rumore, ecc.). In generale, l'azione è soggetta all'eccesso di limiti predefiniti. Questi limiti sono valori misurabili che riflettono le condizioni critiche di un componente o di un sistema. L'obiettivo della manutenzione basata sulle condizioni è quindi quello di determinare lo stato di deterioramento di uno o più componenti mediante la



misurazione di un parametro per avviare successivamente la manutenzione. A questo proposito, è importante notare che le operazioni di manutenzione vengono eseguite dopo aver superato il valore limite, una “soglia di avviso”, definita in modo che l’azione di manutenzione venga implementata prima che si verifichi il guasto.

- Programmata: È un tipo specifico di manutenzione preventiva che comprende le attività di manutenzione eseguite periodicamente in base a intervalli o cicli di utilizzo predeterminati e finalizzati a ridurre la probabilità di guasto o degrado del funzionamento del sistema. Questa politica di manutenzione mira a sostituire il componente danneggiato prima che si rompa o prima che possano verificarsi conseguenze impreviste sulle prestazioni operative. La manutenzione preventiva programmata comprende inoltre:
  - Controlli previsti dalla legge o dalle norme di sicurezza;
  - Pulizia;
  - Ispezioni generali.

La frequenza della manutenzione si basa solitamente su valori empirici definiti dall’esperienza di progettisti e operatori. L’azione di manutenzione può essere eseguita considerando un periodo di tempo costante o considerando l’utilizzo del componente. Il vantaggio principale della manutenzione preventiva programmata è che consente l’uso dei tempi di fermo della produzione per eseguire varie attività di manutenzione preventiva (a condizione che sia disponibile personale qualificato).

Non sempre la manutenzione predittiva ha dei benefici, poiché necessariamente c’è bisogno di un sistema di monitoraggio che richiede personale specializzato, il quale ha un costo di formazione che ha bisogno di un investimento importante. Inoltre, i modelli per calcolare la vita residua del componente non sempre sono di semplice implementazione e a volte, oltre che a dei sensori molto sviluppati, richiedono anche dei modelli di calcolo di elevata difficoltà ed una notevole e costosa infrastruttura informatica.

Un programma di manutenzione basato su ispezioni periodiche e riparazioni o sostituzioni è uno dei modi per controllare e intervenire sull’affidabilità di un sistema. Una politica di questo tipo, oggi, è ancora largamente considerata un costo evitabile se paragonato all’effettiva utilità della sua applicazione. Per poterla implementare con successo è necessario programmare e svolgere efficacemente le ispezioni e gestire le informazioni con criterio.

#### **1.1.4 - Costi di manutenzione**

La manutenzione è una funzione aziendale che comporta dei costi, ma se sfruttata nel modo adeguato può portare a molti vantaggi anche economici. Questo valore aggiunto deriva da tutti quei risparmi sulle

spese che la manutenzione permette di avere andando a prevenire i guasti con conseguente aumento della disponibilità dei macchinari e di conseguenza un incremento del volume produttivo del sistema industriale.

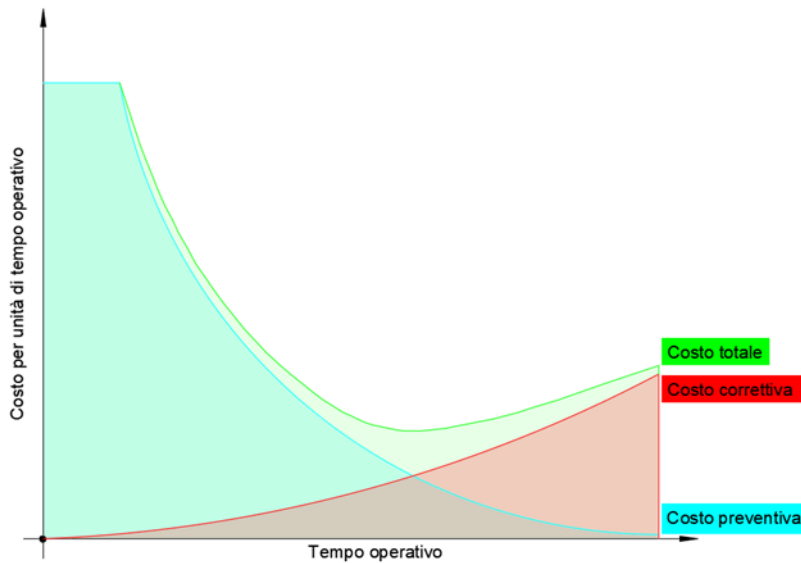


Figura 1.3 Grafico dei costi di manutenzione preventiva e correttiva

Nei costi manutentivi rientrano tutte quelle spese legate agli strumenti per la manutenzione fisica (componenti di ricambio, strumenti, personale etc.) ma anche tutti quei costi che essa produce in termini di servizi erogati.

In particolare, è importante differenziare tra i costi legati all'attività manutentiva ordinaria, cioè tutte quelle spese di manutenzione legate agli interventi previsti dal piano manutentivo, e i costi legati alla manutenzione straordinaria, cioè le spese dovute a tutte quelle attività manutentive non direttamente comprese nel programma di manutenzione, che comportano spesso degli investimenti la cui importanza deriva appunto dalla tipologia di intervento che viene fatto sul componente. Inoltre, è di fondamentale importanza suddividere ulteriormente i costi di manutenzione ordinaria in costi propri, ovvero quelli dovuti al costo delle risorse utilizzate, e in costi indotti, cioè le spese legate al servizio mancante a causa del guasto. Nei costi propri a questo punto si possono definire i costi diretti ed i costi indiretti. (Furlanetto et al., 2006)

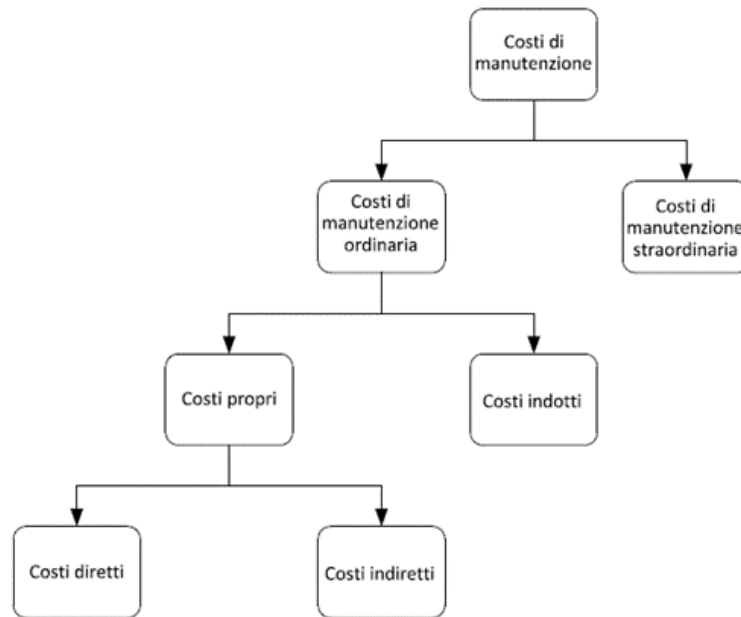


Figura 1.4 Schema a blocchi dei costi della manutenzione

#### 1.1.4.1 Costi propri

I costi propri comprendono tutti i costi direttamente associati all'esecuzione delle attività manutentive, come la manodopera e i materiali, e comprendono inoltre anche i costi indiretti organizzativi e di funzionamento, come ad esempio i costi delle strutture legate alla manutenzione, i costi delle attrezzature e dei servizi utilizzati.

I costi propri possono essere divisi in due sottocategorie: costi propri diretti e costi propri indiretti. I costi propri diretti sono legati alle risorse impegnate per le attività di manutenzione e comprendono:

- Costi di manodopera aziendale diretta: sono tutte quelle spese legate al personale manutentivo direttamente impegnato negli interventi (saldatori, elettricisti, meccanici etc.).
- Costi dei materiali: sono i costi dei materiali prelevati a scorte e di acquisto diretto quando vengono reperiti da fornitori esterni.
- Costi legati alle prestazioni di terzi: si intendono le spese legate ad interventi manutentivi fatte da aziende esterne su un macchinario del sistema.

I costi propri indiretti sono determinati dalle seguenti voci:

- Costi della struttura di manutenzione, cioè del personale indiretto che è legato all'organizzazione del servizio di manutenzione.

- Costi dei servizi tecnici e delle attrezzature di funzionamento necessari per l'operatività del servizio di manutenzione.
- Costi di immobilizzo dei materiali di ricambio e di consumo diretto a disposizione del servizio di manutenzione che comportano degli oneri finanziari.
- Costo dei servizi ausiliari utilizzati.

(Furlanetto et al., 2006)

### 1.1.4.2 Costi indotti

I costi indotti nascono come conseguenza dell'interruzione del funzionamento di un dato apparato, dovuta ad un guasto o ad un fermo volontario per effettuare delle attività di manutenzione. Nel caso di fermo volontario dell'apparato ci si trova in un caso di manutenzione programmata. Un'attività di manutenzione preventiva ben programmata è una potenziale fonte di costi indotti perché può necessitare di un fermo impianto, ma è di fondamentale importanza nel ridurre la probabilità di guasto e quindi nel ridurre il numero di interruzioni per guasto che potrebbero determinare dei costi indotti molto più elevati di un fermo programmato. Principalmente esistono tre cause che determinano i costi indotti: manutenzione ritardata, manutenzione inadeguata e conduzione inadeguata.

I costi indotti possono essere trattati anche a livello più generale. Si possono considerare indotti tutti quei costi legati a:

- Mancato utilizzo dell'entità, cioè quando durante il fermo impianto a seguito di un guasto non è possibile utilizzare la capacità di servizio della macchina oppure quando abbiamo del personale che dovrebbe lavorare alla macchina ma non può perché in quel momento non è essa stessa in funzione.
- Costi legati alla qualità del prodotto, ovvero i costi legati al prodotto lavorato durante l'attività manutentiva svolta in corsa, quindi a impianto funzionante ma che necessariamente produrrà un lavorato di qualità non conforme alla vendita al cliente, che quindi dovrà essere scartato o rigenerato.
- Costi legati alla riduzione di efficienza del sistema, quindi tutte quelle spese di energia elettrica, acqua, gas e metano utilizzati in più a causa del degradamento del sistema.
- Costi legati alle spese extra in caso di incidente, in questo settore rientrano tutte quelle spese assicurative, ad esempio l'aumento del premio da pagare all'assicurazione a seguito dei danni

causati dall'entità, oppure danni ambientali, infortuni sul lavoro, danni di immagine, costi legali o anche spese per la ricerca di un sistema che non comporti danni così gravi all'azienda.

(Furlanetto et al., 2006)

## 1.2 - Stato dell'arte delle tecniche di manutenzione predittiva

Nonostante l'elevato costo di investimento in ICT (Information and Communication Technology) e sensoristica necessari allo sviluppo di un sistema di manutenzione predittiva, questa risulta la tecnica di manutenzione preventiva più appropriata nell'89% dei modi di guasto (Hashemian, 2010).

L'interesse degli studiosi verso questa politica manutentiva innovativa è sempre maggiore anno dopo anno fin dal primo articolo pubblicato nel 1952, come mostrato dal grafico in Figura 1.5:

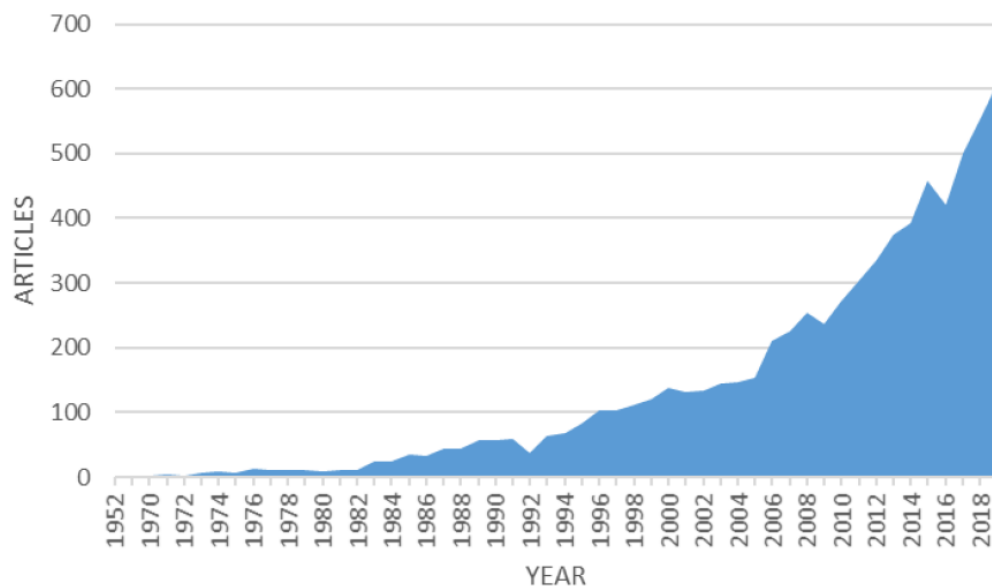


Figura 1.5 Numero di articoli pubblicati per anno (Keywords: Predictive Maintenance, Limits: Articles Fonte: Scopus.com)

Per una maggiore comprensione sull'andamento storico degli articoli riguardanti la manutenzione predittiva viene riportata una breve analisi suddivisa in tre periodi: 1990-2000, 2000-2008, 2008-2018. (Hoppenstedt et al., 2018)

### Periodo 1990-2000

In questo primo periodo l'analisi si basa su 175 articoli e il primo articolo riportato è a cura di Special Interest in Manufacturing (SIGMAN), un'associazione americana sull'intelligenza artificiale. In questo

periodo gli articoli più pubblicati provengono da: Stati Uniti (29.7%), Francia (8.0%) e Canada (7.4%). In tale arco temporale non viene evidenziato un argomento di ricerca che spicchi sugli altri ed è presente una forte connessione tra la predictive maintenance e il monitoring.

### **Periodo 2000-2008**

In questo periodo l'analisi si basa su 574 articoli con gli Stati Uniti come paese più attivo sotto il profilo della ricerca, subito seguiti dalla Cina che compare per la prima volta con la pubblicazione di 54 articoli, mentre tra i paesi più industrializzati i meno partecipativi risultano essere Giappone e Germania. In questo periodo gli argomenti più analizzati risultano essere i predictive models, i cost-based models e, in generale, gli aspetti manageriali beneficiano di un interesse crescente.

### **Periodo 2008-2016**

Anche in questo periodo gli Stati Uniti rimangono leader nella ricerca subito seguiti dalla Cina con l'istituto di Pechino Tsinghua come forza trainante. Tra gli argomenti maggiormente trattati si trova il multi-component applications in cui si vuole combinare dati provenienti da componenti diversi in un unico modello predittivo. Grande interesse viene dato alle turbine eoliche, ai cuscinetti rotanti e in generale alla manutenzione ferroviaria e alla corrosione dei metalli.

Per quel che concerne i contributi di questa prima fase di ricerca bibliografica, i maggiori sono:

- Una panoramica sulla manutenzione predittiva, attraverso un'approfondita ricerca bibliografica, andando, in primis, ad individuare i paesi maggiormente interessati nel finanziare questo studio. Successivamente, la ricerca si è concentrata sull'individuazione dei settori, dei componenti e dei loro parametri più interessati alla trattazione;
- Approfondimento sulle tecniche più frequenti di previsione della RUL del componente industriale più citato;
- Uno schema generale che mostri il modus operandi per la previsione della RUL così che possa essere utilizzato da chiunque si approcci ad un'analisi di predizione.

### **1.2.1 - Raccolta dei dati**

La raccolta del materiale impiegato per lo studio trattato è avvenuta in due fasi:

- scelta del database;

- selezione degli articoli tramite inserimento di keywords e lettura abstracts.

La scelta dei databases è ricaduta su Scopus Preview e IEEE Xplore Digital Library.

Il primo utilizzato è stato Scopus Preview: la prima keyword inserita è stata predictive maintenance con la quale è stato trovato un numero ingente di articoli, per questo è stato necessario inserire le key-words Industrial e Prognostic. Per snellire ulteriormente la ricerca sono stati aggiunti la parola chiave RUL e di seguito un filtro temporale che copre l'intervallo dal 2010 al 2019; si è scelto tale intervallo per trovare un giusto compromesso tra le ultime ricerche e la necessità di avere un panorama più ampio dell'argomento.

Nella Tabella 1.1 viene illustrata la relazione tra le keywords inserite e il numero di articoli corrispettivi.

**Tabella 1.1 Keywords e numero di articoli Scopus**

KEYWORDS	NUMERO ARTICOLI (Other + Open Access)
Predictive maintenance	11122
Predictive maintenance + Industrial	2383
Predictive maintenance + Industrial + Prognostic	558
Predictive maintenance + Industrial + Prognostic + RUL	76
Predictive maintenance + Industrial + Prognostic + RUL + 2010-2019	76

È possibile osservare che all'aumentare del numero delle keywords diminuisce la quantità di articoli inerenti e ciò è evidente in Figura 1.6 numero di articoli per Keywords.

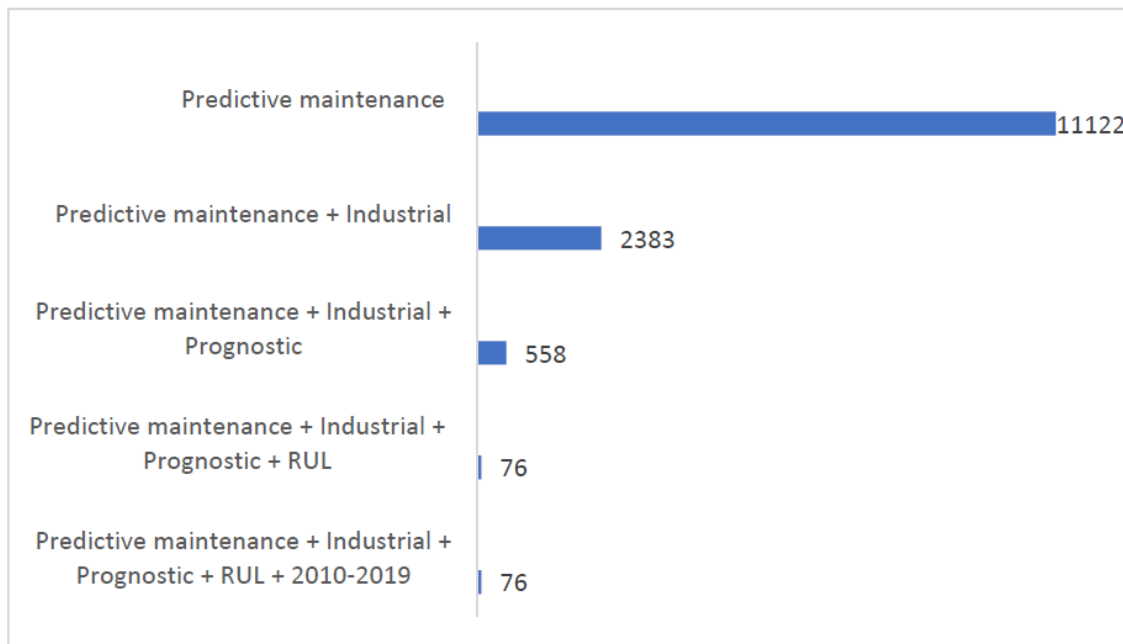


Figura 1.6 numero di articoli per Keywords Scopus

Con IEEE Xplore Digital Library la selezione degli articoli è stata eseguita in maniera simile. Prima di tutto, sono stati scelti solo quelli appartenenti a Journal & Magazines, escludendo di conseguenza i Books e le Conferences. L'ordine di inserimento delle keywords è stato lo stesso del caso di Scopus Preview: il risultato, inserendo solamente Predictive Maintenance, Industrial, Prognostic e RUL è stato di 8 articoli. Successivamente, è stato inserito l'intervallo temporale 2010-2019 ma l'applicazione del filtro non è stata possibile, in quanto gli articoli avevano come anno di pubblicazione almeno il 2016. Questa modalità di ricerca, indicata con Caso1, viene confrontata con Caso2 e Caso3, in cui la sequenza delle keywords è stata leggermente modificata.

Restano invariate Predictive Maintenance e Industrial: nel Caso2 e nel Caso3 vengono rispettivamente sostituiti RUL e Prognostic con il filtro temporale dal 2010 al 2019. In entrambi i casi ciò non è stato possibile dato che l'anno di pubblicazione non partiva dal 2010.

Per tale motivo è stato necessario aggiungere Caso4 che prevede l'utilizzo di Predictive Maintenance, Industrial e solamente il filtro temporale 2010-2019.

Viene illustrata la relazione tra le key-words inserite e il numero di articoli corrispettivi del Caso1 in Tabella 1.2, del Caso2 in Tabella 1.3, del Caso3 in Tabella 1.4 e del Caso4 in Tabella 1.5.



Tabella 1.2 Keywords e numero di articoli IEEE caso 1

KEY-WORDS Caso1	NUMERO ARTICOLI (All Results)
Predictive Maintenance	436
Predictive Maintenance + Industrial	75
Predictive Maintenance + Industrial + Prognostic	18
Predictive Maintenance + Industrial + Prognostic + RUL	8
Predictive Maintenance + Industrial + Prognostic + RUL + 2010-2019	0

Tabella 1.3 Keywords e numero di articoli IEEE caso 2

KEY-WORDS Caso2	NUMERO ARTICOLI (All Results)
Predictive maintenance	436
Predictive maintenance + Industrial	75
Predictive maintenance + Industrial + Prognostic	18
Predictive maintenance + Industrial + Prognostic + 2010-2019	0

Tabella 1.4 Keywords e numero di articoli IEEE caso 3

KEY-WORDS Caso3	NUMERO ARTICOLI (All Results)
Predictive maintenance	436
Predictive maintenance + Industrial	75
Predictive maintenance + Industrial + RUL	9
Predictive maintenance + Industrial + RUL + 2010-2019	0

Tabella 1.5 Keywords e numero di articoli IEEE caso 4

KEY-WORDS Caso4	NUMERO ARTICOLI (All Results)
Predictive maintenance	436
Predictive maintenance + Industrial	75
Predictive maintenance + Industrial + 2010-2019	61

Nella Figura 1.7 vengono confrontati i risultati dei casi in modo da rendere più evidente il motivo per cui è stato scelto come ordine finale di inserimento delle keywords quello del Caso4.

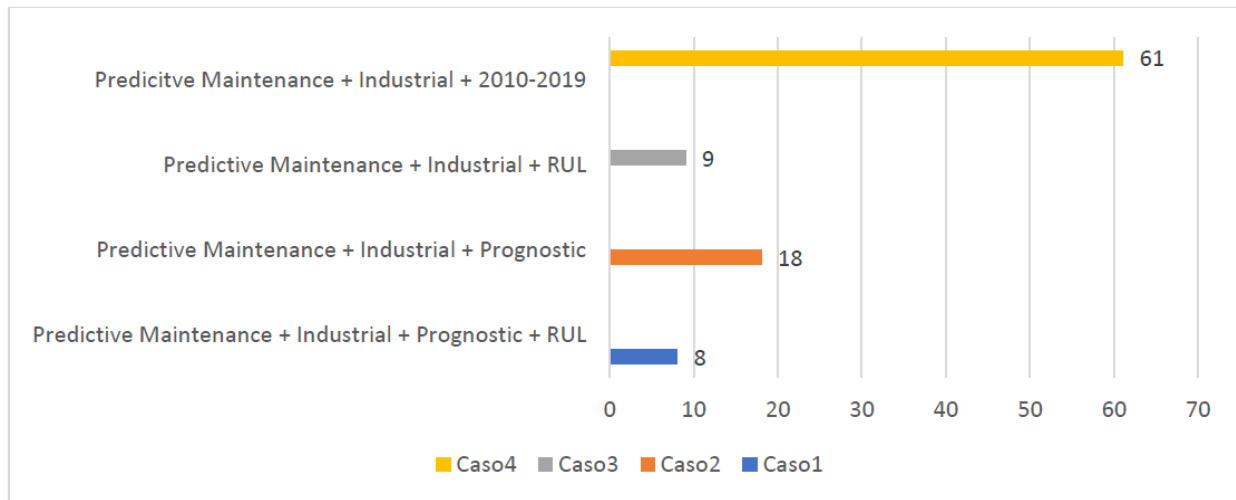


Figura 1.7 numero di articoli per caso IEEE

Per entrambi i databases è stato utilizzato il Software Zotero per la gestione del materiale bibliografico. La lettura degli abstracts è stato il primo metro di valutazione utilizzato per la selezione degli articoli su Zotero; a seguito di una lettura completa di ogni articolo aggiunto, è stata rivalutata l'inerenza del contenuto di alcuni articoli che quindi sono stati eliminati dallo studio.

Sono stati individuati 58 articoli finali di cui 45 da Scopus Preview e 13 da IEEE Xplore Digital Library.

Per Scopus Preview sono stati selezionati come parametri:

- Citazioni;
- Field-weighted citation impact: indica quanto il documento viene citato confrontandolo con documenti simili su una finestra di tre anni. Un punteggio superiore a 1.00 indica che il documento in esame sia più citato della media; si tiene conto di:
  - Anno di pubblicazione
  - Tipo di documento
  - Discipline associate alla ricerca
- CiteScoreTracker 2018: viene definito come il rapporto tra  $\frac{\text{Citation Count 2018}}{\text{Documents 2015-2017}}$
- Anno di pubblicazione.

Di seguito nella Tabella 1.6 Parametri degli articoli scelti vengono riportati gli articoli selezionati con i rispettivi parametri.

Tabella 1.6 Parametri degli articoli scelti su Scopus

SCOPUS	N citazioni	Field-weighted citation impact	CiteScoreTracker 2018	Anno pubblicazione
A Bayesian approach for a damage growth model using sporadically measured and heterogeneous on-site data from a steam turbine (Choi et al., 2019)	0	0	5.39	2019
A Bayesian network model integrated in a prognostics and health management system for aircraft line maintenance (Ferreiro et al., 2011)	8	0.47	1.59	2011
A Comparative Study on Machine Learning Algorithms for Smart Manufacturing: Tool Wear Prediction Using Random Forests (Wu et al., 2017)	39	8.25	3.24	2017
A hybrid prognostics technique for rolling element bearings using adaptive predictive models (Ahmad et al., 2017)	10	2.29	9.66	2017
A Model-Based Method for Remaining Useful Life Prediction of Machinery (Lei et al., 2016)	60	9.86	3.78	2016
A multicriteria model for optimization of maintenance in thermal energy production systems in hospitals: A case study in a Spanish hospital (Carnero and Gómez, 2017)	1	0.23	3.00	2017
A new dynamic predictive maintenance framework using deep learning for failure prognostics	0	0	5.39	2019

(Nguyen and Medjaher, 2019)				
A SAW wireless sensor network platform for industrial predictive maintenance (Gombé et al., 2019)	0	1.08	4.20	2019
A wiener-process-model-based method for remaining useful life prediction considering unit-to-unit variability (Li et al., 2018)	2	8.22	9.66	2018
An ensemble learning-based prognostic approach with degradation-dependent weights for remaining useful life prediction (Li et al., 2019)	5	22.14	5.39	2019
Challenges Building a Data Value Chain to Enable Data-Driven Decisions: A Predictive Maintenance Case in 5G-Enabled Manufacturing (AAkerman et al., 2018)	0	0	1.39	2018
Condition based maintenance in railway transportation systems based on big data streaming analysis (Fumeo et al., 2015)	30	10.36	1.48	2015
Condition-based predictive order model for a mechanical component following inverse Gaussian degradation process (C. Wang et al., 2018)	0	0	1.13	2018
Cost-oriented predictive maintenance based on mission reliability state for cyber manufacturing systems (He et al., 2018)	5	3.59	1.24	2018
Battery health prognosis using brownian motion modeling and	3	2.40	9.66	2018

particle filtering (Dong et al., 2018)				
Data analysis and feature selection for predictive maintenance: A case-study in the metallurgic industry (Fernandes et al., 2019)	0	0	8.75	2019
Data science for vibration heteroscedasticity and predictive maintenance of rotary bearings (Lee et al., 2019)	0	0	3.16	2019
Deep digital maintenance (Rødseth et al., 2017)	1	0.24	2.52	2017
Enhanced Particle Filtering for Bearing Remaining Useful Life Prediction of Wind Turbine Drivetrain (Cheng et al., 2018)	0	0	9.66	2019
Enhanced Restricted Boltzmann Machine With Prognosability Regularization for Prognostics and Health Assessment (Liao et al., 2016)	32	4.79	9.66	2016
Ensemble machine learning and forecasting can achieve 99% uptime for rural handpumps (Wilson et al., 2017)	3	0.85	3.02	2017
Estimating Mean Residual Life for a Case Study of Rail Wagon Bearings (Ghasemi and Hodkiewicz, 2012)	32	2.82	3.78	2012
Evaluation of the effects of a machine failure on the robustness of a job shop system-proactive approaches (Paprocka, 2019)	0	0	3.0	2018
Hidden Markov model with auto-correlated observations for remaining useful life prediction and optimal maintenance policy (Chen et	0	0	5.39	2019

al., 2019)				
Lithium-Ion Battery Prognostics with Hybrid Gaussian Process Function Regression (Peng et al., 2018)	4	2.69	3.16	2018
Machine and component residual life estimation through the application of neural networks (Herzog et al., 2009)	34	2.69	5.39	2009
Mechanical fault detection in electric motors measured by a digital signal processing device in an optical mouse (Ribeiro et al., 2019)	0	0	3.33	2019
Multi-branch Hidden semi-Markov modeling for RUL prognosis (Le et al., 2015)	6	3.09	CONFERENCE PAPER	2015
Multiple-Change-Point Modeling and Exact Bayesian Inference of Degradation Signal for Prognostic Improvement (Wen et al., 2018)	0	0	6.10	2018
Optimal state selection and tuning parameters for a degradation model in bearings using Mel-Frequency Cepstral Coefficients and Hidden Markov Chains (Holguín et al., 2016)	0	0	0.36	2016
Optimization of aeroengine shop visit decisions based on remaining useful life and stochastic repair time (Cai et al., 2016)	0	0	1.13	2016
Particle filter-based approach to estimate remaining useful life for predictive maintenance (Yang et al., 2015)	1	0	1.05	2015
Prediction of Railcar Remaining Useful Life by Multiple Data Source	13	1.40	7.98	2015

Fusion (Li and He, 2015)				
Prediction of industrial equipment Remaining Useful Life by fuzzy similarity and belief function theory (Baraldi et al., 2017)	8	2.39	6.35	2017
Prediction of Remaining Useful Lifetime (RUL) of turbofan engine using machine learning (Mathew et al., 2017)	1	3.63	CONFERENCE PAPER	2018
Predictive Maintenance by Risk Sensitive Particle Filtering (Compare and Zio, 2014)	16	1.78	3.78	2014
Predictive Maintenance Optimization for Aircraft Redundant Systems Subjected to Multiple Wear Profiles (Vianna and Yoneyama, 2017)	2	1.65	5.17	2017
Prognostics and Health Management: A Review of Vibration Based Bearing and Gear Health Indicators (Wang et al., 2017)	23	6.13	4.95	2017
Remaining useful life estimation – A review on the statistical data driven approaches (Si et al., 2011)	744	13.86	5.00	2011
Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction (Liao and Köttig, 2014)	90	8.30	3.78	2014
Stochastic hybrid estimator based fault detection and isolation for wind energy conversion systems with unknown fault inputs (Shi et al., 2018)	0	0	3.16	2018
Techniques and emerging trends for	0	0	2.51	2018

state of the art equipment maintenance systems-A bibliometric analysis (Hoppenstedt et al., 2018)				
Testing an innovative predictive management system for bus fleets: outcomes from the Ravenna case study (Corazza et al., 2017)	0	0	2.73	2018
Uncertain generalized remaining useful life prediction-driven predictive maintenance decision (Xu et al., 2015)	0	0	CONFERENCE PAPER	2015
Vehicle remote health monitoring and prognostic maintenance system (Shafi et al., 2018)	3	2.94	2.00	2018

Per IEEE Xplore Digital Library sono stati selezionati come parametri:

- Citazioni;
- Full Text Views;
- Impact Factory: è la media del numero di citazioni in una finestra di due anni dalla pubblicazione;
- Anno di pubblicazione.

Di seguito nella Tabella 1.7 vengono riportati gli articoli selezionati con i rispettivi parametri.

**Tabella 1.7 Parametri degli articoli scelti su IEEE**

IEEE	N citazioni	Full text Views	Impact Factor	Anno pubblicazione
A Predictive Maintenance System for Epitaxy Processes Based on Filtering and Prediction Techniques (Susto et al., 2012)	17	653	1.336	2012
A Two-Stage Fault Detection and Isolation Platform for industrial Systems Using Residual Evaluation (Heydarzadeh and Nourani, 2016)	18	396	2.794	2016



Connexionist-Systems-Based Long Term Prediction Approaches for Prognostics (Gouriveau and Zerhouni, 2012)	38	585	2.729	2012
Creating Self-Aware Low-Voltage Electromagnetic Coils for Incipient Insulation Degradation Monitoring for Smart Manufacturing (K. Wang et al., 2018)	NON DEFINITO	106	3.557	2018
Direct Remaining Useful Life Estimation Based on Support Vector Regression (Khelif et al., 2016)	33	1502	7.05	2016
Failure Risk Indicators for a Maintenance Model Based on Observable Life of Industrial Components With an Application to Wind Turbines (de Andrade Vieira and Sanz-Bobi, 2013)	19	723	2.729	2013
Forecasting Obsolescence Risk and Product Life Cycle With Machine Learning (Jennings et al., 2016)	4	898	1.66	2016
Industrial Big Data in an Industry 4.0 Environment: Challenges, Schemes, and Applications for Predictive Maintenance (Yan et al., 2017)	9	6569	3.557	2017
Integrated Predictive Analytics and Optimization for Opportunistic Maintenance and Operations in Wind Farms (Yildirim et al., 2017)	NON DEFINITO	931	5.255	2017
K-PdM: KPI-Oriented Machinery Deterioration Estimation Framework for Predictive Maintenance Using Cluster-Based Hidden Markov Model (Wu et al., 2018)	NON DEFINITO	429	3.557	2018
Model Predictive Control-Based Optimal Operations of District Heating System With Thermal Energy Storage and Flexible Loads (Verrilli et al., 2016)	18	1158	3.667	2016

PDF and Breakdown Time Prediction for Unobservable Wear Using Enhanced Particle Filters in Precognitive Maintenance (Pang et al., 2014)	12	274	2.794	2014
State-of-the-Art Predictive Maintenance Techniques (Hashemian, 2010)	55	3533	2.794	2010

Partendo dai parametri individuati nei due databases impiegati per valutare la rilevanza del singolo articolo e della rivista in cui esso è contenuto, è stato individuato il loro valore medio così da avere una panoramica della rilevanza degli articoli selezionati. Successivamente la lettura degli articoli ha portato all'elaborazione di tre differenti risultati: le aree in cui sono presenti i maggiori finanziatori per la ricerca sulla manutenzione predittiva, i settori e i componenti più presenti.

Di seguito vengono riportati i valori medi dei parametri provenienti da Scopus Review (Tabella 1.8) e da IEEE (Tabella 1.9) per valutare complessivamente l'importanza degli articoli selezionati.

**Tabella 1.8** Valor medio dei parametri degli articoli scelti su Scopus

N medio citazioni	Mean field-weighted citation impact	Mean CiteScoreTracker 2018
26,13	2,85	4,07

**Tabella 1.9** Valor medio dei parametri degli articoli scelti su IEEE

N medio citazioni	Mean Full Text Views	Mean Impact Factory
17.15	1365.92	3.34

Andando ad analizzare l'anno di pubblicazione degli articoli selezionati in entrambi i databases, è possibile osservare il crescente interesse per la manutenzione predittiva; nella Figura 1.8 Percentuale per anno degli articoli selezionati viene riportato sull'asse delle ascisse l'anno di pubblicazione e sull'asse delle ordinate la percentuale di articoli per ciascun anno, in proporzione al numero totale di articoli per database.

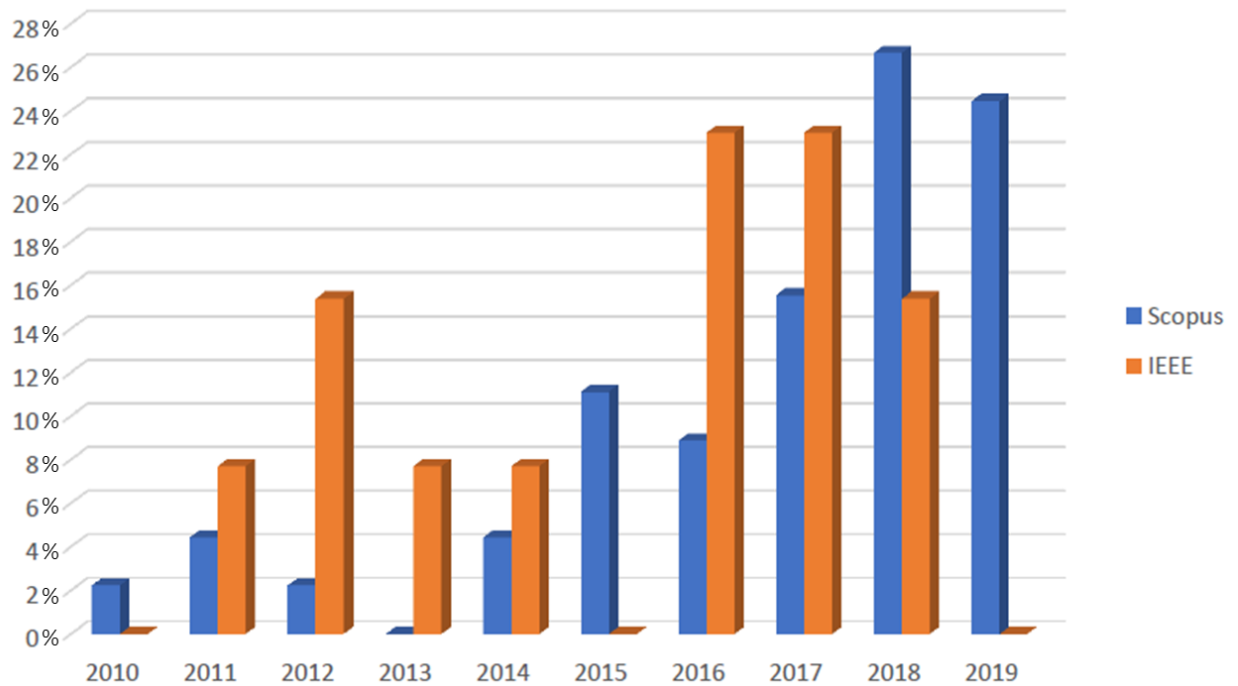


Figura 1.8 Percentuale per anno degli articoli selezionati

In entrambi i casi si nota un maggiore interesse verso l'argomento a partire dal 2016 e, rispettivamente per Scopus e IEEE, gli anni con un ingente numero di pubblicazioni sono 2018 e 2016/2017.

### 1.2.2 - Analisi delle aree geografiche degli articoli

Per capire quale sia il paese più interessato alla predictive maintenance, è stata fatta una ricerca utilizzando come metro di valutazione i finanziatori dei singoli articoli. Occorre specificare come non in tutti gli articoli sia stato possibile individuare i finanziatori che, quindi, risultano essere:

- Nordamerica: U.S. Department of Energy, US National Science Foundation;
- Sudamerica: CAPES, FAPEMIG, CNPq e FINEP in Brasile, Departement o Administrativo de Ciencias in Colombia;
- Asia: National Natural Science Foundation of China, Singapore Ministry of Education through the Academic Research Funding, Toyota Motor Corporation, Ministry of Science and Technology Israel, Korea Electric Power Corporation, Korea Institute of Energy and Technology Evaluation and Planning;
- Oceania: Australian Government Endeavour Post-doctoral Fellowship award;
- Europa: Politecnico della Slesia, Swedish Agency for Innovation Systems, Franco-Swiss INTERREG IV, European Commission.

È stata quindi fatta una percentuale del rapporto tra il numero degli articoli che i singoli finanziatori hanno supportato e il numero totale di articoli analizzati in cui fosse presente un finanziatore. In Figura 1.9 gli Stati sono classificati tramite una scala di colori indicando con il giallo il valore minore del parametro precedentemente definito e con il rosso il maggiore.

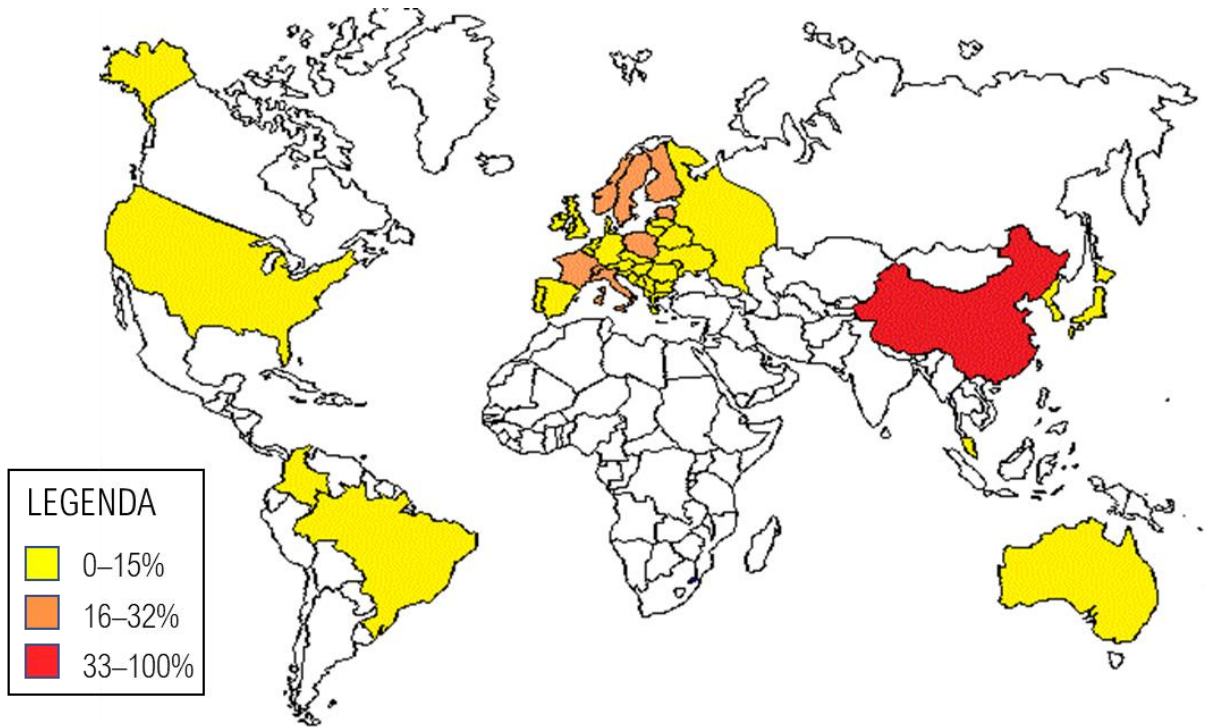


Figura 1.9 Paesi finanziatori degli articoli selezionati

Dalla Figura 1.9 è possibile constatare come la National Natural Science Foundation of China sia il principale finanziatore a livello mondiale.

### 1.2.3 - Analisi dei settori industriali degli articoli selezionati

Successivamente sono stati analizzati i settori industriali maggiormente coinvolti nella ricerca sulla manutenzione predittiva e rappresentati in Figura 1.10:

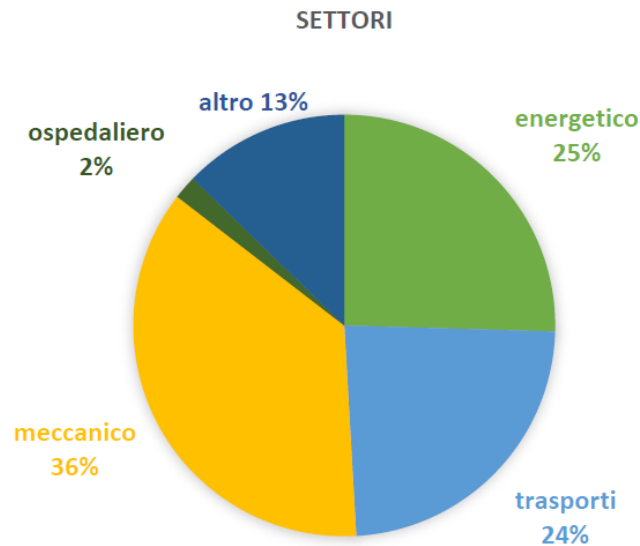


Figura 1.10 Settori industriali degli articoli selezionati

In particolare, il settore:

- trasporti: comprende quello aeronautico, ferroviario e automobilistico;
- meccanico: raccoglie tutti i componenti di natura meccanica di cui non viene specificata la macchina di appartenenza;
- altro: contiene gli articoli bibliografici.

Come si può osservare, il settore meccanico è il più interessato con il 36% seguito dai trasporti e dal settore energetico.

#### 1.2.4 - Analisi dei componenti trattati negli articoli selezionati

Dalle ricerche condotte per questo studio si è deciso di considerare un intervallo temporale che copre dal 2010 al 2019 ed anche in tale periodo l'analisi svolta ha mostrato un interesse verso gli argomenti trattati nel precedente paragrafo con una maggiore concentrazione verso cuscinetti, turbine e motori. Infatti, come si evince dalla Figura 1.11 sui componenti trattati negli articoli, i cuscinetti rappresentano il 35%, le turboventole il 20% e le turbine 9%.

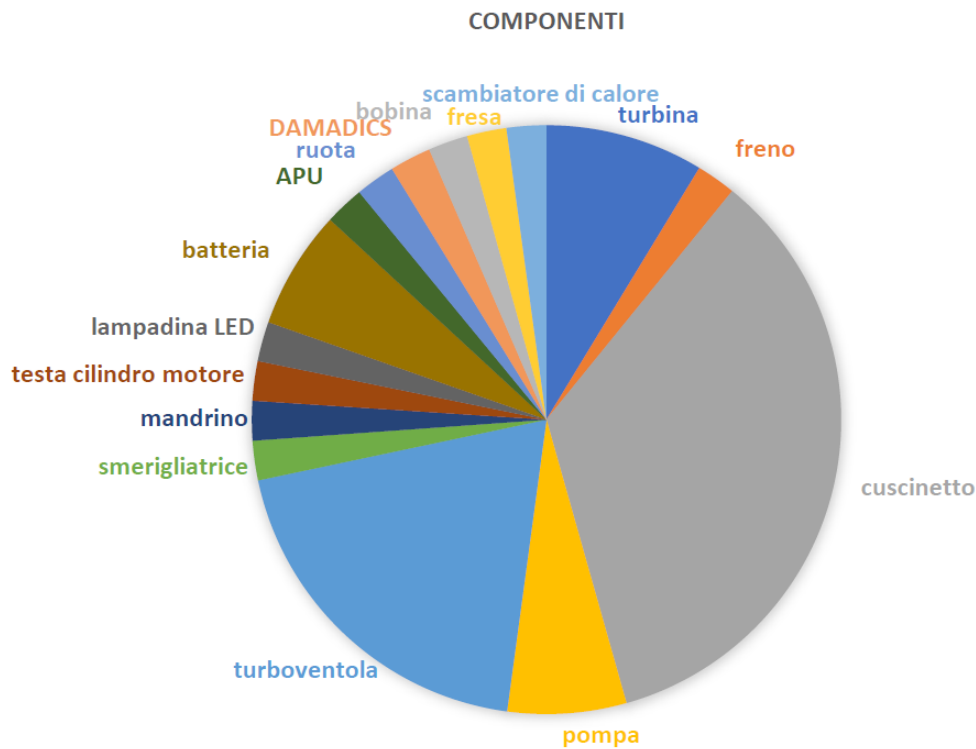


Figura 1.11 Componenti trattati negli articoli selezionati

### 1.2.5 - Analisi delle tecniche di stima della RUL

La manutenzione predittiva si occupa di monitorare macchinari e operazioni in modo continuo cercando di prevedere futuri guasti (Lee et al., 2019); l'integrazione tra particolari sensori e tecniche di manutenzione predittiva permette di evitare la sostituzione non necessaria delle apparecchiature, di ridurre i costi e di migliorare la sicurezza, la disponibilità e l'efficienza dei processi. (Hashemian, 2010) È stato studiato come la manutenzione predittiva sia l'opzione più valida per l'89% dei casi mentre per il restante 11% sia più prudente una manutenzione basata sul tempo (Hashemian, 2010). Infatti, la *bath tub curve*, utilizzata nella manutenzione basata sul tempo, non può essere impiegata come modello dominante per il degrado e il guasto delle apparecchiature in quanto la sua precisione è solo del 4%. Seguendo la *bath tub curve*, le apparecchiature vengono sostituite periodicamente per evitare di raggiungere la fine del ciclo di vita non tenendo conto che se un componente venisse sostituito prematuramente, potrebbe verificarsi la mortalità infantile. Data l'inaffidabilità dei metodi di manutenzione basata sul tempo, nei processi industriali dovrebbe essere utilizzata la manutenzione predittiva non solo quando le apparecchiature sono obsolete ma anche per tutta la loro durata così da identificarne l'inizio dello stato di degradazione e dell'eventuale guasto (Hashemian, 2010). La manutenzione predittiva è composta da due fasi: diagnosi (*Diagnostics*) e prognosi (*Prognostics*); la prima è il processo di rilevamento e isolamento di guasti, mentre la seconda è il processo di previsione dello

stato futuro o della vita utile rimanente (remaining useful life, RUL) in base alle condizioni attuali o storiche.

I modelli di previsione RUL in letteratura possono essere classificati, come mostrato in Figura 1.12, (Liao and Köttig, 2014), in:

A. experience-based models:

I modelli basati sull'esperienza mettono in correlazione le conoscenze specialistiche e l'esperienza ingegneristica per desumere la RUL da misurazioni o eventi storici; tali modelli, tuttavia, tendono a basarsi troppo sulla capacità dell'esperto di indicare le regole del sistema.

B. data-driven models:

I modelli basati sui dati sono composti da dati precedentemente osservati per prevedere il futuro stato del sistema e tra questi si trovano modelli statistici, funzioni di affidabilità e modelli di intelligenza artificiale. Trattandosi di dati storici tali modelli sono applicabili solo quando si ha a disposizione una sufficiente quantità di dati. L'output generato tende a fornire informazioni più precise rispetto a quelle ottenute dall'esperienza, tuttavia, la maggior parte dei risultati di tali modelli è difficile da correlare ad un significato fisico.

C. physics-based models:

I modelli basati sulla fisica comportano una descrizione matematica del processo di degrado del sistema per stimare la RUL. Nonostante l'output di tali modelli tenda ad avere un significato fisico diretto, è richiesta un'ampia quantità di dati empirici e, inoltre, i modelli sono costruiti caso per caso non permettendo l'applicabilità a sistemi diversi.

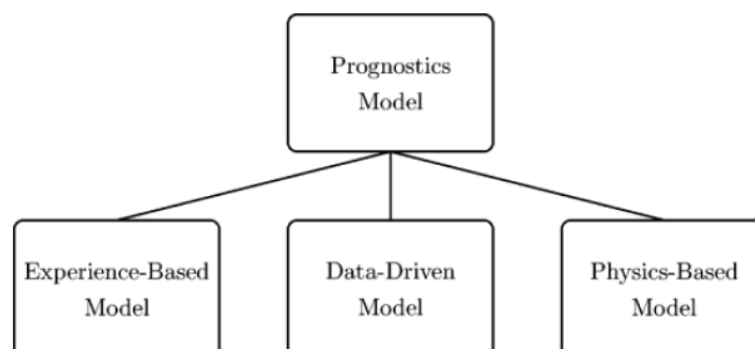


Figura 1.12 Classificazione modelli prognostici (Liao and Köttig, 2014)

Un'ulteriore suddivisione dei metodi di previsione della RUL, sulla quale si basa questo studio, viene riportata in Figura8 (Lee et al., 2019).

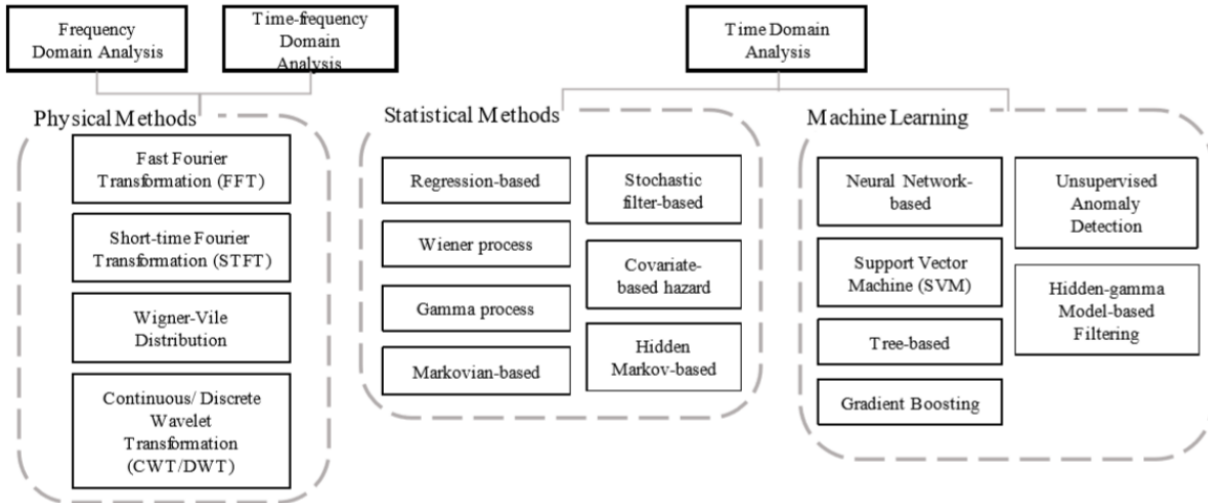


Figura 1.13 Classificazione modelli prognostici (Lee et al., 2019)

I metodi fisici non sono stati presi in considerazione in questa analisi preliminare della letteratura di settore in quanto, negli articoli trattati, vengono utilizzati nella fase iniziale dei metodi statistici: nel (Peng et al., 2018) per eliminare il rumore dai dati di una batteria a litio, nel (Ferreiro et al., 2011) per la rivelazione dell'usura di un freno e nel (Lee et al., 2019) per trasformare il dominio del tempo in dominio di frequenza.

Come precedentemente spiegato, per studiare i modelli di previsione della RUL c'è un forte bisogno di dati; per lo studio dei guasti si deve stabilire se il difetto procura un segnale percepibile e, se il segnale è monitorabile con guasti causati con eventi non poissoniani, è possibile attuare la manutenzione predittiva.

Volendo individuare il numero di articoli facenti parte della metodologia statistica e machine learning, viene di seguito riportata la Figura 1.14 con le percentuali dei risultati trovati. Il metodo statistico è il più utilizzato nel 60% dei casi contro il 40% in cui viene impiegato il machine learning.



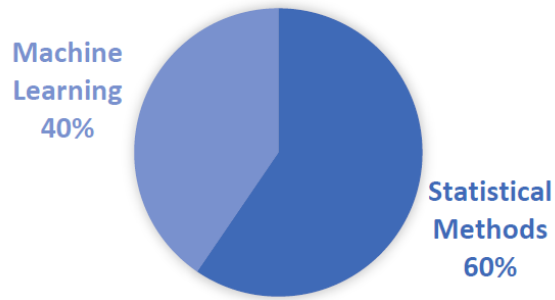


Figura 1.14 Modelli prognostici negli articoli selezionati

Un'ulteriore analisi è stata effettuata sulle variabili considerate per i case study descritti negli articoli: dalla Figura 1.15 è possibile concludere che vibrazione e temperatura sono i parametri maggiormente usati, seguiti da pressione, usura e velocità.

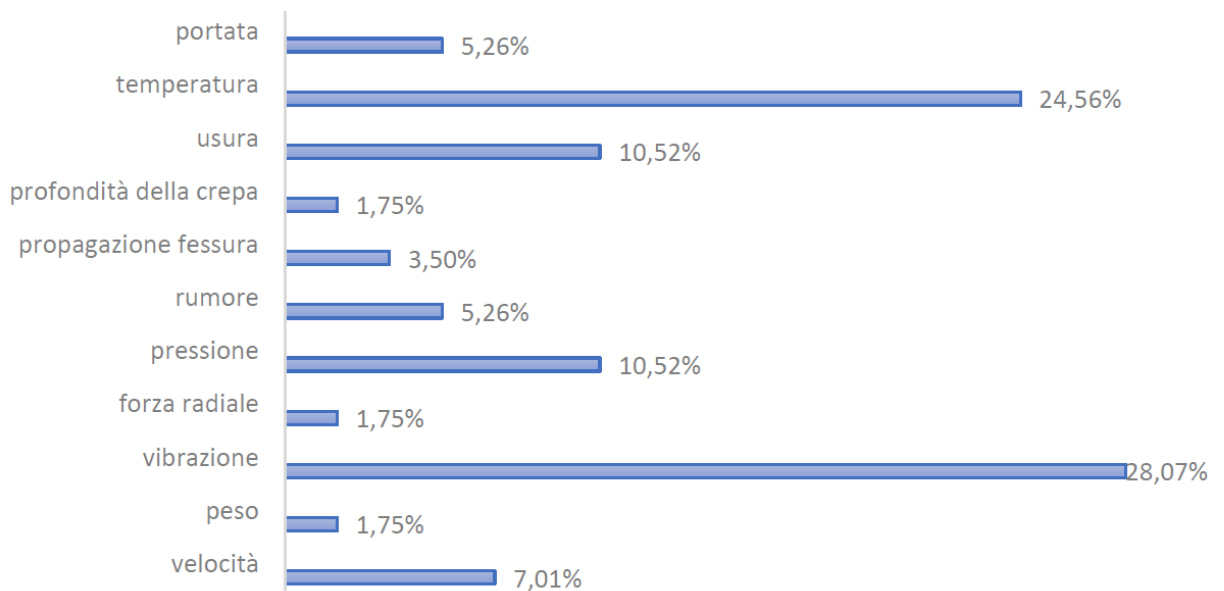


Figura 1.15 Variabili considerate per la stima della RUL

Di seguito vengono riportate delle analisi incrociate componente-variabile (Tabella 1.10) e componente-metodo (Tabella 1.11).

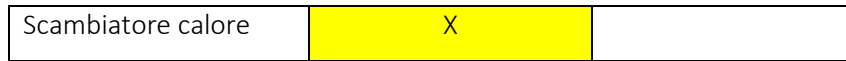
Tabella 1.10 Analisi componente analizzato – variabili selezionate

	Usura	Portata	Temperatura	Propagazione fessura	Profondità crepa	rumore	pressione	Forza radiale	vibrazione	peso	velocità
Turbina			X						X		X
Freno	X									X	X
Cuscinetto			X	X					X		

Pompa							X		X		
Turboventola			X	X			X				
Smerigliatrice			X						X		
Mandrino			X					X			X
Testa cilindro motore	X										
Lampada LED	X										
Batteria	X										
motore			X			X					
APU			X				X				
Ruota						X					
DAMADICS		X									
Bobina			X				X		X		
Fresa	X										
Scambiatore calore		X	X				X				

**Tabella 1.11**      **Analisi componente analizzato – metodo scelto**

	Statistical methods	Machine Learning
Turbina	X	X
Freno	X	
Cuscinetto	X	X
Pompa	X	X
Turboventola	X	X
Smerigliatrice		X
Mandrino	X	
Testa cilindro motore	X	
Lampada LED	X	
Batteria	X	
motore	X	
APU	X	
Ruota		X
DAMADICS		X
Bobina		X
Fresa	X	



Una volta individuato il cuscinetto come componente più analizzato negli articoli trovati, il passo successivo consiste nel raggruppare le tecniche descritte nelle tre metodologie. Dalla Figura 1.16 si osserva nuovamente come il metodo statistico sia largamente il più utilizzato nel 70% dei casi.

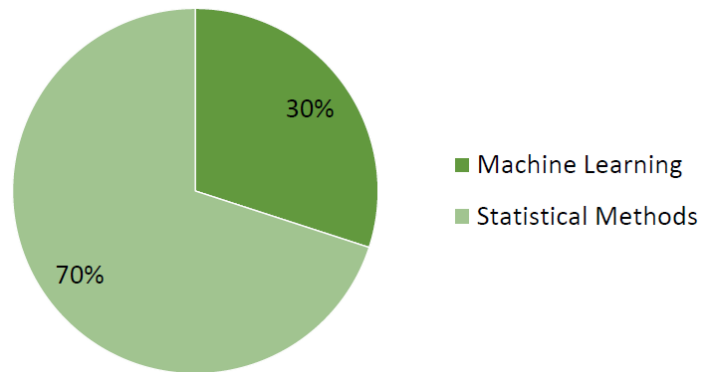


Figura 1.16 Percentuale dei metodi di stima della RUL utilizzati per il componente cuscinetto

Volendo fare una descrizione più accurata della precedente figura relativa esclusivamente ai cuscinetti, vengono riportate in Figura 1.17 le tecniche analizzate negli articoli, suddivise in base al metodo a cui appartengono.

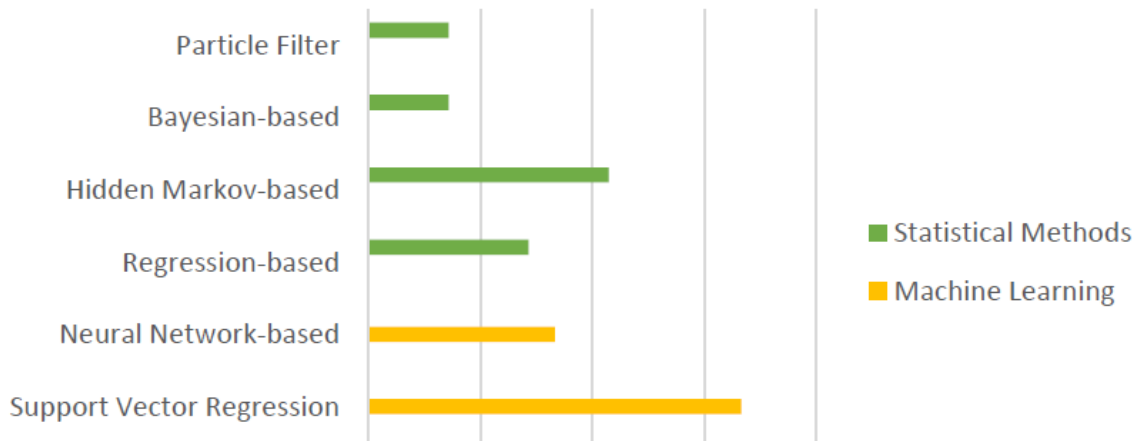


Figura 1.17 Tecniche di stime della RUL degli articoli selezionati

Nei prossimi capitoli verranno analizzate due tecniche per ciascun metodo; il criterio di scelta si è basato sul precedente grafico, prendendo in considerazione solamente le tecniche maggiormente presenti negli articoli riguardanti i cuscinetti.

Il prossimo capitolo sarà incentrato sul metodo statistico, del quale verranno analizzati i metodi di markov e di regressione facendo riferimento agli articoli ad essi associati. Successivamente, un’analoga procedura verrà eseguita per il machine learning del quale sono state rilevate solo due tecniche quali support vector regression e reti neurali.

### 1.2.5.1 Metodi statistici

I metodi statistici vengono suddivisi in (Si et al., 2011):

- modelli basati sui dati diretti provenienti dal monitoraggio condizionale (direct Conditional Monitoring data): Regression-based, Wiener processes, Gamma processes, Markovian-based models;
- modelli basati sui dati indiretti provenienti dal monitoraggio condizionale (indirect Conditional Monitoring data): Stochastic filtering-based models, Covariate-based hazard models, Hidden Markov model (HMM), Hidden semi-markov model (HSMM).

La freccia da indirect CM data a direct CM data implica che sia possibile che gli indicatori rappresentino lo stato di salute del sistema direttamente tramite indirect CM data usando tecniche di elaborazione del segnale, come Fourier e trasformazione Wavelet.

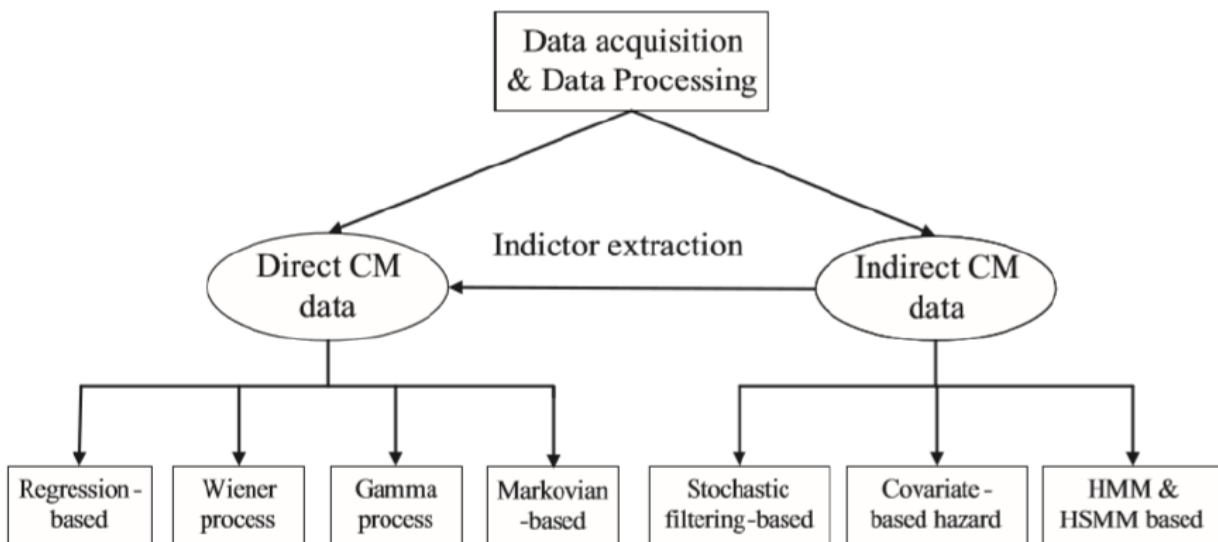


Figura 1.18 Tecniche statistiche di stime della RUL degli articoli selezionati

Nei prossimi paragrafi verranno analizzati i due metodi di previsione della RUL maggiormente presenti negli articoli considerati: metodi markoviano e di regressione.

### 1.2.5.1.1 Modello Markoviano

Quando in un processo stocastico la variabile dipende soltanto dall'evento immediatamente precedente a quello considerato, il processo prende il nome di markoviano, ossia processo senza memoria. Se l'insieme degli eventi  $\{x(t)\}$  è finito o infinitamente numerabile prende il nome di catena di Markov.

In una catena di Markov una sequenza di variabili casuali  $X = (X_1, X_2, \dots)$  ha una distribuzione di probabilità congiunta descritta dalla probabilità condizionata  $P(X_i | X_{i-1}, X_{i-2}, \dots, X_1)$ . Un tipico esempio esplicativo che viene fatto è quello del tempo meteorologico, per cui  $X_i \in L = \{\text{sunny, rainy}\}$ . Il tempo nel giorno  $i$  può essere influenzato dal tempo dei giorni precedenti, ma nella forma più semplice di catena di Markov, la dipendenza del tempo di un giorno è collegata esplicitamente solo al tempo del giorno antecedente. Essa è anche collegata implicitamente, come effetto a catena, a tutti i giorni precedenti. Questa è un'ipotesi di Markov del prim'ordine, per cui:

$$P(X_i | X_{i-1}, X_{i-2}, \dots, X_1) = P(X_i | X_{i-1})$$

La Figura 1.19 rappresenta una catena di markov del prim'ordine, adatta per l'esempio delle previsioni metereologiche (Bellotti and Trotta, n.d.).



Figura 1.19 Esempio di catena markoviana del prim'ordine

Nella pratica affidabilistica si utilizzano sovente le catene di Markov a tempo discreto in cui la variabile tempo viene incrementata con intervalli finiti. Per la rappresentazione grafica delle catene di Markov si utilizza il diagramma degli stati che si compone essenzialmente di:

- circonferenze che rappresentano gli stati del componente;
- archi orientati a cui sono associati i tassi di transizione che rappresentano la transizione tra gli stati.

Di seguito si riportano i due esempi base di diagramma di stato per un componente non riparabile (Figura 1.20) e per uno riparabile (Figura 1.21). Convenzionalmente si indica lo stato di funzionamento con 0=OK e quello di guasto con 1=KO.

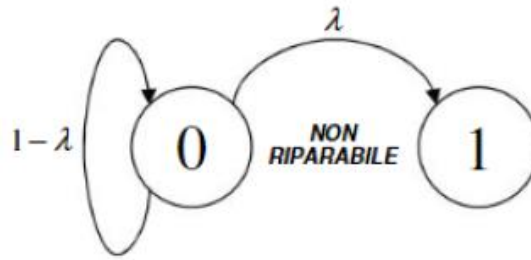


Figura 1.20 Diagramma di stato per componenti non riparabili

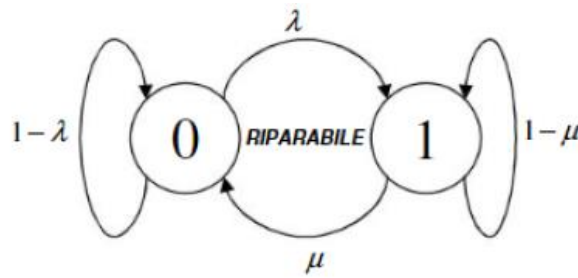


Figura 1.21 Diagramma di stato per componenti riparabili

Si può notare come regola applicativa che la sommatoria di tutti i tassi uscenti da uno stato deve necessariamente essere uguale ad uno.

Si definisce probabilità di transizione ad un passo per una catena di markov a tempo discreto la probabilità che si transiti dallo stato generico  $x_{k-1}$  a quello successivo  $x_k$  durante l'intervallo di tempo  $[t_{k-1}, t_k]$  che viene, appunto, detto passo. Quanto appena definito può essere scritto simbolicamente come segue:

$$\text{Probabilità di transizione ad un passo: } P(x(t_k) = x_k | x(t_{k-1}) = x_{k-1})$$

Si definisce una catena di markov a tempo discreto omogenea quando la relativa probabilità di transizione ad un passo è funzione esclusivamente dell'ampiezza del passo:

Catena di Markov omogenea a tempo discreto:  $P(x(t_k) = x_k | x(t_{k-1}) = x_{k-1}) = f(|t_k - t_{k-1}|)$   
 Solitamente la transizione ad un passo tra gli stati generici  $i$  e  $j$  si indica con la scrittura  $P_{i,j}(\Delta t)$  o più brevemente  $P_{i,j}$ . La matrice composta da elementi  $P_{i,j}$  è detta matrice di transizione che è uguale alla matrice quadrata di ordine  $m+1$  che rappresenta il numero dei possibili stati.

$$[P] = \begin{bmatrix} P_{00} & \dots & P_{0m} \\ \dots & \dots & \dots \\ P_{m0} & \dots & P_{mm} \end{bmatrix}$$

Si nota che tale matrice ha come proprietà fondamentale la relazione:

$$\sum_{j=0}^m P_{i,j} = 1 \quad \forall i = 0,1,2, \dots, m$$

ossia la somma degli elementi di ciascuna riga è sempre uguale a 1.

### 1.2.5.1.2 Hidden Markov model

L'Hidden Markov Model (HMM) è una catena di Markov in cui non sono osservabili tutti gli stati in quanto alcuni sono nascosti. Quindi la differenza sostanziale tra una catena di Markov e un hidden Markov model è che per gli HMM non c'è una corrispondenza one-to-one tra gli stati e le osservazioni. Non c'è possibilità di dire in quale stato si trova il modello osservando solamente gli output della catena. Quindi si differenziano gli HMM con la catena di Markov per l'introduzione di un processo di osservazione. Di seguito viene riportata una tipica struttura di Markov (Figura 1.22): partendo dalle condizioni iniziali a priori, si ha un processo di hidden Markov che è necessariamente formato da una catena markoviana seguita da un processo di osservazione; a seguito di queste due fasi si avrà l'output del processo (Candy, 2016).

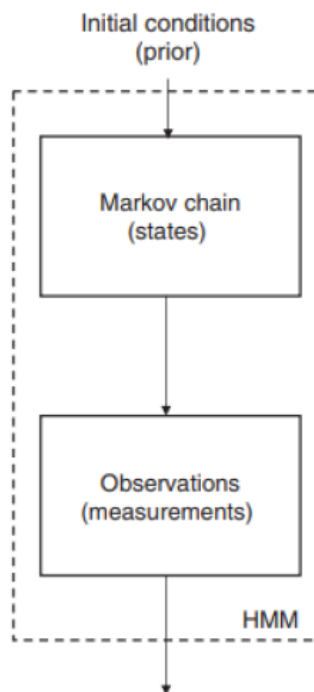


Figura 1.22 Tipica struttura di Markov

Il modello hidden Markov è composto da due processi stocastici: la catena hidden Markov  $\{Z_n, n \geq 0\}$  che è inosservabile e rappresenta lo stato reale del deterioramento, e un processo osservabile  $\{Y_n, n \geq 0\}$  che è

l'informazione di monitoraggio condizionato osservata attraverso il monitoraggio e i test. Si assume che il processo di degradazione  $Z_n$  evolva secondo una catena markoviana su uno stato dello spazio finito. Generalmente, si usa una probabilità condizionata  $P(Y_n | Z_n = i), i \in \Phi \setminus \{N\}$  per rappresentare la relazione tra  $Z_n$  e  $Y_n$ . Utilizzando la tecnica HMM, il RUL può essere definito come  $X_n = \inf\{x_n : Z_n + x_n = N | Z_n \neq N, Y_j, 0 \leq j \leq n\}$  (Si et al., 2011).

Negli HMM la sequenza di stato non è nota ed è infatti nascosta nella sequenza di osservazione. Quindi ad ogni passo temporale  $t$  il sistema genera un'osservazione  $y(t)$  in accordo con la distribuzione di probabilità che dipende dallo stato  $x(t)$ . Il numero di osservazioni  $N_y$  corrisponde ad un noto set discreto che è, al tempo  $t$ , lo stato di osservazione  $y(t) \in Y$  con  $Y = \{Y_1, \dots, Y_{N_y}\}$ . Definiamo la corrispondente probabilità di osservazione discreta con:

$$C_{kl}(t, t) := Pr(y_l(t) | X_k(t)) \text{ con } c_{kl}(t, t) \rightarrow C_{kl} \text{ e } C_{kl} \geq 0 \text{ e } N_y = 1 \text{ } C_{kl} = 1.$$

Si ha la matrice di osservazione di probabilità data da:

$$C = [c_k] \text{ per } k = 1, \dots, N_x; i = 1, \dots, N_y \text{ con } C \in R^{N_y \times N_x}.$$

L'ingrediente finale che caratterizza l'HMM è  $Pr(x(0) = X_i(0)); i = 1, \dots, N_x$  che rappresenta la probabilità iniziale della catena. Quindi, per concludere, un Hidden Markov Model può essere riassunto con un modello:  $\Sigma = \{A, C, P_r(x_i(0))\}$  (caso omogeneo) in cui si ha:

- La matrice di probabilità di transizione dello stato:  $A = [a_{mn}] = Pr(x_n(t) | x_m(t - 1));$

$$m, n = 1, \dots, N_x;$$

- La matrice di probabilità di osservazione:  $C = [c_k] = Pr(y_l(t) | x_k(t)) \text{ per } k = 1, \dots, N_x; = 1, \dots, N_y;$

- La probabilità precedente:  $Pr(x_i(0)); i = 1, \dots, N_x.$

$N_x, N_y$  sono il numero di stati e le osservazioni. Occorre notare che al contrario dei sistemi fisici dinamici dove gli stati e le misure possono assumere qualunque valore reale, gli stati e le osservazioni nei HMM possono solo assumere valori predefiniti interi,  $X = \{X_1, \dots, X_n\}$  e  $Y = \{Y_1, \dots, Y_n\}$ . Va notato come con l'aggiunta di un processo di osservazione, è semplice definire un modello dello spazio degli stati HMM come:

$$x(t) = Ax(t - 1) + w(t - 1)$$



$$y(t) = Cx(t) + v(t)$$

dove  $w$ ,  $v$  sono sequenze non correlate,  $x$  e  $y$  sono rispettivamente gli stati normali e le sequenze osservabili,  $x(0)$  è l'associata condizione iniziale (Candy, 2016).

Vengono adesso analizzati i principali limiti dei modelli hidden Markov; il modello ha un potere limitato nel modellare le strutture temporali nei problemi di predizione a causa dell'indipendenza dagli eventi passati dovuti alla proprietà di markov. Ulteriore limitazione è che solo media e varianza della RUL possono essere stimate non completamente ed è difficile ottenere una distribuzione esplicita della RUL in una forma chiusa (Si et al., 2011). Si vanno ora ad analizzare le proprietà dell'HMM. L'HMM è un modello probabilistico della raccolta congiunta di variabili casuali  $(Y_t, X_t)$ . Le proprietà critiche dell'HMM si basano su metodologie e sviluppi bayesiani. Le nozioni utili ereditate dalla teoria della catena di Markov sono le due proprietà dell'indipendenza condizionale che vengono usate più volte insieme alla regola di Bayes:

1. Le variabili nascoste sono Markov di primo ordine (Markov prim'ordine):  $Pr(x(t)|X_{t-1}, Y_{t-1}) = Pr(x(t)|x(t-1))$  (transizione di stato);
2. L'osservazione è indipendente dalle altre variabili dato lo stato (al tempo  $t$ ):  $Pr(y(t)|Y_{t-1}, X_t) = Pr(y(t)|x(t))$  (probabilità) (Candy, 2016).

### 1.2.5.1.3 Hidden semi Markov model

Una generalizzazione dell'HMM è il modello Hidden Semi Markov (HSMM) che si utilizza in molte applicazioni ingegneristiche e il cui processo di stato inosservato è una catena semi-markov (Si et al., 2011). Infatti, nei modelli HMM standard, a causa dell'assunzione Markoviana, il tempo trascorso in uno stato è o geometrico distribuito, nel caso temporale discreto, oppure esponenziale distribuito, nel caso temporale continuo, ma poiché la maggior parte dei sistemi reali non esibisce tale proprietà si utilizza, in sostituzione, il processo semi-markov (Le et al., 2015). Nonostante i risultati da HSMM siano promettenti, questo possiede lo stesso difetto dell'HMM nell'assumere che le osservazioni siano indipendenti dal tempo. Inoltre, un altro grande limite è che tali metodi, come per gli HMM, possono solo predire il RUL e la sua varianza e non possono fornire la funzione di densità di probabilità (probability density function, PDF) del RUL. Nonostante l'HSMM sia più potente dell'HMM per stimare il RUL, l'HSMM porta a maggiori problemi di identificazione dei parametri complessi (Si et al., 2011). A differenza dell'HMM standard, il tempo di permanenza dello stato in HSMM può seguire qualsiasi distribuzione arbitraria, come la Gaussiana. Una volta inserito uno stato, il processo rimane in tale stato per un periodo di tempo la cui lunghezza è determinata dalla distribuzione arbitraria. Alla fine di questo periodo, il processo si sposta casualmente in un altro stato in base alla matrice di probabilità di transizione dello stato. Si assume che il

processo semi-markov nell'HSMM sia rigorosamente sinistro-destro e che le osservazioni siano distribuzioni gaussiane. Analogamente all'HMM standard, un HSMM può essere caratterizzato dai seguenti parametri: la distribuzione dello stato iniziale  $p$ , il modello di transizione  $A$ , la distribuzione della durata dello stato indicata da  $D$  e il modello di osservazione  $B$ . Quindi, un HSMM può essere scritto come:  $\lambda = (p, A, D, B)$  (Le et al., 2015).

#### 1.2.5.1.4 Applicazioni

Nei seguenti articoli viene utilizzata una catena hidden markov:

1. Multi-branch Hidden semi-Markov modeling for RUL prognosis (Le et al., 2015);
2. Optimal state selection and tuning parameters for a degradation model in bearings using Mel-Frequency Cepstral Coefficients and Hidden Markov Chains (Holguín et al., 2016).

1. Viene analizzato il modello semi-hidden markov in una variazione multi-branch per ovviare al fatto che gli HSMM siano adatti a modellare solo una modalità di deterioramento alla volta. Il caso applicativo viene svolto su un cuscinetto i cui dati sulla propagazione della fessura vengono raccolti con un modello Fatigue Crack Growth (FGC) basato sulla legge Paris-Erdogan. Il multi-branch viene usato per affrontare la coesistenza di molteplici modalità di deterioramento concorrenti. Il modello proposto è composto da diversi rami, ognuno dei quali rappresenta una modalità di deterioramento. Alcune importanti ipotesi sono state fatte nella costruzione del modello. In primo luogo, si presume che una volta che il degrado sia iniziato seguendo un determinato percorso, l'apparecchiatura segua questa modalità di deterioramento fino a quando non raggiunge la fine della vita, ovvero non vi è alcun passaggio consentito tra i rami. La seconda ipotesi è che la condizione normale può durare molto a lungo senza segni anormali nelle osservazioni. Le osservazioni emesse in questo stato non sono quindi di interesse nel quadro della modellizzazione del deterioramento. Inoltre, una volta che l'apparecchiatura ha fallito, viene fermata e non è possibile ottenere ulteriori misure. Da questi punti, quello iniziale  $S_0$  così come il finale  $S_i$  sono gli stati considerati come stati fittizi senza emissione di osservazione nel modello. Pertanto, ogni ramo può essere visto come un singolo modello HSMM a cui viene assegnata una probabilità a *priori* (Figura 1.23).

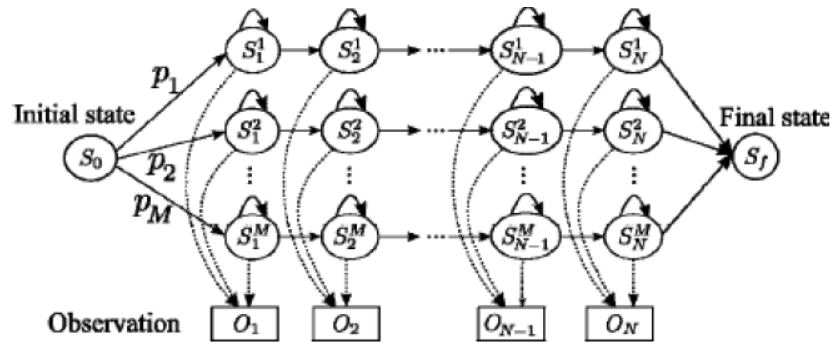


Figura 1.23 Schema del modello HSMM

Viene proposta una struttura a due fasi basata sul modello MB-HSMM per la diagnosi e la prognosi (Figura 1.24):

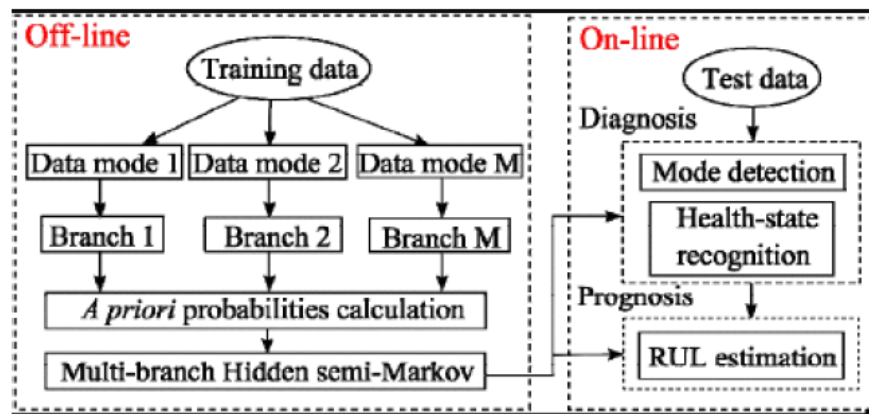


Figura 1.24 Modello MB-HSMM di diagnostica e prognostica

**Fase offline:** Questa fase ha lo scopo di apprendere un modello MB-HSMM dai dati storici. L'idea è di allenare ogni ramo individualmente e quindi combinarli per ottenere un modello finale. Per fare ciò, dividiamo dapprima il set di dati in  $M$  diversi gruppi in cui ognuno corrisponde a una modalità di deterioramento. In questo studio, si suppone che il numero di gruppi  $M$  sia conosciuto in anticipo. Ciascun gruppo di dati viene quindi utilizzato per addestrare un HSMM costituente utilizzando l'algoritmo che utilizza il forward-backward (FB) come processo chiave per l'apprendimento dei parametri e la stima della probabilità. Dopo aver addestrato i rami costitutivi, le probabilità a priori possono essere stimate mediante l'equazione:

$$p_k = P(\lambda_k) = \frac{K_k}{K}, \sum_{k=1}^M p_k = 1 \quad (2)$$

e si ottiene un modello MB-HSMM completo, dove  $\lambda_k$  denota l'HSMM costituente per il ramo  $k$ ,  $K_k$  è il numero di sequenze di dati di allenamento corrispondenti alla modalità  $k$ ,  $K$  è il numero delle sequenze di dati di allenamento totali e  $M$  è il numero totale dei rami.

**Fase online:** In questa fase, il modello generato viene utilizzato per valutare lo stato di salute effettivo dell'apparecchiatura monitorata e per stimare la RUL. Al fine di minimizzare il tasso di errata classificazione, il ramo che ha il massimo di probabilità posteriore considerando i dati, può essere considerato corrispondente al meccanismo di deterioramento attuale, cioè:

$$\check{K} = \operatorname{argmax}_k P(\lambda_k | O)$$

Usando il teorema di Bayes, queste probabilità possono essere date da:

$$P(\lambda_k | O) = \frac{P(O|\lambda_k)P(\lambda_k)}{\sum_{k=1}^M P(O|\lambda_k)P(\lambda_k)} \quad (3)$$

Dove:

- $P(O|\lambda_k)$  è la probabilità del modello  $\lambda_k$  considerati i dati  $O$  e possono essere calcolati attraverso l'algoritmo forward-backward;
- $P(\lambda_k)$  è calcolato dall'equazione:  $p_k = P(\lambda_k) = \frac{K_k}{K}, \sum_{k=1}^M p_k = 1$ .

Dopo aver identificato la modalità di deterioramento, la fase successiva della diagnosi è riconoscere lo stato di salute effettivo dell'apparecchiatura. Questo può essere eseguito grazie all'algoritmo di Viterbi, generalmente utilizzato per trovare la migliore sequenza di stati (detta Viterbi path) in una sequenza di eventi osservati in un processo markoviano. Data la modalità rilevata, la sequenza singolo stato più probabile è quella che fornisce la massima probabilità congiunta del percorso e delle osservazioni:

$$Q^* = \operatorname{argmax}_{Q_{\check{K}}} P(O, Q_{\check{K}} | \lambda_{\check{K}})$$

Dove  $Q_{\check{K}}$  rappresenta un possibile percorso sotto la modalità  $K$ . Lo stato di salute effettivo dell'apparecchiatura è considerato come l'ultimo del percorso  $Q^*$ .

La distribuzione RUL è calcolata come media delle distribuzioni posteriori in modalità costitutive, ponderate in base alle probabilità della modalità posteriore:

$$P(RUL|O) = \sum_{k=1}^M P(RUL|\lambda_k, O) P(\lambda_k | O) \quad (4)$$

con  $P(\lambda_k | O)$  dato dall'equazione (3).

La procedura può ora passare alla stima del RUL per ogni singolo modello HSMM costituente. Si suppone che il componente attraversi stati degradati  $S_1, S_2, \dots, S_N$  prima di raggiungere lo stato di errore  $S_f$  e che sia nello stato  $S_i$  al momento attuale  $t$ . Poiché il modello è rigorosamente sinistra-destra, il RUL può essere calcolato come la somma di due termini: il tempo residuo di permanenza nello stato corrente e la durata per l'apparecchiatura di rimanere negli stati di salute futuri prima di entrare nell'errore. Definiti  $D_i$  e  $D_i^t$  come variabili casuali che rappresentano rispettivamente il tempo totale e il tempo residuo della permanenza nello stato  $S_i$  per  $i = 1, 2, \dots, N$ , si ha:

$$RUL_i^t = D_i^t + \sum_{j=i+1}^N D_j$$

Intuitivamente, il tempo residuo è una variabile casuale condizionale  $D_i^t = D_i - \bar{D}_i | D_i > \bar{D}_i$  dove  $\bar{D}_i$  è il tempo trascorso dall'attrezzatura nello stato  $S_i$  che può essere calcolato grazie all'algoritmo di Viterbi dalla fase diagnostica. Dato che  $D_i$  è distribuito come una gaussiana, come ad esempio  $D_i: N(\mu_i, \sigma_i)$ , si può dedurre che il tempo residuo  $D_i^t | D_i^t > 0$  segue una distribuzione normale con media  $\mu_i$  e deviazione standard  $\sigma_i$ .

Oltre a ciò, dovuto anche all'assunzione Gaussiana, l'elemento  $\Sigma$  è distribuito normalmente. Si vede come:

$$Z = \sum_{j=i+1}^N D_j, \text{ abbiamo } Z: N(\mu_z, \sigma_z) \text{ dove } \mu_z = \sum_{j=i+1}^N \mu_j \text{ e } \sigma_z = \sqrt{\sum_{j=i+1}^N \sigma_j^2}$$

La RUL diventa la somma di una distribuzione normale troncata  $D_i^t$  e una variabile distribuita normale  $Z$ . La funzione di distribuzione cumulativa del RUL è, quindi, data da:

$$F_{RUL}(x) = \frac{1}{1 - \phi\left(\frac{a}{\sigma}\right)} \int_{-\infty}^{x-1} \left[ \phi\left(\frac{x-u}{\sigma}\right) - \phi\left(\frac{a}{\sigma}\right) \right] \varphi(u) du$$

Dove:

$a = -\frac{\mu_t - \bar{D}_t}{\sigma_z}$ ,  $\sigma = -\frac{\sigma_t}{\sigma_z}$  e  $\phi(\xi) = \frac{1}{\sqrt{2\pi}} e^{-\frac{\xi^2}{2}}$  è la funzione di densità di probabilità della distribuzione normale standard e  $\Phi(\cdot)$  è la sua funzione di distribuzione cumulativa.

Dopo aver valutato tutti i singoli rami, la stima RUL sotto MB-HSMM può essere eseguita con l'equazione (4).

In conclusione, la MB-HSMM mostra un risultato molto promettente nell'affrontare il meccanismo di deterioramento multimodale.

2. Questo articolo presenta lo sviluppo di un sistema automatizzato che identifica tipi di guasti nei cuscinetti di macchinari elettrici etichettandoli secondo tre livelli di gravità (basso, medio, alto) in tre differenti elementi dei cuscinetti -anello interno, esterno ed elemento rotante- di cui viene studiato il segnale di vibrazione. Una sintesi del metodo proposto può essere osservata in Figura 1.25.

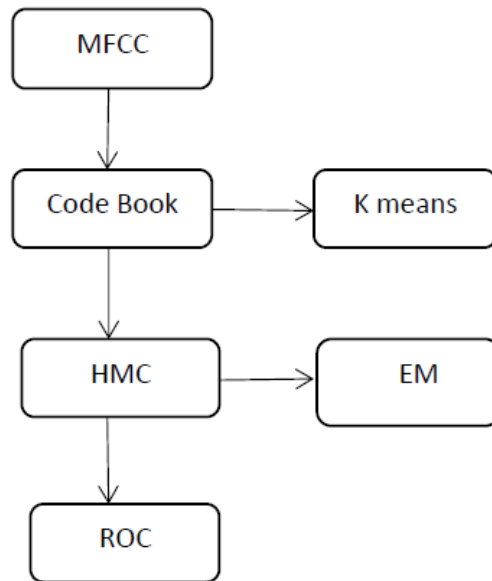


Figura 1.25 Metodologia utilizzata nell'articolo

Tale metodo inizia con i *Cepstral Coefficients on the Mel scale* (MFCC) che sono dei coefficienti che collettivamente rappresentano uno spettro di potenza e che abilitano informazioni sul segnale dal dominio di tempo e di frequenza e permettono l'estrazione di caratteristiche dinamiche lineari e non. Tale fase viene seguita da un Code Book in cui le osservazioni continue vengono associate al centroide discreto più vicino; la tecnica più utilizzata per far ciò è l'algoritmo k-means. Infatti, per addestrare una catena hidden markov con osservazioni discrete, è obbligatorio rendere discrete le caratteristiche. Qui le osservazioni sono vettori di dimensione d (numero di caratteristiche) organizzati in  $X = \{x_1, x_2, \dots, x_y\}$ . L'algoritmo k-means permette di prendere le osservazioni e di dividerle in k sottoinsiemi  $S = S_1, S_2, \dots, S_k$ . Tale algoritmo itera in due fasi principali:

- In primo luogo, le osservazioni che sono vicine l'una all'altra sono associate a risorse:

$$S_i^t \{x_p: ||x_p - m_i^t|| \leq x_p - m_j^t||\} \text{ con } 1 \leq j \leq k$$

- Viene, in seguito, fatto un aggiornamento stimando nuove misure per i centroidi di ogni divisione:

$$m_i^{t+1} = \frac{1}{|S_i^t|} \sum_{X_j \in S_i^t} X_j$$

L'algoritmo, infine, converge quando non si verificano modifiche significative nel passaggio di attualizzazione.

Successivamente si ha una catena hidden markov con osservazioni discrete che viene utilizzata come metodo di classificazione e rappresentazione delle osservazioni.

La catena hidden markov ha le seguenti caratteristiche:

- N: numero di stati del modello
- M: numero di simboli di differenti stati osservabili, noti come  $V = \{v_1, v_2, \dots, v_M\}$ .
- $A = [a_{i,j}]$ : distribuzione di probabilità di una transizione di stato.
- $B = [b_{i,(k)}]$ : distribuzione di probabilità di un'osservazione simbolo nello stato  $j$
- $\pi = [\pi_i]$ : distribuzione di stato iniziale dove  $\pi_i = P\{q_1 = S_i\}$ .
- O: sequenza di osservazione, dove  $O = \{O_1, O_2, \dots, O_T\}$ , con  $O_t \in V$ .
- $\lambda$ : Set di parametri del modello HMC, dove  $\lambda = \{A, B, \pi\}$ .

Con l'aiuto della catena hidden markov, si può osservare che:

- data la sequenza di osservazione e un modello, stima in modo efficiente  $P\{O | \lambda\}$ , cioè la probabilità della sequenza di osservazione dal modello specificato.
- vista la sequenza di osservazione e un modello, seleziona la migliore sequenza di stato che spiega meglio le osservazioni.
- date le osservazioni, regolarizza i parametri del modello per massimizzare  $P\{O | \lambda\}$

Per addestrare i modelli HMC, i parametri  $\lambda = \{A, B, \pi\}$  devono essere regolati in base alle osservazioni, in modo da massimizzare  $P\{O | \lambda\}$ . Viene usato l'algoritmo *Expectation Maximization* (EM) dove la probabilità di essere nello stato  $i$  al tempo  $t$  e in uno stato  $j$  al tempo  $t + 1$ , in modo da stimare i parametri del modello, è:

$$\xi_t(i, j) = P\{q_t = i, q_{t+1} = j | O, \lambda\} = \frac{P\{q_t = i, q_{t+1} = j | O, \lambda\}}{P\{O | \lambda\}}$$

$\xi_t(i, j)$  può essere riformulata anche come funzione di due nuove variabili  $a_t(j)$  e  $b_t(i)$ ,

$$\xi_t(i, j) = \frac{a_t(i) a_{i,j} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha(i) a_{i,j} b_j(o_{t+1}) \beta_{t+1}(j)}$$

Dove i parametri nuovamente stimati sono:

$$a'_{i,j} = \frac{\sum_{t=1}^{T-1} a_t(i) a_{i,j} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{T-1} \sum_{j=1}^N \alpha(i) a_{i,j} b_j(o_{t+1}) \beta_{t+1}(j)}$$

$$b'_j(k) = \frac{\sum_{t=1}^T \sum_{i=1}^N \sum_{t=1}^{T-1} a_t(i) a_{i,j} b_j(o_{t+1}) \beta_t = v_k}{\sum_{i=1}^{T-1} \sum_{j=1}^N \alpha(i) a_{i,j} b_j(o_{t+1}) \beta_{t+1}(j)}$$

Le iterazioni vengono eseguite fino a  $P \{O \mid \lambda\}$ . Per valutare i differenti modelli HMC che meglio rappresentano le osservazioni, è stata utilizzata una curva ROC (*Receiver Operating Characteristics*, ROC) definita come un insieme di schemi grafici per un classificatore binario, dalle quali è risultato che quattro è il numero consigliato di stati per i quali un modello hidden markov dovrebbe essere addestrato per ottenere una precisione diagnostica con un costo computazionale inferiore.

### 1.2.5.2 Modello di regressione

I metodi basati sulla regressione sono generalmente usati nell'industria e in campo accademico per la stima della vita residua. Il principio fondamentale di questi metodi è che la salute del sistema sotto analisi può essere rappresentata tramite alcune variabili chiave e poi, la vita utile residua (*remaining useful life*, RUL) può essere stimata tramite monitoraggio, analisi dell'andamento e previsione di quelle stesse variabili con una soglia  $w$  predefinita. Il metodo di regressione usa i dati provenienti dal monitoraggio condizionale (*conditional monitoring*, CM) in modo da rappresentare il percorso del CM e poi dedurre la distribuzione della vita utile. Tale modello può fornire la funzione di densità di probabilità (*probability density function*, PDF) della RUL che è essenziale per l'analisi del rischio e per il processo decisionale di manutenzione; nella maggior parte dei casi deve essere usata un'approssimazione graduale o una simulazione per trovare la RUL (Si et al., 2011).

#### 1.2.5.2.1 Analisi di regressione

L'analisi della regressione comprende l'insieme di tecniche statistiche per indagare e modellare la dipendenza funzionale di una variabile dipendente  $Y$  su una o più variabili indipendenti chiamate predittori e denotate  $X_1, X_2, \dots, X_k$ .



Grafico a dispersione, correlazione, covarianza e analisi di regressione sono intimamente correlati, ma hanno ruoli molto diversi. Da un lato, un grafico a dispersione fornisce un'immagine visiva della relazione tra una variabile di risposta e una variabile esplicativa. La maggior parte delle volte questa immagine è troppo dettagliata ed essenzialmente qualitativa, ma è utile in quanto fornisce alcune informazioni sulla struttura di un set di dati. Se, utilizzando il diagramma a dispersione, si trova una relazione tra le variabili, allora si usa la correlazione o la covarianza per misurare la forza e la direzione di questa relazione. Correlazione e covarianza quantificano come due variabili costantemente variano insieme, ma non aiutano a descrivere esattamente la relazione specifica che esiste tra le due variabili. Questo è precisamente il ruolo dell'analisi di regressione, che può essere vista come una estensione dei concetti sulla correlazione, covarianza e diagramma di dispersione. L'analisi di regressione misura una curva attraverso i dati presenti sul grafico a dispersione e la descrive per mezzo di un'equazione che ha un potere esplicativo e, nella maggior parte dei casi, un potere predittivo. Il tema dominante dell'analisi di regressione è la riduzione delle dimensioni con una perdita minima di informazioni. L'equazione generale che è alla base di ogni regressione è:

DATI OSSERVATI = COMPONENTI PREVEDIBILI + COMPONENTI IMPREVEDIBILI

Dove:

- DATI OSSERVATI = valori osservati della variabile dipendente
- COMPONENTI PREVEDIBILI = previsioni generate dall'equazione di regressione
- COMPONENTI IMPREVEDIBILI = parti rimanenti o imprevedibili dei dati

L'idea generale dell'analisi di regressione è quella di spostare quante più informazioni possibili nel componente prevedibile, lasciando la componente imprevedibile senza informazioni importanti o pattern particolari.

La forma più comune di regressione è la regressione lineare semplice: considera solo un quadro di due variabili dove la  $X$  è la variabile indipendente che spiega i cambiamenti nella variabile dipendente  $Y$ . È definita come:  $Y = a + (bX)$

dove i parametri  $a$  e  $b$  forniscono una descrizione quantitativa della relazione tra  $X$  e  $Y$ .

Generalmente, i valori esatti dei parametri  $a$  e  $b$  sono sconosciuti. L'obiettivo dell'analisi di regressione è di stimare questi due parametri usando un campione di  $N$  coppie di osservazioni  $(Y_i, X_i)$ , con  $i = 1, \dots, N$ . Vengono indicate le stime di  $a$  e  $b$  con  $\hat{a}$  e  $\hat{b}$ , i cui valori, idealmente, dovrebbero essere il più vicino possibile a quelli di  $a$  e  $b$ .

Se si crede che la vera relazione tra X e Y non sia esattamente lineare, deve essere aggiustato il modello originale aggiungendo un termine di errore; la descrizione della relazione tra X e Y diventa:

$$Y = a + (bX) + \varepsilon$$

dove  $\varepsilon$  è una variabile casuale che tiene conto delle discrepanze che emergono tra i valori effettivi osservati di Y e i valori che sono previsti dal modello "vero".

La scelta della pendenza b e dell'intercetta a della linea di regressione avviene in modo da minimizzare la funzione di tutti i termini di errore nel campione. L'approccio più comunemente usato per la regressione - chiamato minimi quadrati ordinari (ordinary least squares, OLS) e proposto dallo scienziato tedesco Karl Gauss - tenta di minimizzare la somma degli errori al quadrato. Vengono scelti  $\hat{a}$  e  $\hat{b}$  in modo da:

$$\text{minimize } \sum_{i=1}^N \hat{e}_i^2$$

o equivalentemente:

$$\text{minimize } \sigma_{\hat{e}}^2$$

con  $\sigma_{\hat{e}}^2$  definita come la varianza dei singoli errori.

Inoltre, l'approccio OLS è molto facile da impiegare in maniera computazionale, fornisce soluzioni semplici in forma chiusa per i parametri  $\hat{a}$  e  $\hat{b}$  e penalizza severamente i grandi errori individuali garantendo che tutti gli errori rimangano mediamente piccoli.

L'analisi di regressione multipla è una tecnica che estende la regressione semplice in modo da consentire a più variabili indipendenti di accedere all'analisi simultaneamente.  $X_1, X_2, \dots, X_k$  denotano diverse variabili indipendenti la cui relazione con una variabile di risposta Y deve essere investigata.

Un modello di regressione lineare multipla può essere scritto come:

$$Y = a + (b_1 \times X_1) + (b_2 \times X_2) + \dots + (b_k \times X_k) + \varepsilon$$

dove:

Y è la variabile dipendente o espressa;

$X_1, X_2, \dots, X_k$  sono variabili indipendenti o esplicative, chiamate regressori;  $\varepsilon$  è l'errore non correlato a tutti i regressori. Le maggiori fonti di errore nello studio della regressione sono: la deviazione variabile omessa, le variabili estranee, la multi-collinearità, l'eteroschedasticità e la correlazione seriale.

#### **La deviazione variabile omessa:**

Quando una variabile, che dovrebbe essere inclusa in una regressione, viene lasciata fuori, l'effetto stimato delle variabili che vengono incluse nel modello è probabile che sia in errore. Questo crea un "errore di variabile omesso", che può influire sulla validità dell'intera regressione.

Vi sono due casi da prendere in considerazione:

- Se la variabile omessa non è correlata con le altre variabili esplicative considerate, il suo contributo (omesso) viene catturato solo dall'intercetta e dal termine dell'errore;
- Se la variabile omessa è correlata con alcune delle variabili esplicative considerate, allora i coefficienti di queste variabili esplicative saranno distorti verso l'alto o verso il basso in quanto riflettono non solo una stima dell'effetto della variabile a cui sono associati, ma anche in parte gli effetti della variabile omessa.

Omettere le variabili rilevanti provoca comunque gravi problemi. Nel migliore dei casi, complicherà i test di ipotesi e nel peggiore dei casi, invaliderà l'intero risultato della regressione.

#### **Variabili estranee:**

Il problema legato alle variabili estranee è l'opposto della distorsione da variabile omessa. Accade quando la regressione include una variabile indipendente che non ha impatto sulla variabile dipendente. Vengono distinti due casi:

- Se la variabile estranea non è correlata con le altre variabili indipendenti, non interferirà con l'impatto delle variabili rilevanti sulla variabile dipendente. Quindi, non ci sarà distorsione ma semplicemente la variabile indesiderata verrà esclusa e verrà rivalutata la regressione solo con le variabili rimanenti rilevanti.
- Se la variabile estranea è correlata con alcune delle altre variabili indipendenti, il suo effetto sarà più sottile. Non interferirà con l'impatto delle variabili rilevanti sulla variabile dipendente, ma per l'equazione sarà più difficile "assegnare" con precisione la varianza di Y alle X e risolvere gli effetti di variabili indipendenti. Ciò ridurrà la precisione delle stime.

Per evitare la presenza di variabili estranee, è necessario applicare il buon senso quando si definiscono le variabili indipendenti e ispezionare attentamente la loro correlazione con la variabile dipendente.

#### **Multi collinearità**

Si parla di multi-collinearità quando diverse variabili indipendenti in un modello di regressione multipla sono correlate. Quando due variabili indipendenti sono altamente correlate, entrambe trasmettono essenzialmente le stesse informazioni. In questo caso, nessuna delle due può contribuire in modo significativo al modello una volta che l'altra è stata inclusa ma insieme contribuiscono molto. Quindi, il

valore coefficiente di correlazione R è alto (il modello generale soddisfa bene i dati), il p-value del test F di Fisher-Snedecor, per il confronto tra varianze, è basso (la buona misura è improbabile sia una coincidenza), ma nessuna variabile X apporta un contributo significativo. E se entrambe le variabili fossero rimosse dal modello, il risultato sarebbe peggiore. In particolare, la multi-collinearità aumenta l'errore standard delle stime, che riduce la statistica t e quindi riduce il grado di fiducia che può essere inserito nelle stime. Gli intervalli di confidenza possono persino includere zero, il che significa che non è possibile neppure essere sicuri che un aumento del valore X venga associato ad un aumento, o una diminuzione, in Y. L'unica via d'uscita è identificare le variabili che creano la multi-collinearità e rimuoverne una. Un approccio di base consiste nel calcolare il coefficiente di correlazione per ciascuna coppia di variabili predittive. Le grandi cifre di correlazione (sia positive che negative) di solito indicano la fonte del problema e quindi si lascia solo una di quelle molto correlate.

### **Eteroschedasticità:**

Una delle assunzioni classiche dell'analisi di regressione è che i residui siano dispersi in modo casuale e omogeneo nell'intervallo della variabile dipendente stimata: questa caratteristica è chiamata omoschedasticità. In altre parole, la varianza dell'errore residuo dovrebbe essere costante per tutti i valori delle variabili indipendenti. Se questa ipotesi viene violata, si dice che gli errori siano eteroschedastici. Esistono essenzialmente due metodi per trattare l'eteroschedasticità:

- rispettare completamente il modello;
- utilizzare l'opzione di regressione dei *minimi quadrati ponderati (weighted least-squares, WLS)*. Questa è una tecnica di regressione in cui i residui più piccoli vengono pesati di più quando si calcolano gli intervalli di confidenza dei coefficienti di regressione.

### **Correlazione seriale**

I termini di errore della stima dell'equazione OLS devono essere distribuiti indipendentemente l'uno dall'altro e quindi la covarianza tra qualsiasi coppia di errori o termini residui deve essere zero. Se questa covarianza non è zero, allora si dice che i residui sono autocorrelati e si può osservare una relazione lineare tra i termini di errore presenti e passati.

Una semplice regressione può essere espressa come  $Y_t = a + (b \times X_t) + \varepsilon_t$ , dove  $\varepsilon_t = \rho \varepsilon_{t-1} + u_t$ .

Come si può vedere, il termine errore  $\varepsilon$  non è distribuito in modo indipendente. Di conseguenza, le stime di regressione saranno comunque imparziali, ma inefficaci; la varianza degli errori sarà molto sottovalutata e gli intervalli di confidenza troppo grandi. Perciò le inferenze affidabili non possono essere fatte dai risultati della regressione. Il test per la correlazione seriale incluso nella maggior parte dei pacchetti software è il test Durbin-Watson, che rileva solo la correlazione seriale di primo ordine. Si calcola così:

$$DW = \frac{\sum_{t=2}^N (\varepsilon_t - \varepsilon_{t-1})^2}{\sum_{t=2}^N \varepsilon_t^2}$$

dove  $\epsilon$  è il residuo stimato dalla regressione OLS e  $N$  è la dimensione di regressione del campione. Il valore di DW varia da 0 a 4. I valori vicini a 0 indicano una correlazione seriale estremamente positiva; valori vicini a 4 indicano una correlazione seriale estremamente negativa; valori vicini a 2 indicano nessuna correlazione seriale. Come regola generale, DW dovrebbe essere compreso tra 1,5 e 2,5 per indicare l'indipendenza delle osservazioni. La correlazione seriale positiva significa che gli errori standard dei coefficienti sono troppo piccoli, quella negativa indica che gli errori standard sono troppo grandi. Oltre al test di Durbin-Watson, esistono altri test, a seconda del tipo di correlazione seriale ipotizzata, come la stima dei minimi quadrati generalizzati (generalized least squares, GLS) che consente all'utente di correggere automaticamente la correlazione seriale.

Approcci alternativi di regressione sono la regressione non lineare, parametrica e la regressione graduale (stepwise regression).

Dato che, quando ci sono un gran numero di potenziali variabili esplicative per una regressione multipla, non vi è un'unica alternativa nella scelta di quali includere nel modello, la regressione graduale può aiutare determinando la combinazione statisticamente ottimale di variabili indipendenti basate esclusivamente sulla statistica  $t$  dei loro coefficienti stimati.

Esistono essenzialmente tre modalità di funzionamento della regressione graduale:

1. La backward elimination approach inizia con tutte le variabili indipendenti nel modello. Ad ogni passo, al massimo una variabile viene eliminata dal modello.
2. La forward selection approach inizia con nessuna variabile indipendente e contiene solo l'intercetta. Ad ogni passaggio, è possibile aggiungere una sola variabile al modello.
3. La both-sided approach combina i due metodi precedenti.

Correttamente utilizzata, la regressione graduale mette più energia e informazioni a disposizione dell'utente rispetto all'opzione di regressione multipla ordinaria. È particolarmente utile per setacciare un gran numero di potenziali variabili indipendenti e/o perfezionare un modello facendo passare le variabili dentro o fuori. Tuttavia, utilizzato in modo improprio, potrebbe convergere su un modello scadente fornendo un falso senso di sicurezza (Kleinbaum et al., 2013).

#### **1.2.5.2.2 Applicazioni di modelli regressivi**

I modelli basati sulla regressione sono stati descritti nei seguenti articoli riguardanti i cuscinetti:

1. Data science for vibration heteroscedasticity and predictive maintenance of rotary bearings (Lee et al., 2019);
2. A hybrid prognostics technique for rolling element bearings using adaptive predictive models (Ahmad et al., 2017).

1. Viene proposto un algoritmo per predire la RUL di un cuscinetto di un motore elettrico: i segnali di vibrazione dei cuscinetti sono raccolti dalla piattaforma sperimentale e vengono sviluppati dispositivi di rilevamento dei guasti per estrarre le caratteristiche importanti dei cuscinetti nel dominio di tempo (time domain) e nel dominio di frequenza (frequency domain). La regressione dei minimi quadrati pesata (weighted least squares regression, WLS) e la regressione possibile dei minimi quadrati generalizzati (feasible generalized least squares regression, FGLS) sono utilizzati per affrontare il problema dell'eteroschedasticità nel set di dati di vibrazione.

L'approccio proposto (Figura 1.26) per estrarre le caratteristiche chiave e prevedere la RUL dei cuscinetti include la raccolta dati (data collection), la preelaborazione dei dati (data preprocessing), l'estrazione delle funzionalità (feature extraction), la selezione delle funzionalità (feature selection) e i modelli di stima (prediction models).

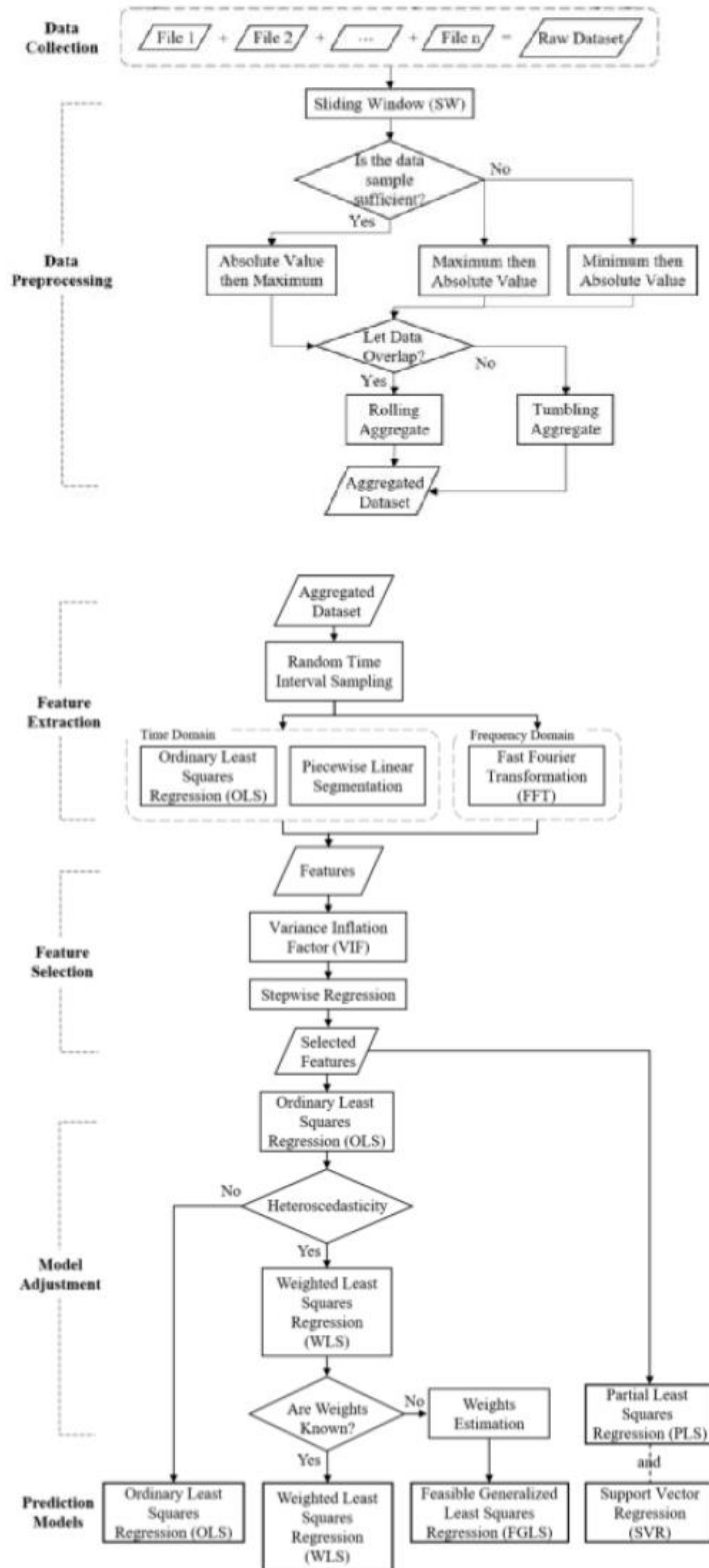


Figura 1.26 Metodologia utilizzata per il modello regressivo

Il processo di raccolta dati, che avviene tramite sensori fino a quando non si verifica un guasto, produce un set di dati su larga scala computazionalmente gravoso. Per ridurre il volume dei dati e mantenere le

caratteristiche del set di dati originale, si passa alla fase di preelaborazione dei dati in cui le informazioni non correlate e ridondanti vengono rimosse e i file di dati dello stesso esperimento vengono uniti in un'unica tabella generando un nuovo set di dati chiamato dati aggregati. I dati con un intervallo di tempo fisso vengono campionati in modo casuale ed ogni dimensione del campione equivale a venti secondi di raccolta dei dati; le caratteristiche dei dati vengono estratte dal dominio temporale e dal dominio di frequenza. Nella feature selection, prima di eseguire un modello di regressione viene rivelato il problema della multi-collinearità tramite un fattore di inflazione di varianza (variance inflation factor, VIF):

$$VIF = \frac{1}{1 - R_i^2}$$

dove  $R_i^2$  è il coefficiente di determinazione del modello di regressione utilizzando l' $i$ -esimo predittore  $x_i$  come variabile di risposta sui predittori rimanenti. Se VIF supera il valore di 10, esiste una multi-collinearità significativa e le caratteristiche con la minima perdita di informazioni devono essere rimosse. Dopo la correzione della multi-collinearità, la regressione graduale viene utilizzata per ottenere le caratteristiche che influenzano in modo significativo la RUL e per rimuovere quelle meno importanti.

Nella fase di regolazione e stima del modello (Model Adjustment and Prediction), viene applicato per primo il metodo dei minimi quadrati ordinari (Ordinary Least Squares, OLS): il concetto di base è quello di ridurre al minimo la somma degli errori quadrati e se l'ipotesi di omoschedasticità degli errori non viene rifiutata, si utilizza OLS per prevedere il valore della RUL. In caso contrario, ovvero se viene rivelata l'eteroschedasticità nel test bianco (White test), WLS è applicato per eliminare tale effetto e per correggere le varianze del coefficiente delle variabili indipendenti (ad esempio, le funzionalità selezionate). Invece, se la struttura dei pesi della regolazione delle varianze non è nota, si stimano i pesi e si aggiungono nel modello OLS originale, denominato FGLS. Per il confronto della precisione di previsione, vengono utilizzati anche la regressione parziale dei minimi quadrati (Partial Least Squares Regression, PLSR) e Support Vector Regression (SVR).

Si ricorda che OLS garantisce un'ottima stima lineare imparziale tramite l'assunzione di:

- a) normalità;
- b) omoschedasticità;
- c) indipendenza seriale.

Il metodo di stima del parametro per la regressione lineare ha la seguente forma:

$$y_t = \beta_p x_{pt} + \varepsilon_t, t = 1, 2, \dots, n$$



dove l'indice  $t$  rappresenta il tempo, il dataset consiste in  $n$  osservazioni con  $p$  variabili indipendenti, una variabile dipendente  $y_t \in \mathbb{R}^{n \times 1}$ , una variabile indipendente  $x_{pt} \in \mathbb{R}^{n \times (p+1)}$ , il coefficiente  $\beta_p \in \mathbb{R}^{(p+1) \times 1}$  di variabili indipendenti, e il termine errore  $\varepsilon_t \in \mathbb{R}^{n \times 1}$ .

I segnali di vibrazione violano l'assunzione di omoschedasticità dal momento che l'ampiezza aumenta esponenzialmente quando il cuscinetto si usura nel tempo. Teoricamente, l'omoschedasticità può essere definita dalla regressione lineare semplice:  $y_t = \beta_0 + \beta_1 x_t + \varepsilon_t$ ;  $\text{Var}(\varepsilon_t | x_t) = \sigma^2$ . Se esiste eteroschedasticità, la varianza degli errori non è identica rispetto a  $x_t$ :  $\text{Var}(\varepsilon_t | x_t) \neq \sigma^2$ . In pratica, il White test è usato per rilevare l'effetto di eteroschedasticità ed in questo studio, WLS e FGLS sono utilizzati per correggere l'eteroschedasticità, aggiungendo i pesi rispettivamente noti o stimati su ciascuna variabile indipendente  $x_t$ .

Come precedentemente accennato, oltre alla regressione dei minimi quadrati ordinari, nell'algoritmo viene utilizzata la support vector regression, preceduta dalla fase di regressione PLS. La regressione PLS è una tecnica che riduce i predittori a un insieme più piccolo di componenti non correlate ed esegue la regressione dei minimi quadrati su questi componenti, anziché sui dati originali; è particolarmente utile quando i predittori sono molto collineari o quando si hanno più predittori rispetto alle osservazioni (Geladi and Kowalski, 1986). È un approccio basato sull'analisi delle componenti principali (principal component analysis, PCA) ed utilizza un insieme di variabili latenti o non osservate per identificare la relazione tra le componenti principali (principal component, PC) e la variabile di risposta. Un tipico metodo PCA trasforma le variabili originali in diversi componenti principali ortogonali per la riduzione della dimensionalità ed elimina il problema della multi-collinearità. Invece di trovare iperpiani della massima varianza delle variabili indipendenti in PCA, PLS estrae un piccolo numero di variabili latenti proiettando la variabile indipendente e la variabile di risposta in un nuovo spazio simultaneamente. Quindi, la PLS consiste in uno sviluppo ulteriore della PCA, in quanto le componenti utilizzate sono derivate non solo dal set di predittori, ma anche dall'insieme delle risposte: in questo modo è possibile massimizzare la varianza non solo delle variabili indipendenti del sistema, ma anche di quelle dipendenti. Così facendo, la scelta dei fattori, ovvero i componenti principali, da impiegare per fare la regressione è eseguita in modo ancora più mirato ed efficace dato che i componenti principali, che spiegano la maggior parte della varianza dei predittori, non è detto che siano anche i più rilevanti ai fini della regressione. La PLS si differenzia dalla PCA perché utilizza il set di dati delle risposte in modo attivo durante l'analisi statistica: ciò permette di bilanciare meglio l'informazione contenuta nelle  $x$  e nelle  $y$ , riducendo l'effetto di grandi ma irrilevanti variazioni dei predittori, ai fini della modellizzazione del fenomeno (Chamberlain, 1982).

Pertanto, PLS conserva i meriti di PCA e non è limitato a variabili non correlate come OLS. PLS gestisce il rumore e la collinearità, non vi è alcun requisito di indipendenza tra le osservazioni e modella anche diverse variabili di risposta simultaneamente.

La *Support Vector Regression* è una variante di regressione della *Support Vector Machine (SVM)* ed è usata comunemente per la predizione piuttosto che per la classificazione. Usa un nuovo tipo di funzione di perdita, vale a dire la funzione di perdita insensibile  $\epsilon$ , per ridurre al minimo la norma del vettore normale relativa all'iperpiano, che riduce la complessità del modello di predizione. SVR è formulato come programmazione quadratica:

$$\min_w \frac{1}{2} \|w\|^2 \text{ soggetto a } \|y_t - (w^T x_t + b)\| \leq \epsilon$$

dove una coppia  $(x_t, y_t)$  è un'osservazione con variabile indipendente  $x_t$  e variabile di risposta  $y_t$  al periodo  $t$ ,  $w$  è il vettore normale dell'iperpiano, e l'errore  $\epsilon \geq 0$  è un parametro predeterminato. L'obiettivo è trovare un iperpiano  $w^T x_t + b$  con un'appropriata deviazione di  $\epsilon$  tra  $y_i$  e il valore previsto.

Lo studio numerico ha mostrato che FGLS fornisce una decisione più conservativa e sicura rispetto ad OLS, che tende a sopravvalutare la RUL causando un ritardo nei tempi di manutenzione delle apparecchiature e, di conseguenza, guasti alle apparecchiature, gravi perdite di capacità o difetti. Pertanto, viene suggerito di applicare FGLS per garantire una minore perdita di OLS.

In conclusione, si può affermare che OLS e FGLS hanno ottenuto rapidamente risultati di predizione mentre SVR ha fornito un valore di predizione più accurato, sebbene abbia comportato un onere computazionale.

2. Viene proposto un approccio basato sul modello di regressione per determinare lo stato di salute dei cuscinetti volventi stimando la RUL. L'approccio ha due fasi distinte: rilevamento del tempo per avviare la previsione (Time to Start Prediction, TSP) e stima RUL. (Figura 1.27)

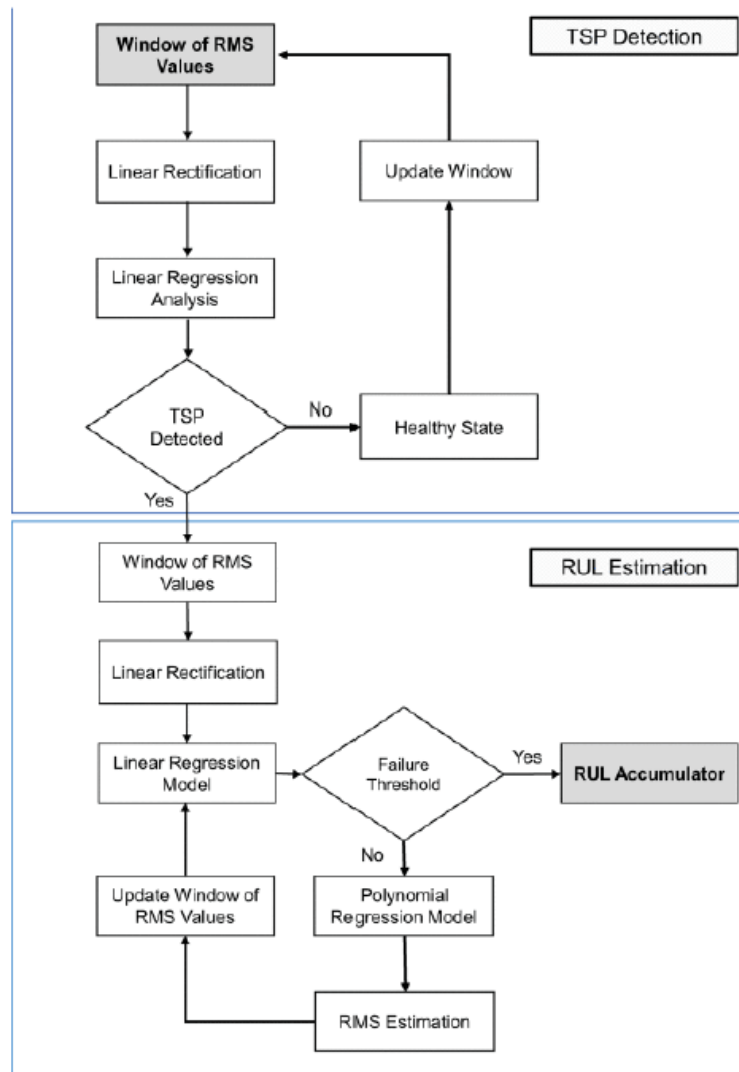


Figura 1.27 Metodologia proposta per il modello regressivo

Fase 1) Il TSP indica l'inizio della degradazione dei cuscinetti e finché non viene rilevato, l'algoritmo proposto non stima la RUL. Il rilevamento TSP inizia con una finestra di valori ottenuti con la radice quadrata media (*Root Mean Square*, RMS), che viene prima livellata usando la tecnica di rettifica lineare (*linear rectification technique*, LRT) proposta. Un modello di regressione lineare viene quindi montato sulla finestra dei valori RMS. Il gradiente del modello di regressione lineare viene utilizzato per determinare se il cuscinetto ha iniziato a degradarsi o meno. Se il rilevamento è dichiarato valido, la finestra dei valori RMS viene aggiornata incorporando le nuove misurazioni e questo processo viene ripetuto finché non viene rilevato il TSP.

Fase 2) Una volta rilevato il TSP, il metodo proposto presuppone l'inizio di un processo irreversibile di degradazione del cuscinetto ed entra nella fase di stima della RUL. Durante tale stima, una finestra di valori RMS viene prima livellata usando la LRT proposta, e quindi, un modello di regressione lineare è

montato su questa finestra. Il gradiente del modello di regressione lineare viene utilizzato per determinare se il rilevamento ha avuto esito negativo o meno. Se il rilevamento è fallito, il processo di stima RUL termina. La RUL del cuscinetto viene stimata calcolando il tempo accumulato prima che il gradiente dell'indicatore di stato raggiunga la soglia di errore. Tuttavia, se la soglia di errore non è stata ancora raggiunta, i valori futuri dell'indicatore di stato sono stimati utilizzando un modello di regressione polinomiale. Questi valori stimati dell'indicatore di stato vengono quindi incorporati nella finestra a dimensione fissa e viene utilizzato un modello di regressione lineare per determinare se viene raggiunta la soglia di errore. Questo processo viene ripetuto fino al raggiungimento della soglia di errore, e quindi poi termina il processo di stima RUL. Invece di richiedere grandi quantità di dati storici, il metodo proposto funziona solo su una piccola finestra di dati, cioè i valori più recenti dell'indicatore di stato.

È necessario tenere a mente che:

- I. Viene presentato un nuovo metodo per determinare il TSP, che utilizza il tasso di crescita dell'indicatore di salute per determinare l'inizio della degradazione nelle condizioni di salute di un cuscinetto. Un TSP accurato migliora le prestazioni predittive di un algoritmo prognostico, specialmente quando la degradazione è nelle sue fasi iniziali e il cuscinetto non è entrato nella fase di degradazione grave;
- II. Una tecnica di rettifica lineare LRT viene proposta per gestire fluttuazioni spurie dell'indicatore di salute. Ciò garantisce che l'indicatore di salute sia sempre una funzione monotona non decrescente, che migliora l'accuratezza del modello regressivo e l'accuratezza delle RUL previste.

Di seguito, vengono definite più nel dettaglio le due fasi, precedentemente accennate.

### **Fase 1) RILEVAMENTO TSP**

Il metodo proposto inizia con una finestra di  $n$  valori RMS e su di essa si adatta un modello di regressione lineare. Come indicato in (1), mostrato di seguito, i parametri  $w$  e  $b$  del modello di regressione lineare sono determinati utilizzando l'approccio ordinario dei minimi quadrati, che riduce al minimo la somma dei residui al quadrato del modello di regressione. Le espressioni di forma chiusa per  $w$  e  $b$  sono date in (2) e (3), rispettivamente, mostrate sotto. Queste espressioni sono ottenute quando il problema di minimizzazione in (4) è risolto per  $w$  e  $b$ :

$$y=wx+b \tag{1}$$

Dove

$$w = \frac{\sum x_i y_i - \frac{\sum x_i \sum y_i}{n}}{\sum x_i^2 - \frac{(\sum x_i)^2}{n}} \quad (2)$$

$$b = \frac{\sum y_i - w \sum x_i}{n} \quad (3)$$

$$arg_{w, b} min\{Q(w, b)\} = arg_{w, b} min\{\sum_{i=1}^n (y_i - wx_i - b)^2\} \quad (4)$$

Il coefficiente  $w$  rappresenta il gradiente della variabile dipendente (l'indicatore di salute) riguardo alla variabile indipendente (tempo). In ogni momento  $t$ ,  $w_t \leq 0$  significa che il cuscinetto è in uno stato di salute, mentre  $w_t > 0$  significa che il cuscinetto ha iniziato a degradarsi. La degradazione può essere lineare o esponenziale. Il TSP rappresenta l'istante in cui il gradiente o il coefficiente di regressione del modello di regressione lineare è uguale o migliore di un certo valore positivo. La relazione tra il gradiente dell'indicatore di salute e una degradazione del cuscinetto viene dedotto dai dati sperimentali.

## Fase 2) STIMA DELLA RUL:

Una volta determinato il TSP, l'algorithmo proposto viene impostato per stimare la RUL del cuscinetto, in qualsiasi momento durante il suo funzionamento. Quando un cuscinetto inizia a deteriorarsi, l'indicatore di stato può seguire una tendenza polinomiale lineare o superiore. Sulla base dei dati presentati nell'articolo, l'indicatore di salute mostra quasi sempre un comportamento di ordine superiore dopo TSP, specialmente quando un rilevamento si avvicina alla fine della sua vita (end of life, EOL). In questo studio, un modello di regressione quadratica viene costruito dopo la determinazione del TSP, che può essere utilizzato per prevedere i valori futuri dell'indicatore di stato (ad esempio, RMS). I parametri del modello, inclusi  $w_1$ ,  $w_2$  e  $b$ , come indicato in (5), sono determinati usando l'approccio ordinario dei minimi quadrati risolvendo il problema di minimizzazione in (6):

$$y = w_1 x^2 + w_2 x + b \quad (5)$$

$$arg_{w_1, w_2, b} min\{Q(w_1, w_2, b)\} = arg_{w_1, w_2, b} min\{\sum_{i=1}^n (y_i - w_1 x_i^2 - w_2 x_i - b)^2\} \quad (6)$$

I futuri valori di RMS sono, quindi, previsti utilizzando il modello di regressione quadratica montato sull'insieme dei  $n$  dati più recenti  $(x_{1..n}, y_{1..n})$ . La RUL viene calcolata in base al numero di passi eseguiti dall'algorithmo fino a raggiungere la soglia di errore e la durata tra due valori successivi del RMS.

In conclusione, il rilevamento del TSP contribuisce in maniera più accurata nella previsione della RUL, in particolare durante le prime fasi di degrado dei cuscinetti. Gli errori casuali o le fluttuazioni spurie dell'indicatore di salute vengono rimossi usando un LRT. Dopo aver rilevato il TSP, l'approccio proposto costruisce un modello di regressione quadratica sulla stessa finestra di campioni RMS che è stato utilizzato in precedenza per determinare il TSP, in modo sia da apprendere la tendenza in evoluzione dell'indicatore di salute che da prevedere i valori futuri. Con ogni previsione dei futuri valori RMS, la finestra viene aggiornata e il gradiente del modello di regressione lineare viene quindi calcolato in base ad essa. Questo gradiente viene verificato rispetto ad un valore predeterminato al fine di determinare se un cuscinetto ha raggiunto o meno la soglia di guasto.

Il metodo proposto produce migliori stime RUL rispetto ai metodi esistenti, in particolare, quando un cuscinetto non è entrato in uno stato grave di degrado.

### 1.2.5.3 Metodi Machine Learning

Il *machine learning* rientra nella disciplina dell'intelligenza artificiale ed è costituito da due fasi: la fase di allenamento, in cui apprende a partire da esempi, e la fase di funzionamento, successiva alla prima, che permette di generalizzare e gestire nuovi dati nello stesso dominio applicativo. Il *machine learning* è un sistema che impara dagli esempi a migliorare la sua conoscenza sui dati e a gestirli in modo migliore e più efficiente.

Le tecniche di apprendimento nel *machine learning* sono quattro:

1. **Apprendimento supervisionato:** questo tipo di allenamento presenta i dati in un set classificato ed etichettato, quindi ad ogni classe di valori viene associata un'etichetta comune nota. La rete viene allenata fornendo in input dei dati associati ad una etichetta in modo che il sistema si alleni a capire che un certo tipo di dato è associato ad una certa etichetta. Una volta che la rete è stata sottoposta ad un numero sufficiente di casi si può testare e osservare quanto sia precisa. Per testarla viene presentato un caso dove l'etichetta è mancante e la rete dovrà trovarla. Questo sembra essere applicabile per il problema dei valori mancanti, infatti, si hanno casi completi che si utilizzano per l'allenamento e casi incompleti per la verifica che la rete dovrà completare.
2. **Apprendimento non supervisionato:** in questa forma di allenamento il set dei dati non è etichettato. Si lascia quindi al sistema, la rete neurale, il compito di cercare relazioni tra i vari dati ed eventualmente suddividerli in categorie. Una volta che il sistema trova le relazioni e suddivide le categorie sta al programmatore, capire come usarle o dargli un'etichetta appropriata. Questo approccio viene usato quando si ha una forte convinzione che ci siano relazioni interne tra i dati.

3. **Apprendimento semi supervisionato:** qui il set di dati è parzialmente etichettato e viene prevalentemente utilizzato per aiutare a ottimizzare la regola di classificazione.
4. **Apprendimento rinforzato:** in questo caso l'obiettivo è apprendere un comportamento ottimale a partire dalle esperienze passate. Un agente esegue azioni che modificano l'ambiente, provocando passaggi da uno stato a l'altro. Quando l'agente ottiene risultati positivi riceve una ricompensa (*reward*) che però può essere temporalmente ritardata rispetto all'azione, o alla sequenza di azioni, che l'hanno determinata. L'obiettivo è apprendere l'azione ottimale in ciascun stato, in modo da massimizzare la somma dei *reward* ottenuti nel lungo periodo.

Nei prossimi paragrafi verranno analizzati i due metodi di previsione della RUL maggiormente presenti negli articoli considerati: *support vector regression* e reti neurali.

### 1.2.5.3.1 Support Vector Regression

Per poter trattare la *support vector regression* è necessario introdurre la *support vector machine* (SVM). Le SVM sono algoritmi di apprendimento supervisionato automatico introdotti da Vapnik che possono essere utilizzati sia per scopi di classificazione che di regressione. Questo approccio fu originariamente sviluppato per l'interpretazione di dati acquisiti da sensori di vario tipo e viene attualmente utilizzato, ad esempio, per il riconoscimento vocale e delle immagini (Granata and DE MARINIS, 2018).

L'algoritmo SVM ottiene la massima efficacia nei problemi di classificazione binari ed è basato sull'idea di trovare un iperpiano che divida al meglio un set di dati in due classi. Tale metodo si basa sul concetto di vettori di supporto (*support vector*, Figura 1.28) che rappresentano i punti dati più vicini all'iperpiano. Tali punti dipendono dal set di dati che si sta analizzando e se vengono rimossi o modificati alterano la posizione dell'iperpiano divisorio. Per questo motivo, possono essere considerati gli elementi critici di un set di dati.

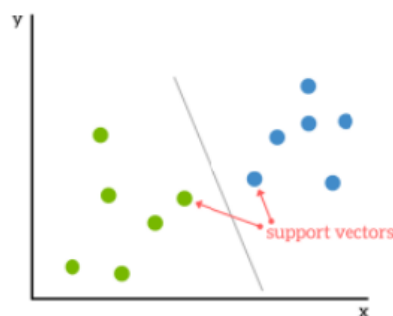


Figura 1.28 Support Vectors

Un altro concetto importante è quello di margine che è definito come la distanza tra i vettori di supporto di due classi differenti più vicini all'iperpiano. Alla metà di questa distanza viene tracciato l'iperpiano, o retta nel caso si stia lavorando a due dimensioni (Figura 1.29). Lo scopo è quello di individuare l'iperpiano che massimizza il margine.

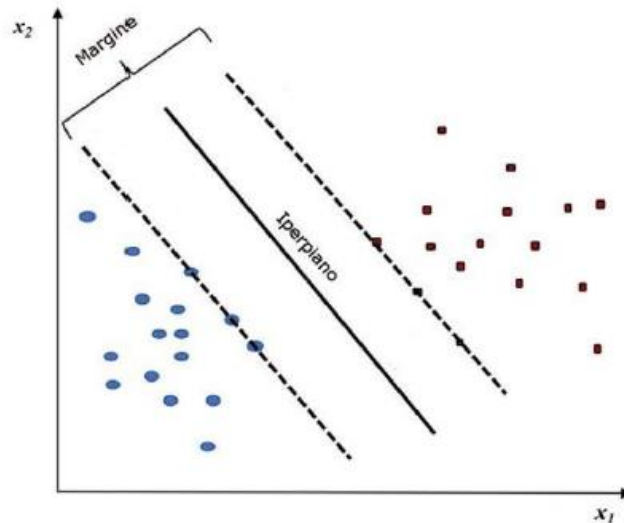


Figura 1.29 Iperpiano di separazione tra le classi

Gli algoritmi SVM utilizzano un insieme di funzioni matematiche definite come *kernel*, il cui scopo è di prendere i dati come input e trasformarli nella forma richiesta qualora non sia possibile determinare un iperpiano linearmente separabile, come avviene nella maggior parte dei casi. In generale il *kernel* può essere definito come:

$$K(x, y) = \langle f(x), f(y) \rangle$$

Dove:

- $K$  è la funzione del kernel;
- $x, y$  sono vettori di input a dimensione  $n$ .
- $f$  è usato per mappare l'input dallo spazio  $n$  dimensionale a quello  $m$  dimensionale (di livello più alto del livello  $n$ ).
- $\langle x, y \rangle$  indica il prodotto scalare ("Algoritmo Support Vector Machine," 2018).

La scelta del *kernel* tra gli altri fattori potrebbe influire notevolmente sulle prestazioni di un modello SVM. L'algoritmo di *support vector machine* si occupa di trovare l'iperpiano migliore ed ha una struttura, riportata in Figura 1.30, divisa in due strati:



- il primo strato è un *kernel* non lineare, costituito dai vettori di supporto, che elabora le serie delle variabili di input;
- il secondo strato è costituito da una somma pesata degli output del *kernel* (Granata and DE MARINIS, 2018).

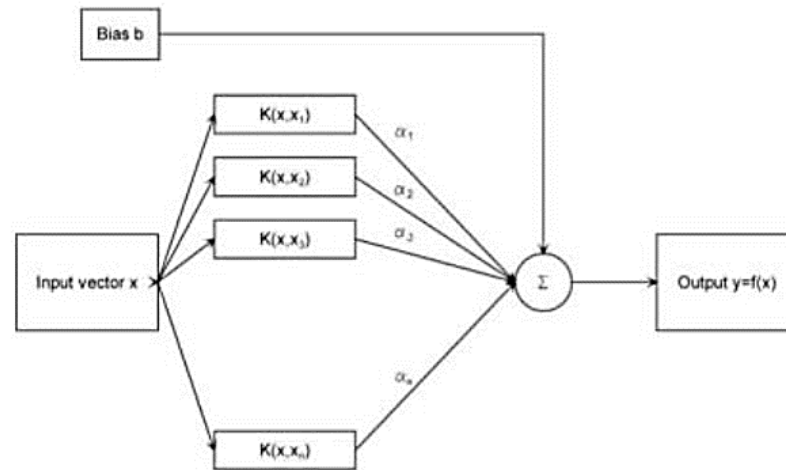


Figura 1.30 Struttura dell'algoritmo di Support Vector

I principali vantaggi di questo algoritmo sono:

- Efficacia in dimensioni spaziali elevate: SVM è uno strumento efficace negli spazi ad alta dimensione ed è particolarmente applicabile alla classificazione dei documenti.
- Efficienza della memoria: per prendere decisioni vengono considerati solo quel sottoinsieme di punti che nella fase di allenamento viene utilizzato nel processo decisionale effettivo di assegnazione di nuovi membri.
- Versatilità: la separazione di classi è spesso altamente non lineare; la capacità di applicare nuovi *kernel* consente una sostanziale flessibilità per i limiti decisionali, portando a una maggiore performance di classificazione.

Tra gli svantaggi principali si hanno:

- Interpretazione non semplice: Uno svantaggio comune delle tecniche non parametriche come gli SVM è la mancanza di trasparenza dei risultati. L'interpretazione dei risultati può comunque essere facilitata dalle tecniche di visualizzazione grafica.
- Metodo non probabilistico: poiché il classificatore funziona posizionando gli oggetti sopra e sotto un iperpiano di classificazione, non esiste un'interpretazione probabilistica diretta per l'appartenenza al gruppo. Tuttavia, una potenziale metrica per determinare l'efficacia della

classificazione è quanto lontano sia il confine della decisione rispetto al nuovo punto (“Algoritmo Support Vector Machine,” 2018).

Quando una *Support Vector Machine* effettua operazioni di regressione, prende il nome di *Support Vector Regression* (SVR). La SVR utilizza gli stessi principi della SVM per la classificazione, con alcune piccole differenze: nel caso della regressione, viene impostato un margine di tolleranza  $\varepsilon$  (Figura 1.31) in approssimazione alla SVM. L’algoritmo è più complicato da prendere in considerazione, tuttavia l’idea principale è la stessa: minimizzare l’errore, individuando l’iperpiano che massimizza il margine;

Dato un training set  $\{(x_1, y_1), (x_2, y_2), \dots, (x_i, y_i)\} \subset X \times R$ , l’obiettivo è trovare una funzione  $f(x)$  che presenti al più uno scarto pari ad  $\varepsilon$ . Per funzioni lineare tali che:

$$y=wx+b \tag{1}$$

La soluzione sarà data dalla minimizzazione della norma euclidea:

$$\min \frac{1}{2} \|w\|^2 \tag{2}$$

Mentre i vincoli:

$$\begin{cases} y_i - wx_i - b \leq \varepsilon \\ wx_i + b - y_i \leq \varepsilon \end{cases} \tag{3}$$

Gli errori minori di  $\varepsilon$  sono tollerati, mentre quelli maggiori no (Awad and Khanna, 2015);

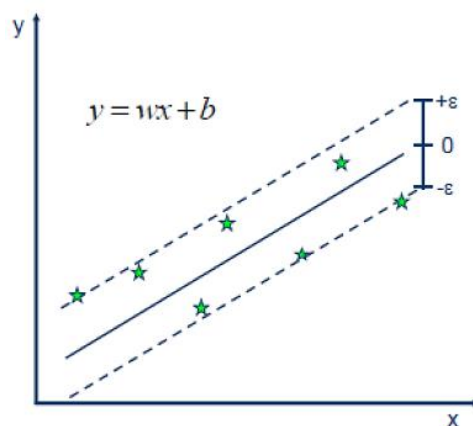


Figura 1.31 Support Vector Regression

Mediante la (2) e la (3) si assume implicitamente che possa esistere una funzione  $f$  che approssima tutte le coppie di valori  $(x_i, y_i)$  con precisione  $\varepsilon$ . In alcuni casi ciò non è possibile, pertanto deve essere tollerato

un certo errore. A tal fine possono essere introdotte delle variabili di rilassamento  $\xi_i, \xi_i^*$  nel problema di ottimizzazione (Figura 1.32), che può quindi essere formulato come segue:

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \quad (4)$$

Soggetto ai vincoli:

$$\begin{cases} y_i - wx_i - b \leq \varepsilon + \xi_i \\ wx_i + b - y_i \leq \varepsilon + \xi_i^* \end{cases} \quad (5)$$

Con  $\xi_i, \xi_i^* \geq 0$  e la costante  $C > 0$  orienta la scelta tra la planarità della funzione  $f$  e l'ammontare delle deviazioni maggiori di  $\varepsilon$  che possono essere tollerate (Granata and DE MARINIS, 2018).

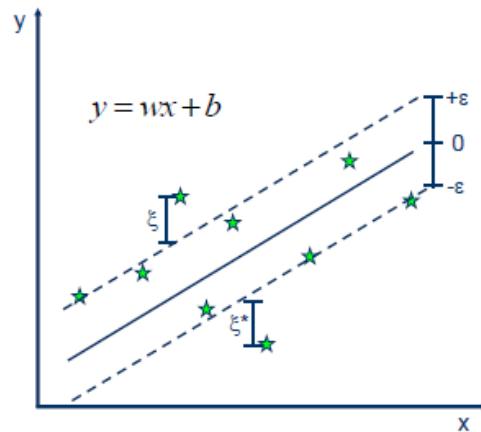


Figura 1.32 Support Vector Regression-funzioni di rilassamento

La *support vector regression* può essere lineare o non lineare e, per quel che concerne il secondo caso, si utilizza la funzione *kernel* per trasformare i dati in uno spazio di caratteristiche dimensionali superiori per consentire di eseguire la separazione lineare (Figura23):

$$y = \sum_{i=1}^N (\alpha_i - \alpha_i^*) \times \langle \varphi(x_i), \varphi(x) \rangle + b$$

$$y = \sum_{i=1}^N (\alpha_i - \alpha_i^*) \times K(x_i, x) + b$$

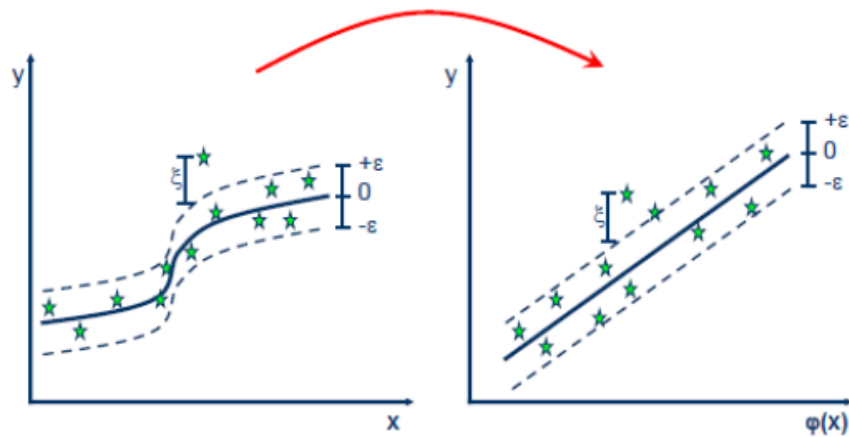


Figura 1.33 trasformazione lineare dello spazio dei dati

Uno dei principali vantaggi della SVR è che la sua complessità computazionale non dipende dalla dimensionalità dello spazio di input. Inoltre, ha eccellenti capacità di generalizzazione, con un'elevata precisione di previsione (Awad and Khanna, 2015).

### 1.2.5.3.2 Applicazioni in letteratura

Per quanto riguarda le applicazioni pratiche, nell'articolo *Condition based maintenance in railway transportation systems based on big data streaming analysis* (Fumeo et al., 2015) viene utilizzata una *Online Support Vector Regression* (OL-SVR) per predire la RUL di un cuscinetto assiale di un treno utilizzando come grandezze la vibrazione, verticale ed orizzontale, e la temperatura. I dati vengono presi da un treno italiano in servizio per otto ore. Il modello si propone di comprimere i dati estraendo solo quelli effettivamente utili per la previsione della RUL e di aggiornare il sistema non appena diventino disponibili nuovi dati aumentando l'accuratezza del sistema e riducendo il costo computazione. Uno schema del metodo proposto viene sintetizzato in Figura 1.34:

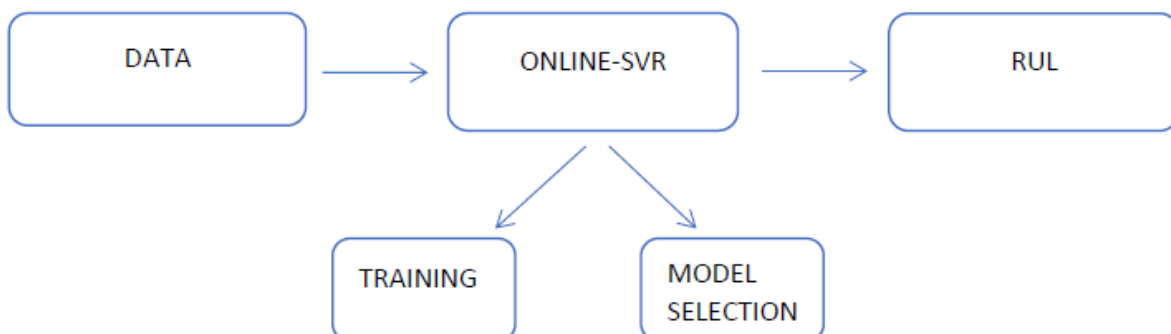


Figura 1.34 Metodologia di applicazione delle SVR

Il metodo si divide sostanzialmente in due fasi:

## 1. Training (TR):

Si andranno a trovare i parametri di interesse ottimizzando il problema. Nella regressione un set di dati  $D_n = \{(x_1, y_1) \dots (x_n, y_n)\}$  viene campionato da una distribuzione non nota  $\mu$  su  $X \times Y$  dove  $X \in R^d$  è uno spazio input ( $x \in X$ ) e  $Y \in R$  è uno spazio output ( $y \in Y$ ). Lo scopo del processo di apprendimento è quello di trovare una funzione  $f: X \rightarrow Y$  che meglio approssimi  $\mu$ . La SVR convenzionalmente cerca un regressore lineare della forma  $y=wx+b$  in uno spazio definito da una funzione non lineare  $\phi: R^d \rightarrow R^D$  dove di solito  $D \gg d$ ;  $w$  e  $b$  si otterranno risolvendo la programmazione quadratica vincolata complessa (*Convex Constrained Quadratic Programming*, CCQP), data dall'equazione (4) precedentemente riportata:

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*)$$

L' algoritmo online aggiorna la funzione SVR ogni volta che un nuovo campione  $(x_{n+1}, y_{n+1})$  viene aggiunto a (o rimosso dal) set di allenamento  $D_n$  e, in tale algoritmo, i parametri di regressione devono essere incrementati o diminuiti in modo incrementale ogni volta che viene aggiunto un nuovo campione.

## 2. Online Model Selection(OL-MS):

si andranno a trovare delle variabili aggiuntive dette iperparametri per trovare la SVR caratterizzata da prestazioni ottimali nella classificazione dei dati. Tale fase viene fatta una sola volta per ogni gruppo, sfruttando i primi  $n$  dati raccolti. Un tipico approccio MS è il *K-Fold Cross Validation* (KCV) che consiste nella suddivisione del dataset totale in  $k$  parti di uguale numerosità e, ad ogni passo, la  $k$ -esima parte del dataset viene ad essere il *validation dataset*, mentre la restante parte costituisce il *training dataset*, evitando così problemi di *overfitting*. Tale approccio viene usato per addestrare un primo set di dati. Quando vengono aggiunti nuovi dati, il set viene aggiornato, aggiungendo i dati più recenti e rimuovendo i più vecchi.

Nell'articolo viene proposta una tecnica euristica di meta-ottimizzazione per poter massimizzare l'accuratezza di un regressore e minimizzare lo sforzo computazionale. L'algoritmo presentato sotto si propone di ottimizzare tutti i parametri coinvolti nella procedura, inglobando in un unico algoritmo i due obiettivi trattati precedentemente. Tale modello generato verrà utilizzato come stimatore della RUL. L'algoritmo creato riesce a raggiungere una buona precisione dei risultati e un onere computazionale minore. I risultati dimostrano i vantaggi di questa soluzione e provano la generalità del metodo che può essere applicato con successo ad altre applicazioni.

### 1.2.5.4 Reti neurali

Una rete neurale può essere definita come un modello dinamico avente la topologia di un grafo orientato, i cui nodi rappresentano i neuroni artificiali e gli archi i pesi sinaptici (Demuth et al., 2014); infatti, il suo funzionamento è simile al cervello umano: nelle sinapsi un neurone invia un segnale su un arco che si attiva avvisando un secondo neurone che deciderà se propagare il segnale su un altro arco oppure no. Le reti neurali sono solitamente composte da più livelli, ognuno dei quali è a sua volta composto da uno o più neuroni. Nel modello più semplice di una rete neurale sono presenti un livello di input e un livello di output e tutti i neuroni del primo livello sono connessi, tramite degli archi, a tutti quelli del secondo livello. Il suo funzionamento può essere descritto in pochi passi. Inizialmente vengono forniti alla rete i valori di input ponendo nel primo livello un numero di neuroni pari al numero di valori in input. A questo punto si moltiplicano per i pesi degli archi e il valore ottenuto viene trasformato con una funzione chiamata “funzione di attivazione”. Il risultato sarà la previsione che la rete neurale ha prodotto per i valori in input. (Figura 1.35)

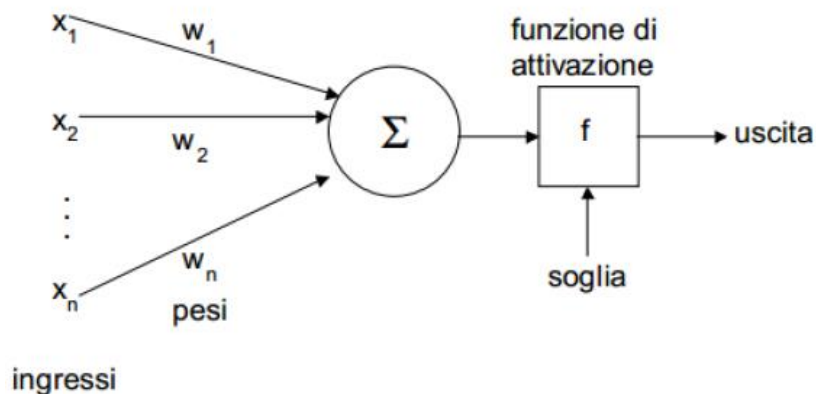


Figura 1.35 Schema rete neurale

Il modello della rete può essere complicato aggiungendo dei livelli intermedi chiamati livelli nascosti (*hidden layers*), che a loro volta avranno un certo numero di neuroni e saranno collegati ai neuroni del livello precedente e a quelli del livello successivo tramite degli archi. Il compito degli strati nascosti è di utilizzare le proprietà (*features*) del dataset per apprendere di nuove. Solitamente prima della fase di creazione di un modello di *machine learning*, il dataset viene analizzato e manipolato per estrarre proprietà significative da quelle di cui già si dispone. Questo processo prende il nome di *feature extraction* e ha un ruolo fondamentale per la creazione di un modello di *machine learning* robusto. Una rete neurale artificiale automatizza il processo di *feature extraction* all'interno di ogni strato nascosto, per questo motivo le reti neurali artificiali sono definite modelli *end-to-end*, appunto perché prendono come

input i dati grezzi del dataset per svolgere attività come classificazione, regressione o clustering (Gullo, 2018). (Figura 1.36)

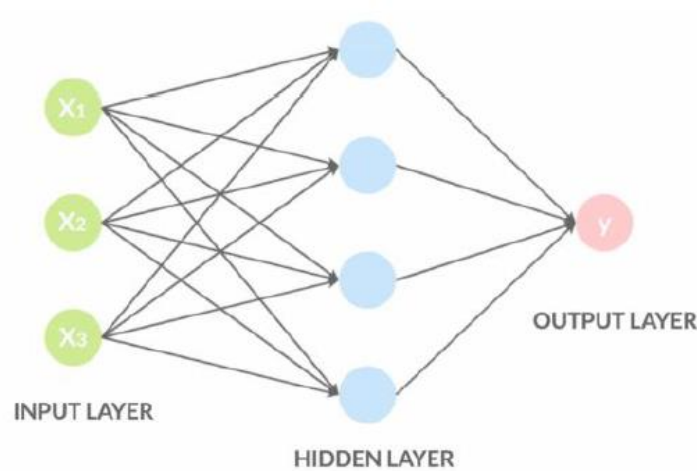


Figura 1.36 Schema di funzionamento di una rete neurale

Maggiore è il numero degli strati, maggiore è la complessità della situazione che la rete può rappresentare ma maggiori sono i tempi necessari per l'apprendimento ed il costo computazionale. Maggiore è il numero di iterazioni, maggiore è il grado di precisione che la rete raggiunge, ma maggiore è il rischio di *overfitting*, cioè il pericolo che la rete rimanga troppo legata ai dati e non sia in grado poi di dare buone risposte di fronte a situazioni mai viste.

Inoltre, in tutti gli strati, tranne quello degli input, ogni nodo è collegato tramite un peso speciale, detto soglia, ad un nodo che, per convenzione, si considera di valore costante. La forma della funzione di attivazione associata ad ogni nodo, tranne quelli di input, è:

$$y = f\left(w_0 + \sum_i w_i x_i\right)$$

Dove  $w_0$  è il valore di soglia,  $x_i$  sono i valori dei neuroni dello strato precedente e  $w_i$  i pesi che collegano quest'ultimi al nodo in esame (Demuth et al., 2014). Il valore di soglia,  $w_0$ , è utile per determinare se l'informazione in entrata elaborata dal neurone debba o meno uscirne procedendo in avanti (*forward propagation*) nella rete, ovvero determina se e in quale misura il neurone debba attivarsi. La presenza di questo parametro incrementa la flessibilità del modello, consentendogli di adattarsi meglio al dataset.

**Allenamento di una rete:**

Nella fase di inizializzazione i pesi degli archi di una rete sono impostati a valori predefiniti o casuali; durante l'allenamento questi valori vengono modificati per trovare la configurazione migliore. L'allenamento di una rete si compone di due fasi principali: propagazione in avanti (*forward propagation*) e all'indietro (*back propagation*). Nella prima fase vengono forniti degli input e calcolato l'output, ripetendo tale operazione per un certo numero di volte. Effettuato questo passaggio avviene la *back propagation* che confronta il valore ottenuto con quello corretto e aggiorna i pesi degli archi a ritroso in modo da minimizzare l'errore. Questo procedimento composto dalle due fasi viene ripetuto molteplici volte durante l'allenamento per ottenere un buon risultato finale. (Figura 1.37, (Gullo, 2018))

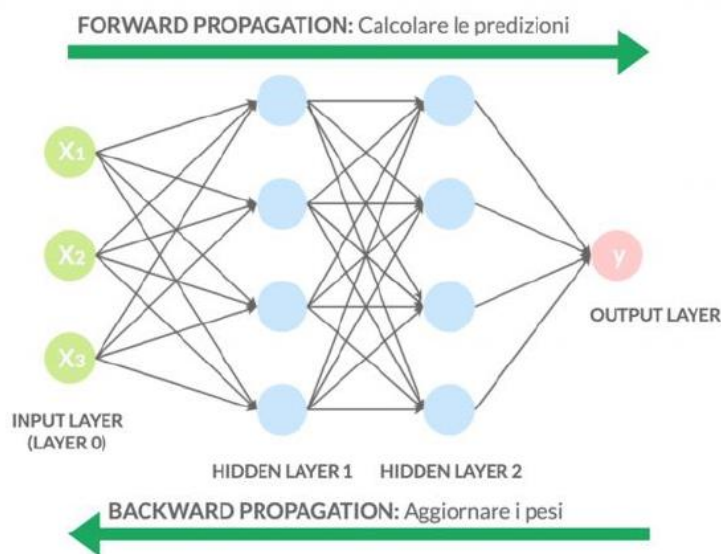


Figura 1.37 Forward and Backward propagation (Gullo, 2018)

Le reti neurali possono essere utilizzate per la regressione, infatti ponendo un unico neurone nel livello di output è possibile allenare una rete neurale a predire valori continui. Nell'ambito della regressione si utilizza l'apprendimento supervisionato; per farlo è necessario un dataset etichettato dove l'etichetta è rappresentata dal valore di output e cioè dalla variabile di cui si vogliono fare previsioni. Per misurare l'errore della rete si utilizza lo scarto quadratico medio. L'obiettivo sarà quindi quello di minimizzarlo in modo da ridurre al minimo l'errore.

I principali vantaggi sono:

- Per come sono costruite, le reti neurali lavorano in parallelo e sono quindi in grado di trattare molti dati mentre nei calcolatori tradizionali ciascun dato viene elaborato individualmente e in successione. Si tratta in sostanza di un sofisticato sistema di tipo statistico dotato di una buona immunità al rumore: se alcune unità del sistema dovessero funzionare male, la rete nel suo



complesso avrebbe delle riduzioni di prestazioni ma difficilmente andrebbe incontro ad un blocco del sistema.

- Le reti neurali non devono essere programmate per svolgere un compito, bensì imparano autonomamente in base all'esperienza o con l'aiuto di un agente esterno.

I maggiori svantaggi sono:

- I modelli prodotti dalle reti neurali, anche se molto efficienti, non sono spiegabili in linguaggio simbolico umano: i risultati vanno accettati "così come sono", da cui la definizione inglese delle reti neurali come sistemi black box.
- Come per qualsiasi algoritmo di modellazione, anche le reti neurali sono efficienti solo per alcuni tipi di variabili e queste vanno scelte con cura.
- Necessitano di una fase di addestramento del sistema che ne ottimizzi i pesi e questa fase può richiedere molto tempo se il numero dei campioni di training e delle variabili analizzate è molto grande.
- Non esistono teoremi o modelli che permettano di definire la rete ottima, quindi la riuscita di una rete dipende molto dall'esperienza del creatore.

#### 1.2.5.4.1 Applicazioni

La rete neurale viene utilizzata nell'articolo *Enhanced Restricted Boltzmann Machine With Prognosability Regularization for Prognostics and Health Assessment* (Liao et al., 2016), incentrato sulla progettazione di un nuovo algoritmo potenziato di estrazione delle caratteristiche, chiamato *Restricted Boltzmann machines* (RBM), per migliorare la precisione di previsione della RUL. Ai fini della valutazione della potenza di previsione delle caratteristiche estratte viene proposto uno schema di previsione RUL, mostrato nella seguente Figura 1.38.

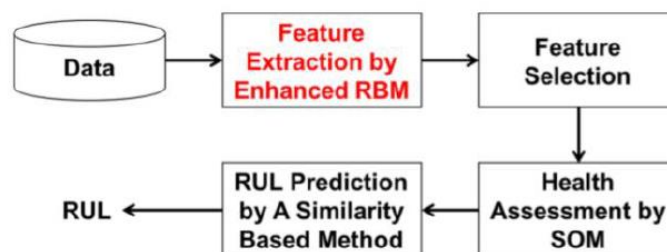


Figura 1.38 Metodologia utilizzata

Dopo che le caratteristiche sono state estratte dall'algoritmo RBM avanzato proposto, queste vengono classificate in base alla monotonia. Un sottoinsieme delle funzioni viene selezionato dalle *features*

estratte e viene aggregato in un valore di salute (*health value*) unidimensionale da una mappa auto-organizzante (*self-organizing map*, SOM) senza supervisione. Il valore di salute viene utilizzato per eseguire una stima RUL tramite un algoritmo di previsione della vita basato sulla somiglianza, che stima la RUL finale come media ponderata dei modelli di degradazione cronologica più corrispondenti.

Per introdurre l'algoritmo RBM, è necessario parlare della macchina di Boltzmann: è un modello grafico probabilistico che può essere interpretato come una rete neurale stocastica. L'addestramento di una macchina di Boltzmann è un lavoro piuttosto impegnativo da un punto di vista computazionale. Tuttavia, questo problema può essere reso più semplice imponendo delle restrizioni sulla topologia della rete su cui si sta lavorando. Questo ci porta alle macchine di Boltzmann ristrette (*Restricted Boltzmann Machines*, RBM); la restrizione, che dà il nome alle RBM, consiste nel non avere connessione tra neuroni dello stesso livello, ma nel connettere ogni neurone con tutti gli altri del livello successivo. Un modo particolarmente utile per studiare le macchine di Boltzmann è quello di considerarle come modelli grafici probabilistici – propriamente modelli grafici non orientati - noti come campi aleatori Markoviani (*Markov random fields*, MRF) (Bellotti and Trotta, n.d.).

La RBM può essere considerata come una rete a due strati che consiste di un livello visibile e uno nascosto; una rappresentazione grafica di una RBM è mostrata nella seguente Figura 1.39.

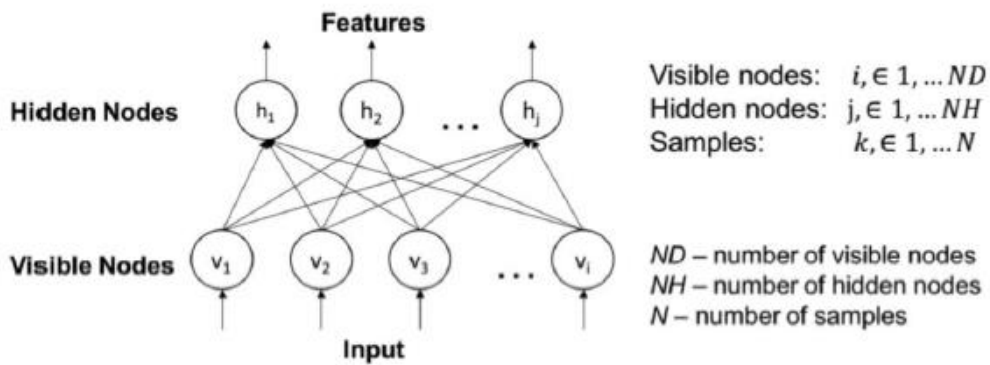


Figura 1.39 Struttura Restricted Boltzmann Machines

In un modello RBM standard, il livello visibile corrisponde alle unità di input osservate  $v$  e il livello nascosto corrisponde ai rilevatori delle caratteristiche dei dati, che sono unità nascoste  $h$ . I dati delle misurazioni di produzione tendono a seguire la distribuzione gaussiana e, pertanto, la funzione energetica dell'RBM è più complessa rispetto al caso binario comune, considerando gaussiani sia i nodi visibili che i nodi nascosti. Aggiornando in sequenza le probabilità per ogni nodo, il sistema converge al suo stato di equilibrio in cui le probabilità per ogni nodo non cambiano più. In questo stato di equilibrio, l'energia della configurazione congiunta dei livelli nascosti e visibili è la somma ponderata negativa di ogni coppia

di nodi meno le distorsioni di tutti i nodi visibili e nascosti. Pertanto, il fine ultimo di RBM è quello di ottimizzare la funzione energetica  $E(v, h)$ , definita come:

$$E(v, h) = \sum_{i \in v_{is}} \frac{(v_i - a_i)^2}{2\delta_i^2} + \sum_{j \in h_{id}} \frac{(h_j - b_j)^2}{2\delta_j^2} - \sum_{i,j} \frac{v_i h_j}{\delta_i \delta_j} \omega_{ij}$$

dove  $v_i, h_j$  sono gli stati dell'unità visibile  $i$  e dell'unità nascosta  $j$ ;  $a_i, b_j$  sono le medie gaussiane per i nodi visibili e nascosti,  $\delta_i$  e  $\delta_j$  sono le deviazioni standard e  $\omega_{ij}$  è il peso tra loro. La probabilità che la RBM assegna ad un vettore visibile è data sommando tutti i vettori nascosti:

$$P(v) = \frac{1}{Z} \sum_h \exp(-E(v, h))$$

Dove  $Z = \sum_{v,h} \exp(-E(v, h))$

I parametri  $(a_i, b_j, \delta_i, \delta_j, \omega_{ij} \in \theta)$  vengono aggiornati tramite il gradiente negativo loglikelihood:

$$\frac{d}{d\theta} (-\log P(v)) = \sum_h P(h|v) \frac{dE(v, h)}{d\theta} - \sum_{v,h} P(h|v) \frac{dE(v, h)}{d\theta}$$

Dove  $P(v, h) = \frac{\exp(-E(v, h))}{Z}$  e  $P(v, h) = \frac{\exp(-E(v, h))}{\sum_h \exp(-E(v, h))}$

L'algoritmo aggiorna i parametri attraverso iterazioni accoppiandole con il tasso di apprendimento (*learning rate*) e/o un parametro di impulso (*momentum parameter*) fino a che non si incontra un criterio di stop. Gli stati di unità nascosta sono utilizzati come caratteristiche usate per la predizione della RUL. Lo scopo dell'applicazione di RBM relativa all'estrazione di caratteristiche meccaniche è quello di estrarle con una migliore tendenza (*trendability*) rispetto ai tradizionali metodi di monitoraggio delle condizioni. Il termine di regolarizzazione supplementare esplora la proprietà interna del sistema meccanico e quindi studia la *trendability* dei nodi nascosti.

La fase successiva alla selezione delle caratteristiche prevede l'utilizzo dell'algoritmo SOM, che può mappare i dati di dimensione elevata in una griglia di dimensioni inferiori. Converte la relazione non lineare del set di dati in una semplice distribuzione geometrica e lo visualizza come mappa della distanza. La mappa è inizializzata con  $m$  neuroni e durante le iterazioni di training, i vettori di peso vengono inizializzati con la stessa dimensione dei vettori di input. La distanza tra ogni input e lo spazio delle caratteristiche selezionate con vettore di peso di ogni neurone, viene calcolata per trovare i neuroni più vicini ai vettori di input dopo ogni iterazione. Questi neuroni più vicini sono denominati come migliori unità di corrispondenza (*best matching units*, BMU). Il vettore di peso viene aggiornato ad ogni iterazione e così i neuroni tendono ad avvicinarsi allo spazio di input. I vettori peso sono aggiornati nel modo seguente:

$$W_l(t+1) = W_l(t) + \alpha(t) h_{l, W_{BMU}}(t) (D - W_l(t))$$

Dove  $t$  è lo step di iterazione,  $h_{l,W_{BMU}}$  denota la vicinanza topologica centrata sulla BMU ( $l \in 1, \dots, m$ ),  $W_{BMU}$  è il peso della BMU,  $D$  è il vettore input e  $\alpha(t)$  è la velocità di apprendimento che è monotonicamente decrescente con  $t$ .

Per la valutazione dell'integrità della macchina, i dati sulla salute della macchina vengono utilizzati per eseguire il training di una mappa SOM di base e quindi il vettore delle caratteristiche dei dati testati viene confrontato con i vettori di peso di tutte le unità nella mappa di base, in modo che la distanza tra questi due venga calcolata come errore minimo di quantizzazione (*minimum quantization error*, MQE), impiegato per quantificare il valore di salute della macchina e definito come segue:

$$\text{Health Value} = \text{MQE} = \min \|D - W_l\|$$

dove  $D$  è il vettore di input e  $W_l$  sono i vettori di peso. Un MQE o un valore di salute più grande è un indicatore di una maggiore deviazione dalla condizione sana del componente (Liao et al., 2016).

In termini più generali, l'algoritmo SOM prevede un insieme di neuroni artificiali, ciascuno con una precisa collocazione sulla mappa rappresentativa degli output, che prendono parte ad un processo noto come "il vincitore prende tutto" (*"the winner takes all"*). Al termine di tale processo, il nodo avente un vettore di pesi più vicino ad un certo input è dichiarato vincitore, mentre i pesi stessi sono aggiornati in modo da avvicinarli al vettore in ingresso. Ciascun nodo ha un certo numero di nodi adiacenti e quando un nodo vince una competizione, anche i pesi dei nodi adiacenti sono modificati, secondo la regola generale che più un nodo è lontano dal nodo vincitore, meno marcata deve essere la variazione dei suoi pesi. Il processo è quindi ripetuto per ogni vettore dell'insieme di training, per un certo numero, usualmente grande, di cicli. Ricordandosi che ingressi diversi producono vincitori diversi, la mappa riesce ad associare i nodi d'uscita con i gruppi o schemi ricorrenti nell'insieme dei dati in ingresso. Se questi schemi sono riconoscibili, essi possono essere associati ai corrispondenti nodi della rete addestrata. In maniera analoga a quella della maggioranza delle reti neurali artificiali, anche la mappa o rete SOM può operare in due distinte modalità:

- durante la fase di addestramento si costruisce la mappa, pertanto la rete si configura ed organizza tramite un processo competitivo. Alla rete deve essere fornito il numero più grande possibile di vettori in ingresso, tali da rappresentare fedelmente la tipologia di vettore che le sarà eventualmente sottoposta nella seconda fase;
- nel corso della seconda fase ogni nuovo vettore d'ingresso può essere velocemente classificato o categorizzato, collocandolo in automatico sulla mappa ottenuta nella fase precedente. Vi sarà sempre un unico neurone vincente, quello il cui vettore dei pesi giace a minor distanza dal vettore appena sottoposto alla rete; tale neurone può essere determinato semplicemente calcolando la distanza euclidea tra i due vettori in questione (Kohonen, 1990).

Essendo un apprendimento non supervisionato, alla rete SOM vengono presentati solo i valori di input e la rete si divide automaticamente in gruppi usando misure di similarità, senza usare confronti con output

noti, e cercando di mettere input simili nello stesso gruppo. È un apprendimento autonomo e non c'è controllo esterno dell'errore (Marmo, 2020).

Infine, per prevedere la RUL, viene utilizzato un metodo di previsione della vita basato sulla somiglianza. In primo luogo, vengono utilizzati diversi set di dati *run-to-failure* con durate di vita e modalità di errore note per generare le curve di degradazione all'interno della libreria di training, dato che, idealmente, avere un maggior numero di dati per l'addestramento può potenzialmente migliorare l'accuratezza della RUL. Il valore di salute è calcolato usando le caratteristiche estratte tramite RBM. In secondo luogo, lo stesso processo di estrazione delle funzionalità viene eseguito sui dati analizzati, in cui sia la durata che la modalità di errore sono sconosciute. Il valore di salute viene calcolato utilizzando tutti i modelli di allenamento relativi ai valori di salute (*training health assessment models*). In terzo luogo, il valore di salute dei dati analizzati viene confrontato con ogni curva di degradazione nella libreria di training. La somiglianza tra i dati ed ogni modello nella libreria viene calcolata utilizzando la distanza euclidea. Per ogni modello nella libreria di training, viene calcolato un valore RUL dalla differenza tra la marca temporale (*time stamp*), attraverso il quale il modello corrisponde meglio ai dati, e la fine del ciclo di vita del modello più corrispondente. Infine, la RUL stimata è definita da una media ponderata dei valori RUL calcolati da ogni modello di training in base alla misura della distanza. Minore è la misura della distanza, maggiore è il peso. Nell'articolo, ciascun peso viene normalizzato in modo che vari da 0 a 1 dividendo il reciproco della misura della distanza per la somma dei reciproci di tutte le misure di distanza.

In conclusione, tramite due casi di studio, che analizzano la vibrazione dei cuscinetti, viene dimostrato che il metodo proposto è efficace nella generazione di caratteristiche che possono migliorare l'accuratezza della stima RUL (Liao et al., 2016).

### 1.2.6 - Conclusioni sull'analisi preliminare della letteratura

A conclusione di tale analisi, sono stati riscontrati aspetti in comune tra le differenti tecniche di previsione. Si è, quindi, voluto definire un *modus operandi* utilizzabile per generalizzare e rendere più immediato l'approccio allo studio della RUL. Il primo step è quello di selezionare i dati (*data collection*) tramite l'utilizzo di sensori che si differenziano in base ai parametri da rilevare. In particolare, come visibile in Figura 1.40, la ricerca dei dati può avvenire basandosi su tre tecniche (Hashemian, 2010):

1. *existing process sensor-based maintenance technique*: utilizza i dati di sensori esistenti come rilevatori di temperatura e pressione per determinare non solo le relative variabili ma anche per identificare eventuali anomalie;
2. *test-sensor-based maintenance technique*: utilizza dati provenienti da *test sensors*, tra cui sensori Wireless, per la misurazione di vibrazioni e la rilevazione di perdite;

3. *test signal-based maintenance technique*: comprende misurazioni come test di resistenza per rilevare difetti quali crepe, corrosione e usura.

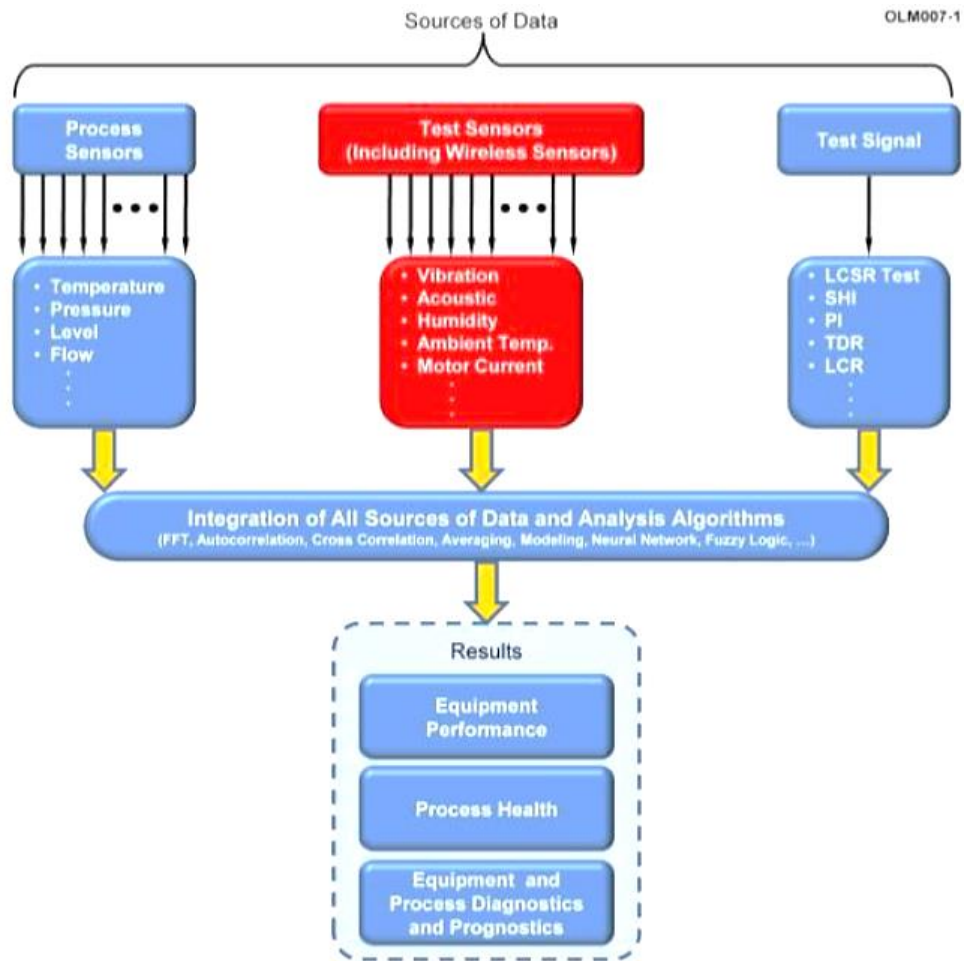


Figura 1.40 Metodologia di ricerca dei dati

Un secondo passo prevede la preelaborazione dei dati (*data preprocessing*) per ridurre il loro volume, mantenendo intatte le caratteristiche del set di dati originali. Successivamente, si ha l'estrazione delle caratteristiche dei dati (*feature extraction*) in cui si generano variabili informative che possono essere difficili da analizzare direttamente o che non sono direttamente paragonabili. A seguito di ciò, viene fatta una selezione delle caratteristiche (*feature selection*) in cui vengono scelte quelle con maggiore importanza o influenza dal set delle features esistenti. Infine, si hanno i due differenti modelli di previsione della RUL: modello statistico e *machine learning*.

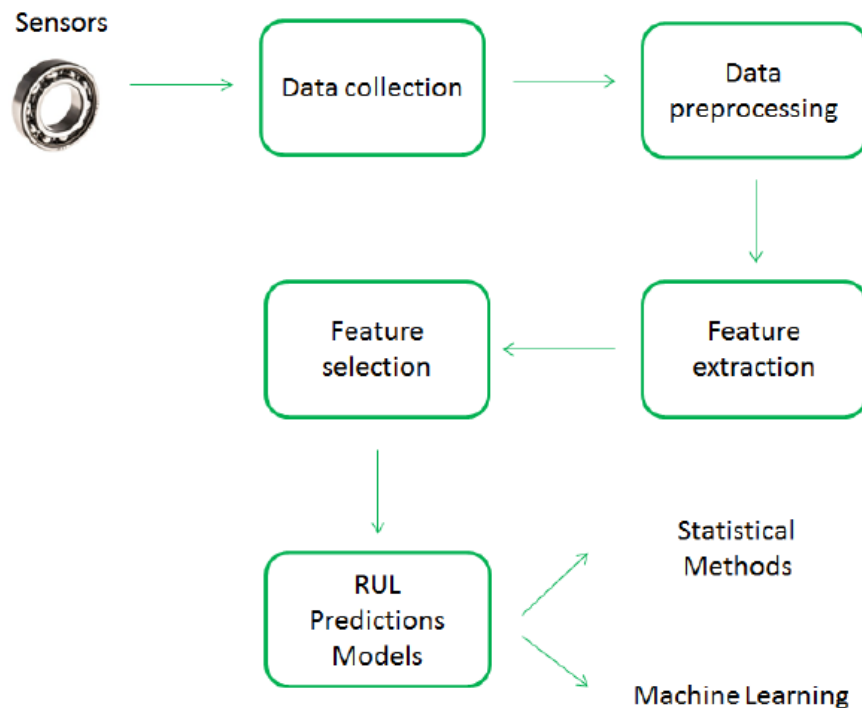


Figura 1.41 Metodologia generale di stima della RUL

### 1.3 - La manutenzione nell'era dell'industria 4.0

Negli ultimi anni, con lo sviluppo delle nuove tecnologie dell'informazione, è emerso il paradigma della quarta rivoluzione tecnologica, ovvero l'Industria 4.0.

Questo concetto si basa su tecnologie eterogenee di integrazione di dati e conoscenze che includono i sistemi cyber-fisici (*Cyber Physical Systems - CPS*), l'*Internet of things* (IoT), l'Internet dei servizi e il *cloud computing*, mantenendo una continua interazione e scambio di informazioni (Roblek et al., 2016). Il suo obiettivo finale è quello di raggiungere una maggiore efficienza operativa, produttività e automazione, migliorando l'efficacia e l'efficienza (Lu, 2017).

Considerando che alcune delle caratteristiche più importanti di questo paradigma sono la digitalizzazione, l'ottimizzazione, l'automazione e l'adattamento, l'interazione uomo-macchina (*Human Machine Interface - HMI*), i servizi a valore aggiunto e le imprese, lo scambio automatico di dati e la comunicazione, è chiaro come l'Industria 4.0 sia un processo industriale di gestione del valore aggiunto e della conoscenza, supportato dall'uso di tecnologie informatiche e algoritmi avanzati (Lu, 2017).

Un aspetto importante dell'Industria 4.0 è la capacità dei sistemi di produzione di monitorare i processi fisici, creando una copia digitale ("*Digital twin*") del sistema fisico, rendendo così possibile la messa in atto

di decisioni intelligenti ottenute grazie alla comunicazione in tempo reale e alla cooperazione con esseri umani, macchine e sensori.

L'obiettivo finale di questa struttura è quello di rendere possibile un flusso continuo e sincronizzato di informazioni attraverso la catena di fornitura, consentendo così un processo di produzione riconfigurabile. (Dalenogare et al., 2018; Zhong et al., 2017).

### 1.3.1 - Rivoluzioni industriali

Il concetto di "Industria 4.0" è stato introdotto nel 2011 su iniziativa del governo federale tedesco in occasione della Fiera di Hannover da un gruppo di lavoro dell'Unione di ricerca Economia-Scienza (Forschungsunion Wirtschaft-Wissenschaft) del Ministero tedesco dell'Istruzione e della Ricerca (Culot et al., 2020).

Si riferisce alla "quarta rivoluzione industriale", l'ultima, ancora in corso, trasformazione dell'ambiente industriale dopo le tre precedenti, caratterizzata rispettivamente dalla macchina a vapore, dall'uso dell'elettricità nei processi industriali e dall'utilizzo dell'ICT (*Information and Communications Technology*) e dell'automazione industriale (Dalenogare et al., 2018; Frank et al., 2019; Vaidya et al., 2018).

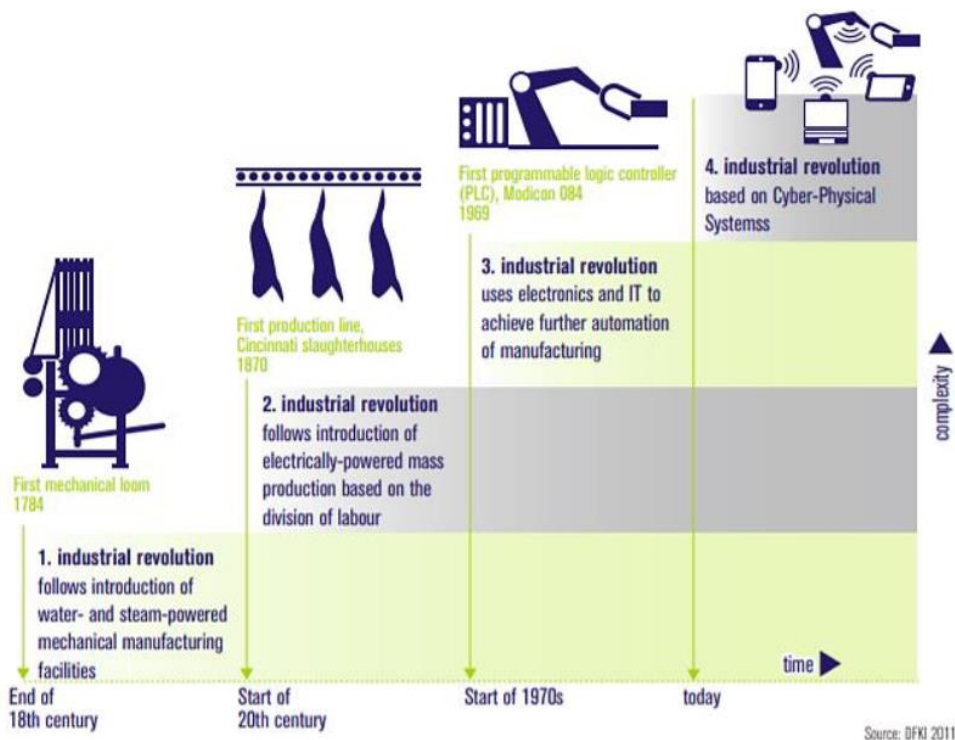


Figura 1.42 La quarta rivoluzione industriale



### **La prima rivoluzione industriale:**

La prima rivoluzione industriale ha avuto origine nella seconda metà del XVIII secolo, segnata da numerose invenzioni nell'ambito delle macchine industriali. Una delle invenzioni più importanti fu il motore a vapore di Thomas Newcomen, che permise la conversione dell'energia termica in energia meccanica. L'uso massiccio di macchine ad acqua e a vapore ha portato a un miglioramento della produttività e dei trasporti, contribuendo al miglioramento del benessere globale. Di conseguenza, questo ha portato ad una rapida crescita della popolazione.

### **La seconda rivoluzione industriale:**

La seconda rivoluzione industriale è stata caratterizzata dall'introduzione della produzione di energia elettrica e risale all'inizio del XX secolo. È stata caratterizzata anche dall'introduzione del concetto di produzione di massa (ad esempio le linee di assemblaggio di Henry Ford). Un'importante innovazione fu l'introduzione del principio della divisione del lavoro: le persone non producevano più prodotti dallo stato iniziale a quello finale, ma si specializzavano in lavori specifici. L'apprendimento dei compiti ripetitivi è stata rapida e facile e ha aumentato il valore aggiunto.

### **La terza rivoluzione industriale:**

La terza rivoluzione industriale è iniziata negli anni '70 del XX secolo con lo sviluppo dell'elettronica e delle tecnologie dell'informazione e della comunicazione. Le innovazioni più importanti che hanno caratterizzato quest'epoca sono state i computer, le comunicazioni mobili e Internet.

L'introduzione dei sensori, degli attuatori e della tecnologia di automazione di controllo dei macchinari nella produzione ha trasformato il contesto industriale, consentendo la riduzione dei costi e portando ad un ulteriore aumento della produttività.

### **Quarta rivoluzione industriale:**

La quarta rivoluzione, la cui concettualizzazione risale a qualche anno fa, si basa essenzialmente sul *networking* digitale del mondo reale e virtuale. Tecnologie come la produzione additiva, la realtà aumentata e i robot autonomi contribuiscono a migliorare l'efficienza dei processi produttivi. Attraverso l'IoT, tutti gli aspetti di un'azienda possono essere collegati, i dati possono essere raccolti e analizzati e il processo di produzione può essere totalmente controllato e ulteriormente ottimizzato.

**Tabella 1.12** Differenze fondamentali di transizione all'industria 4.0 dalle precedenti rivoluzioni industriali (Popkova et al., 2019).

Parametri	Rivoluzioni industriali			
	Prima	Seconda	Terza	Quarta
Periodi	Inizio 18°-19° secolo	Fine 19° - inizio 20° secolo	Seconda metà del 20° secolo	21° secolo
Innovazioni industriali	Produzione di ghisa, macchine a vapore ed industria tessile	Produzione di acciaio di alta qualità, costruzione di linee ferroviarie, elettricità e sostanze chimiche	Energie rinnovabili, tecnologie digitali, organizzazione in rete dei processi di business	Internet of things, robototronica
Modalità tecnologiche	Produzione industriale	Produzione in catena di montaggio	Produzione globale	Produzione automatizzata
Nuove infrastrutture	Attrezzature industriali	Nastri trasportatori, ferrovie	Dispositivi digitali, infrastruttura globale	Internet ad alta velocità, attrezzature robotizzate
Essenza della trasformazione industriale	Formazione della produzione industriale	Formazione della produzione in catena di montaggio	Formazione della produzione globale	Formazione della Produzione automatizzata
Cambiamenti nella logistica	Trasporto a vapore	Trasporto ferroviario	Edifici per la produzione di energia ed elettricità, altri mezzi di trasporto	Esoscheletri, manipolatori, Robototronica
Cambiamenti nella produzione	Prodotti in ghisa	Prodotti in acciaio	Computer	Nuovi materiali da costruzione

### 1.3.2 - Vantaggi e sfide

Come già menzionato nelle sezioni precedenti, il concetto di Industria 4.0 offre numerose opportunità per migliorare i processi aziendali e rendere il processo di produzione più efficiente, più efficace, più individuale e più orientato al cliente a costi relativamente bassi (Vaidya et al., 2018). Questa sezione mira a fornire una panoramica dei benefici e delle sfide associate a questo cambiamento rivoluzionario.

- **Flessibilità:** La flessibilità e la riduzione dei tempi di consegna e del *time-to-market* sono i prerequisiti necessari affinché un'azienda possa rimanere competitiva. Il *lead time* è il tempo necessario per realizzare un prodotto dalle materie prime al prodotto finale. Il periodo di tempo che intercorre tra lo sviluppo del prodotto e la commercializzazione è definito come *time-to-market*. Ciò è possibile grazie ai progressi nella simulazione, nella stampa 3D e nelle reti IoT.

Infatti, queste innovazioni consentono all'azienda di reagire rapidamente e in modo diversificato ai cambiamenti e di ottimizzare i processi in tempo reale. Anche la logistica e l'auto-organizzazione adattiva svolgono un ruolo importante.

- **Prodotto personalizzato:** Attraverso l'interconnessione digitale di aziende e mercati, è possibile sviluppare innovazioni per ridurre al minimo il divario tra domanda e offerta. La considerazione delle esigenze individuali e a breve termine diventa un importante fattore di successo per le aziende ed è possibile solo attraverso la digitalizzazione di tutte le fasi di realizzazione del prodotto (progettazione, pianificazione, produzione). Anche la produzione di piccoli lotti con tecnologie di stampa 3D può giocare un ruolo importante in futuro.
- **Aumento della produttività:** L'ottimizzazione della produzione in termini di consumo di risorse ed energia e l'introduzione di controlli predittivi aumentano la produttività. I dati delle macchine e della produzione sono raccolti e analizzati in piattaforme virtuali.
- **Miglioramento del supporto del personale:** Grazie all'interazione uomo-macchina, come i sistemi di assistenza visiva nelle postazioni di lavoro, è possibile garantire la qualità del prodotto. La formazione digitale del personale può arricchire il *know-how* dei dipendenti e fornire informazioni importanti.

Il paradigma dell'Industria 4.0 è ancora una visione per il futuro, poiché comporta molti tipi di difficoltà e sfide non facili da superare. Queste sfide sono scientifiche, tecnologiche, economiche, sociali e politiche (Zhou et al., 2015).

In effetti, un enorme sforzo sarà legato sia all'installazione che alla configurazione dei dispositivi e delle reti di comunicazione (Zhou et al., 2015). Inoltre, la natura non strutturata dei dati raccolti e le operazioni ad alta velocità richiederanno standard più elevati di acquisizione dati (Cheng et al., 2018).

Infine, l'aumento dell'automazione e il conseguente impatto sulla forza lavoro potrebbero rappresentare una questione importante e dovrebbero essere affrontati in modo consapevole e delicato.

### 1.3.3 - Tecnologie chiave

L'intelligenza è un elemento chiave per lo sviluppo futuro della produzione ed è destinata ad influenzare globalmente il business, a diversi livelli, con l'obiettivo di fornire la flessibilità necessaria per rispondere rapidamente ai cambiamenti di produzione (Karkalos et al., 2019).

Come risultato dei crescenti progressi nell'IT e nell'elettronica, il paradigma dell'Industria 4.0 è collegato al concetto di produzione avanzata o "Smart" e si fonda sull'integrazione di organizzazioni, attrezzature,

persone e prodotti e sulla continua comunicazione tra di essi al fine di stabilire una produzione industriale efficiente, collaborativa e sostenibile (Karkalos et al., 2019; Wang et al., 2016).

Un altro concetto, strettamente connesso a quello dell'Industria 4.0 è, quindi, quello di *Smart Factory*. Infatti, diversi autori (Alcácer and Cruz-Machado, 2019; Lu, 2017; Wagner et al., 2017) considerano il quadro dell'Industria 4.0 come il naturale sviluppo della *Smart Factory*. Una fabbrica intelligente è una fabbrica la cui complessità interna deve aumentare in modo intelligente: decentramento nell'organizzazione, unità organizzative indipendenti, comunicazione tra le singole strutture.

I fondamenti della *Smart Factory* sono i sistemi cyber-fisici (CPS), sistemi di unità di calcolo strettamente connessi al mondo fisico circostante e ai suoi processi in corso, che forniscono e utilizzano servizi di accesso ai dati e di elaborazione disponibili su Internet (Monostori et al., 2016). I CPS sono definiti come "la convergenza del mondo fisico e digitale attraverso la creazione di reti globali per il business che incorporano i loro macchinari, sistemi di immagazzinamento e impianti di produzione" (Lu, 2017) (Figura 1.43). I CPS possono registrare i dati fisici per mezzo di sensori, analizzarli e archivarli, nonché intervenire sui processi tramite attuatori.

Questa combinazione di informazioni e materiali è fondamentale per il miglioramento delle prestazioni industriali globali: la raccolta e l'analisi dei dati potrebbe consentire di aumentare la produttività, migliorare le prestazioni dei dipendenti e produrre prodotti eccellenti a costi inferiori (Lu, 2017).

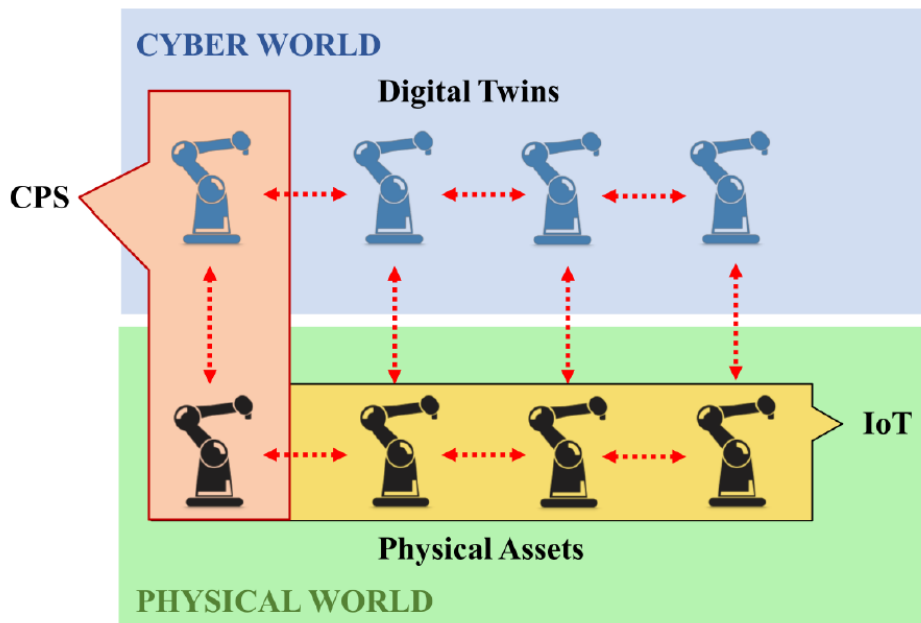


Figura 1.43 Rappresentazione dei Sistemi Cyber-Fisici e dell'Internet of Things

In particolare, l'Industria 4.0 si basa sull'interazione di diversi elementi e tecnologie. Come rappresentato in Figura 1.44 tutti i settori di business (ingegneria, produzione, manutenzione, supply chain, logistica) sono coinvolti in questa trasformazione.

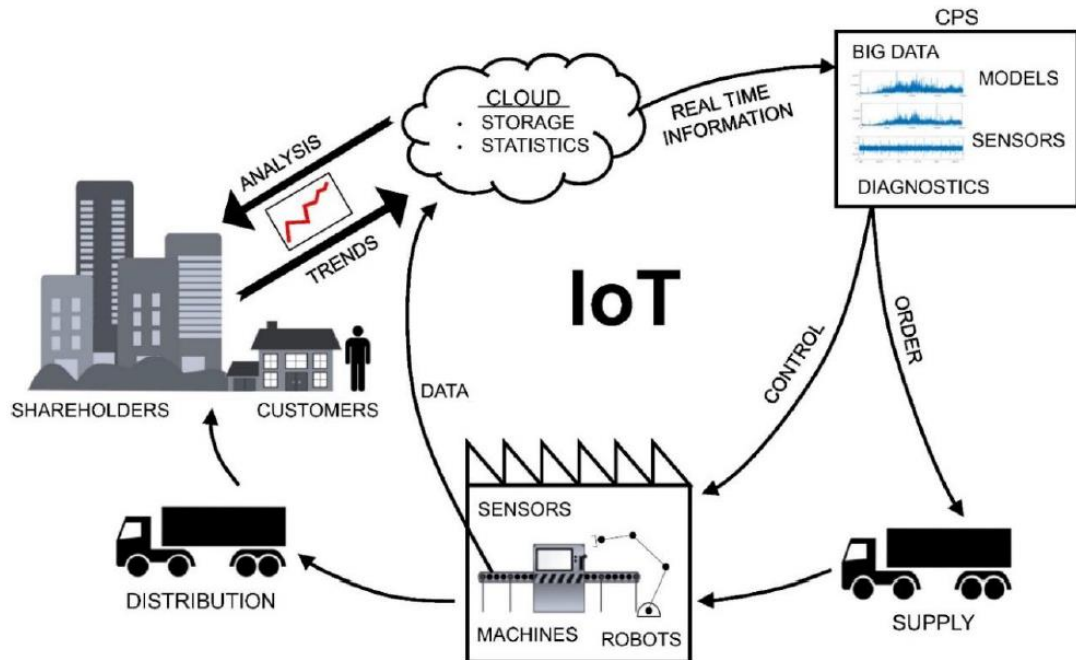


Figura 1.44 Tecnologie abilitanti ed elementi dell'industria 4.0 (Ahuett-Garza and Kurfess, 2018)

Per riassumere i principali aspetti dell'innovazione tecnologica descritti dal paradigma Industria 4.0, nella letteratura scientifica sono state individuate nove tecnologie chiave, o "pilastri" (Alcácer and Cruz-Machado, 2019; Laudante, 2017; Vaidya et al., 2018):

- Internet of Things (IoT);
- Big Data e Analytics;
- Simulazione;
- Cloud;
- Integrazione di sistema: Integrazione di sistema orizzontale e verticale;
- Realtà aumentata;
- Robot autonomi;
- Produzione additiva;
- Sicurezza informatica.

### 1.3.3.1 Internet of things

Il paradigma dell'*Internet of things* (IoT) è definito come un sistema integrato di dispositivi interconnessi e intelligenti che connette oggetti, persone e luoghi (Vaidya et al., 2018). L'IoT è stata descritta come la tecnologia che "permette alle persone e alle cose di essere connesse in qualsiasi momento, in qualsiasi luogo, con qualsiasi cosa e con chiunque, utilizzando idealmente qualsiasi percorso/rete e qualsiasi servizio" (Sezer et al., 2018)].

Il concetto di IoT è focalizzato sulle interazioni tra le cose in rete, permettendo le comunicazioni, grazie all'uso di tecnologie in evoluzione come il rilevamento, la computazione ubiqua (*ubiquitous computing* – *ubicomp*), RFID e *Wireless Sensor Networks* (WSN) (Alcácer and Cruz-Machado, 2019; Li et al., 2015) (Figura 1.45).

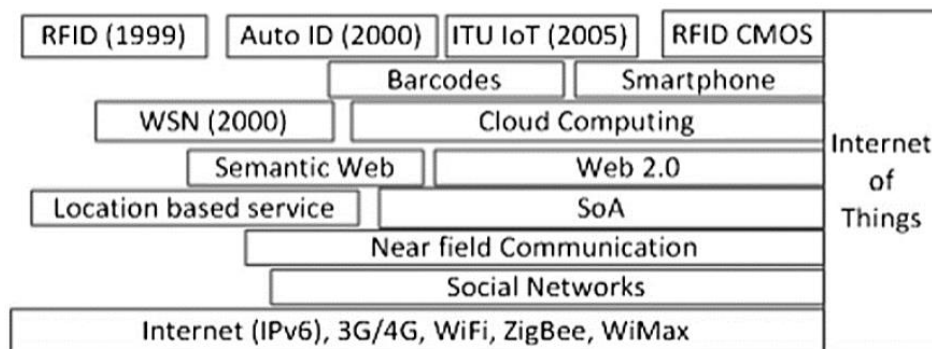


Figura 1.45 Tecnologie abilitanti per IoT (Li et al., 2015)

In questo contesto, un oggetto intelligente è un oggetto che ha la consapevolezza di sé, la capacità di autodiagnosi e la consapevolezza della posizione; può interagire con altri oggetti intelligenti e con l'ambiente circostante (ad esempio gli attuatori) e possiede alcune capacità di elaborazione dei dati. Nell'industria manifatturiera, l'IoT viene utilizzato per acquisire informazioni dall'officina come lo stato delle macchine, i livelli di inventario, il consumo di energia, ecc. (Shrouf and Miragliotta, 2015).

Per distinguere le applicazioni industriali con molteplici possibilità di impiego, viene spesso utilizzato il termine "*Industrial Internet of Things*" (IIoT), che è stato definito come "l'uso delle tecnologie dell'*Internet of Things* (IoT) nella produzione" (Boyes et al., 2018).

### 1.3.3.2 Big Data and Analytics

Grazie all'ampio utilizzo di sistemi di controllo distribuiti, nell'industria di processo è stata raccolta una grande quantità di dati e l'interesse per le possibilità offerte dall'estrazione di informazioni da una così

grande quantità di dati sta crescendo rapidamente. Il *data mining*, l'analisi e l'apprendimento delle macchine sono tra le tecnologie descritte come "4.0" e sono considerate ogni giorno più importanti.

Essi servono come strumento per migliorare la comprensione del processo, giocando un ruolo chiave nel contesto dell'industria. Infatti, i processi innovativi e progressivamente complessi richiedono sempre più l'uso di modellizzazione, monitoraggio, previsione e controllo dei processi basati sui dati. Lo scopo di queste tecniche è quello di esaminare i modelli dei dati di processo e le relazioni tra le variabili al fine di estrarre informazioni utili.

In particolare, al fine di trarre valore da tutti i nuovi dati ora disponibili, sono stati sviluppati nuovi algoritmi, basati sull'apprendimento automatico, per effettuare il *data mining* e l'analisi dei dati.

Il *Machine Learning* (ML) è identificato come un sottoinsieme di intelligenza artificiale. Nella prima fase di un progetto di ML vengono valutate e strutturate grandi quantità di dati (*Big Data*). Nella fase successiva, i dati di input sono elaborati da algoritmi. In particolare, gli algoritmi di ML, come già accennato possono essere classificati in tre tipi principali: apprendimento supervisionato, apprendimento non supervisionato e apprendimento per rinforzo.

Mentre l'apprendimento supervisionato ha lo scopo di apprendere una funzione che mappa gli ingressi in uscita con l'aiuto di esempi di coppie ingresso-uscita, nell'apprendimento non supervisionato il sistema non viene fornito con un feedback esplicito o con l'uscita desiderata e l'obiettivo è quello di scoprire i modelli nei dati di ingresso. Inoltre, per l'apprendimento per rinforzo, il sistema non viene presentato con coppie input-output ma gli viene comunque fornito un feedback sulle sue precedenti esperienze sotto forma di premi o punizioni associate alle azioni. Nella fase finale, il modello sviluppato viene valutato e corretto se necessario (Zhou et al., 2017).

I diversi modelli sono sviluppati per analizzare i dati sia offline che online per varie applicazioni, come il monitoraggio dei processi, l'analisi delle tendenze, la diagnosi e la classificazione dei guasti, il *clustering* di modalità di funzionamento, il *soft sensing* di variabili chiave o variabili di qualità (Ge, 2017).

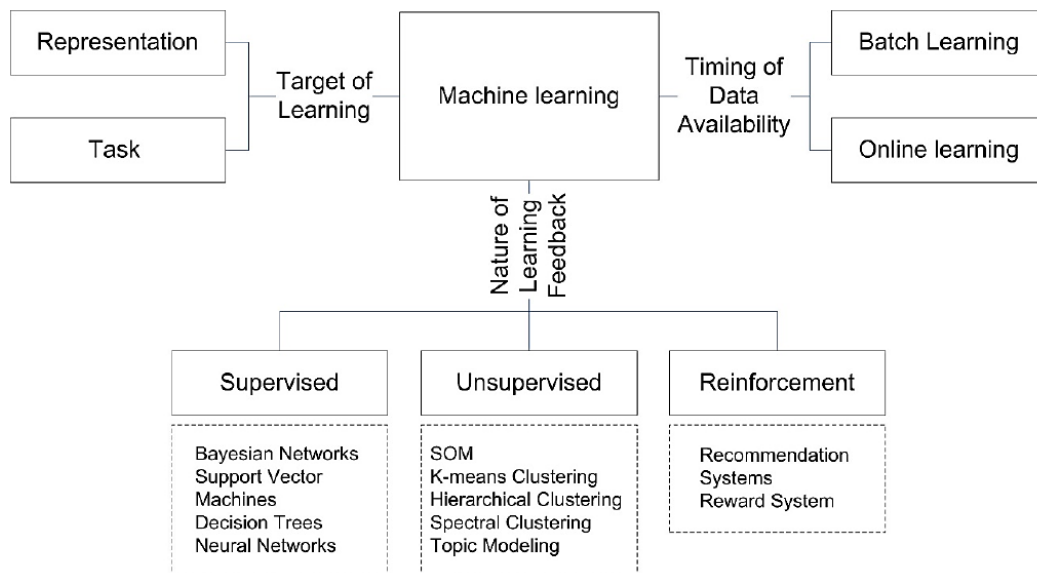


Figura 1.46 Una tassonomia multidimensionale dell'apprendimento automatico (Zhou et al., 2017).

### 1.3.3.3 Simulazione

La simulazione è una tecnica che può essere utilizzata per prevedere il comportamento di sistemi complessi reali negli impianti industriali. In tal modo, la simulazione può fornire le conoscenze necessarie per supportare le decisioni di produzione, come la progettazione di layout di fabbrica, le interoperazioni della catena di fornitura, lo sviluppo di piani di produzione, la valutazione dei benefici e dei rischi delle strategie, l'ottimizzazione dei percorsi e l'assegnazione dei veicoli agli operatori, la pianificazione della capacità e l'ottimizzazione delle linee (Goodall et al., 2019; Gunal, 2019). Inoltre, l'uso della simulazione permette esperimenti per la validazione di prodotti, processi o progettazione e configurazione di sistemi, favorendo la riduzione dei costi, la diminuzione dei cicli di sviluppo e l'aumento della qualità del prodotto (Alcácer and Cruz-Machado, 2019).

### 1.3.3.4 Cloud

Le piattaforme IT, basate sull'uso della tecnologia *cloud*, sono un supporto tecnico utilizzato dalle organizzazioni per la connessione e la comunicazione di vari elementi dell'Industria 4.0, sostenendo la crescente necessità di condividere e memorizzare i dati tra i siti e le aziende con tempi di reazione rapidi (Vaidya et al., 2018).

Inoltre, il *cloud computing* si riferisce alla "fornitura di servizi di calcolo attraverso risorse esposte e scalabili su Internet" (Zhong et al., 2017). Uno degli aspetti più interessanti del cloud computing è la



scalabilità delle risorse, poiché consente alle organizzazioni di personalizzare le proprie esigenze e quindi i propri investimenti nella tecnologia.

Il *National Institute of Standards and Technology* (NIST), definisce il *cloud computing* come segue [Mell e Grance 2011]:

"Il *cloud computing* è un modello per consentire un accesso di rete on-demand ubiquo, conveniente e on-demand a un pool condiviso di risorse di calcolo configurabili (ad esempio, reti, server, storage, applicazioni e servizi) che possono essere rapidamente forniti e rilasciati con il minimo sforzo di gestione o di interazione con i fornitori di servizi".

NIST descrive anche cinque caratteristiche che un modello di *cloud* ideale dovrebbe possedere: *self-service on-demand*, ampio accesso alla rete, *pooling* delle risorse, elasticità rapida e servizio misurato. Inoltre, definisce tre tipologie di modelli di servizio che potrebbero essere implementati (Mell and Grance, 2011):

- *Software as a Service* (SaaS) come capacità di consentire all'utente di utilizzare le applicazioni del provider da diversi dispositivi client.
- *Platform as a Service* (PaaS) come la capacità di consentire all'utente di implementare sull'infrastruttura *cloud* applicazioni create dal consumatore o acquisite da applicazioni create utilizzando linguaggi di programmazione, librerie, servizi e strumenti supportati dal fornitore.
- *Infrastructure as a Service* (IaaS) come la fornitura di risorse informatiche essenziali per consentire al consumatore di distribuire ed eseguire software arbitrari (ad esempio sistemi operativi, applicazioni, ecc.).

Inoltre, i progressi nel *cloud computing* possono fornire l'accesso sia a set di dati più grandi che a *cluster* di processori remoti, consentendo all'utente di ottenere prestazioni di modellazione che altrimenti non avrebbe ottenuto (Oztemel and Gursev, 2020).

### **1.3.3.5 Integrazione di sistema: Integrazione di sistema orizzontale e verticale;**

Un concetto chiave dell'Industria 4.0 è il miglioramento dell'ambiente di produzione al fine di rendere disponibili i dati per tutti gli attori rilevanti.

Il paradigma si estende in tre direzioni di integrazione (Peres et al., 2018):

- Integrazione orizzontale attraverso l'intera rete del valore: la connessione e il coordinamento di tutti gli attori della catena del valore ne permette l'ottimizzazione, creando un ecosistema molto dinamico;
- Ingegneria end-to-end lungo l'intero ciclo di vita del prodotto: la capacità di immagazzinare ed elaborare le informazioni attraverso tutte le fasi del ciclo di vita del prodotto è essenziale per garantire la possibilità di miglioramento, non solo per quanto riguarda la produzione, ma anche per quanto riguarda la progettazione e i fornitori di materiali;
- Integrazione verticale e sistemi di produzione in rete: l'integrazione tra i diversi componenti e attori (come le risorse, le persone e i sistemi di esecuzione della produzione) permette di ottimizzare il processo.

### 1.3.3.6 Realtà aumentata

Realtà aumentata è un termine usato per descrivere una versione migliorata della realtà in cui la visione del mondo fisico (dal vivo o meno) è aumentata con immagini sovrapposte, generate da un computer. Questa tecnologia potrebbe essere utilizzata per migliorare sia il processo di progettazione che le *operations* (Gattullo et al., 2019; Oztemel and Gursev, 2020) (Figura 1.47).

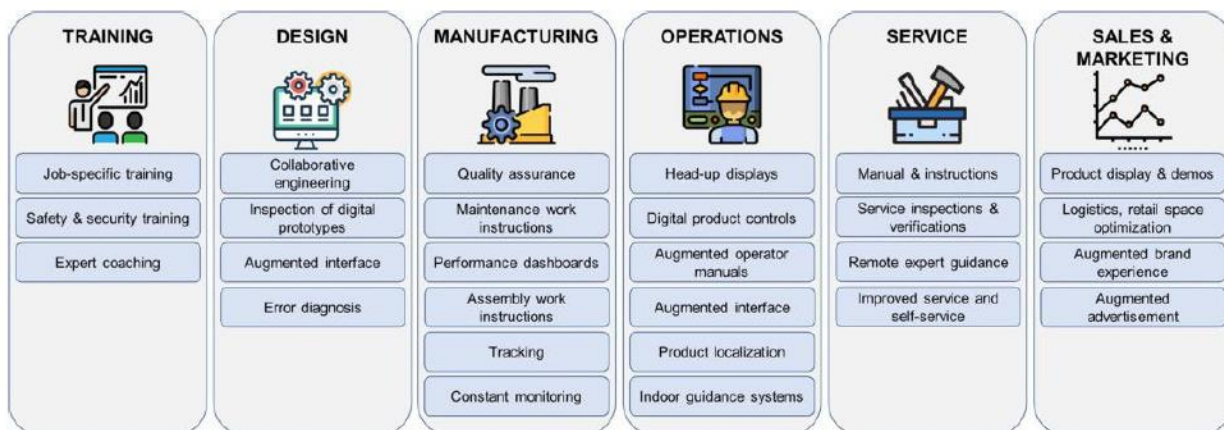


Figura 1.47 Applicazione della Realtà Aumentata nel contesto industriale (Alcácer and Cruz-Machado, 2019).

Mentre la realtà aumentata consiste nel creare un livello di realtà che si sovrappone a quello fisico attraverso l'uso di particolari dispositivi, la realtà virtuale permette di costruire un mondo completamente virtuale con cui interagire. Queste tecnologie possono essere di grande aiuto al mondo industriale in quanto permettono di simulare rapidamente scenari che altrimenti richiederebbero tempo e investimenti (Kipper and Rampolla, 2012).

La realtà virtuale e la realtà aumentata sono tecnologie preziose per semplificare le *operations* in quanto possono essere utilizzate per guidare gli operatori durante il loro lavoro. Inoltre, possono essere utilizzate come strumento per ridurre il *time-to-market* in fase di progettazione in quanto offrono la possibilità di impostare processi di produzione completamente simulati.

### 1.3.3.7 Robot autonomi

Oggi i sistemi robotici sono utilizzati in molte applicazioni, soprattutto per compiti altamente ripetitivi e possono essere eseguiti in ambienti e condizioni non adatte all'uomo.

I robot stanno diventando sempre più autonomi, flessibili e cooperativi, iniziando ad interagire tra loro e con l'ambiente e a lavorare in sicurezza vicino agli esseri umani (Oztemel and Gursev, 2020; Vaidya et al., 2018). Un robot autonomo è un robot flessibile che svolge i compiti assegnati in modo preciso e intelligente e quindi è in grado di supportare o sostituire l'operatore di fabbrica (Vaidya et al., 2018). Un esempio utilizzato nell'industria è lo YuMI® di ABB, un robot a due bracci per l'assemblaggio di piccole parti.

Inoltre, le nuove strategie nelle interazioni di interfaccia uomo-macchina stanno portando allo sviluppo di sistemi robotici collaborativi o sistemi "CoBot", che permettono di "sfruttare la forza e la resistenza dei robot con la flessibilità e il processo decisionale degli operatori umani" (Djuric et al., 2016). Mentre l'impiego convenzionale di robot industriali per automatizzare i processi di produzione è considerato economicamente più vantaggioso rispetto all'automazione dura e al lavoro manuale per una gamma media di lotti, l'uso di robot collaborativi, ammorbidendo i limiti dell'automazione robotica, permette una distribuzione dei compiti tra esseri umani e robot che amplia l'applicabilità dei robot per la produzione industriale (Fryman and Matthias, 2012).

### 1.3.3.8 Produzione additiva

Per soddisfare l'esigenza di una personalizzazione di massa, si stanno sviluppando nuovi metodi di produzione non tradizionali. La produzione additiva è in grado di soddisfare questa richiesta poiché può creare oggetti con attributi avanzati (nuovi materiali, forme complesse) e consentire una produzione delocalizzata, riducendo le distanze di trasporto e la necessità di magazzino (Dilberoglu et al., 2017; Vaidya et al., 2018).

*Additive Manufacturing* (AM) è un insieme di tecnologie che stampano modelli di prodotti CAD 3D. Originariamente la stampa 3D è stata utilizzata per sviluppare nuovi prototipi di prodotti (prototipazione

rapida), poi è stata applicata nel *Rapid Tooling*, e ora mira a produrre prodotti finali (*Rapid Manufacturing*) (Eyers and Potter, 2017). Il modello 3D è suddiviso in strati per mezzo di un software e successivamente realizzato con varie tecnologie. Le materie plastiche sono comunemente usate con queste tecnologie (Gunal, 2019). Nell'ambito dell'Industria 4.0, la ricerca scientifica si concentra anche sui materiali metallici e sui materiali e processi intelligenti. Un altro importante gruppo di materiali esaminati sono i cosiddetti *Smart Materials*. Questi materiali innovativi per la stampa adattano le loro proprietà nel tempo o in base ai cambiamenti delle condizioni esterne. Questi materiali possono essere utilizzati in ambienti estremi, strutture di montaggio o *Soft Robotics* (Dilberoglu et al., 2017).

### 1.3.3.9 Sicurezza informatica

Infine, la sicurezza informatica è una questione critica nel nuovo scenario industriale generato da questa rapida innovazione nell'archiviazione dei dati e nei dispositivi IT.

Infatti, la crescente quantità di dispositivi collegati significa maggiori possibilità di attacchi informatici. Varie proprietà, come la riservatezza, l'integrità, l'autenticazione, l'autorizzazione, il non ripudio, la disponibilità e la privacy, devono essere garantite al fine di fornire sicurezza ai sistemi IoT (Roy et al., 2016).

### 1.3.4 - Applicazione delle tecnologie 4.0 nella gestione della manutenzione

Al fine di fornire una panoramica delle opportunità derivanti dall'Industria 4.0 nel campo della gestione della manutenzione, i nove pilastri affrontati nella Sezione 1.3.3 sono elencati di nuovo, facendo riferimento al loro possibile utilizzo nelle attività di manutenzione, nella Tabella 1.13.

**Tabella 1.13**      **Panoramica dell'implementazione dell'Industria 4.0 nella gestione della manutenzione.**

PILASTRO TECNOLOGICO	OPPORTUNITÀ DI APPLICAZIONE NELLA GESTIONE DELLA MANUTENZIONE
<p><b>Internet of things (IoT)</b></p>	<p>Le soluzioni <i>Internet of Things</i> (IoT) in ambienti industriali possono portare allo sviluppo di sistemi innovativi ed efficienti che mirano ad aumentare l'efficienza operativa in una nuova generazione di fabbriche intelligenti. L'IoT consente la generazione di diversi tipi di strutture dati, che possono essere analizzate attraverso l'uso di tecniche di analisi avanzate (vedi <i>Big Data Analytics</i>). Inoltre, la connessione tra le macchine e la creazione di interfacce uomo-macchina promossa dall'IoT fornisce uno strumento per l'automazione delle contromisure di guasto. (Angelopoulos et al., 2020; Civerchia et al., 2017; Daponte et al., 2018; Peres et al., 2018; Xiaoli et al., 2011; Zhong et al., 2017).</p> <p>Infine, la presenza di dispositivi mobili può aiutare la supervisione delle operazioni manutentive, permettendo così agli operatori di manutenzione</p>

	di lavorare in modo più efficiente. (Cheng et al., 2018; Hao and Helo, 2017)
<b>Big data and analytics</b>	<p>I processi di manutenzione potrebbero trarre vantaggio dall'uso di modelli più complessi, come quelli di machine learning, nel tentativo di migliorare la politica di manutenzione e ridurre i tempi di fermo macchina.</p> <p>Infatti, a seconda della complessità dell'evoluzione del guasto, della disponibilità dei dati e della tecnica utilizzata, è possibile non solo rilevare i guasti negli asset ma anche operare un controllo diagnostico e/o prognostico, identificando la causa del guasto avvenuto ed anticipandolo (Carvalho et al., 2019; Karim et al., 2016; Lee et al., 2014; Popkova et al., 2019; Xiao et al., 2011; Xiao, 2016).</p> <p>L'analisi dei Big Data ha la potenzialità di portare diversi vantaggi, come il raggiungimento di tempi di fermo macchina quasi nulli e l'applicazione di politiche di manutenzione avanzate. (Kumar et al., 2019; Yan et al., 2017).</p>
<b>Simulazione</b>	<p>L'uso della simulazione può avvantaggiare i responsabili delle decisioni, sostenendo la scelta della giusta infrastruttura e delle politiche di manutenzione, fornendo informazioni sui costi-benefici delle potenziali alternative. (Adams et al., 2017; Kumar et al., 2018; Susto et al., 2018).</p> <p>Inoltre, la simulazione può essere utilizzata per prevedere il comportamento dei sistemi di produzione reali, supportando così le decisioni di pianificazione e manutenzione (Goodall et al., 2019; Kumar et al., 2018).</p>
<b>Cloud</b>	<p>La presenza del <i>cloud computing</i> consente l'utilizzo di modelli più complessi (migliore capacità di memorizzazione ed elaborazione dei dati) (Mourtzis and Vlachou, 2018; Oztemel and Gursev, 2020).</p> <p>La presenza di una piattaforma cloud può anche consentire l'archiviazione e la trasmissione in tempo reale di una massiccia quantità di dati (Vaidya et al., 2018).</p>
<b>Sistemi integrati</b>	<p>L'integrazione di tutti gli attori del sistema favorisce lo sviluppo di obiettivi di manutenzione con maggiore efficienza ed efficacia (Kumar et al., 2018; Peres et al., 2018).</p> <p>Questa ampia integrazione di tutte le parti rilevanti nel business e nei processi a valore aggiunto e il collegamento della produzione e dei servizi di alta qualità attraverso tutta la catena del valore può garantire nuove opportunità (Monostori et al., 2016), come l'implementazione di <i>Smart Contract</i>. In presenza di interfacce intelligenti, sarebbe possibile prevedere l'implementazione dell'esecuzione dello <i>Smart Contract</i> per gli interventi di manutenzione: una volta effettuato l'intervento, l'operatore manutentivo potrebbe interagire con la macchina che confermerebbe l'adempimento delle clausole del contratto, rilasciando automaticamente il pagamento (Daniotti et al., 2020).</p> <p>Inoltre, il servizio di manutenzione può essere effettuato anche da parti esterne (es. il costruttore) attraverso il monitoraggio continuo (Allmendinger and Lombreglia, 2005).</p>
<b>Realtà aumentata</b>	<p>L'esecuzione di attività come gli interventi di set-up e manutenzione potrebbe essere notevolmente migliorata dall'utilizzo della realtà aumentata al fine di supportare gli operatori e migliorare la loro sicurezza, ad esempio, fornendo una guida da parte di un operatore più esperto attraverso una connessione remota o rendendo disponibili manuali tecnici (Abramovici et al., 2017; Gattullo et al., 2019; Koch et al., 2017; Masoni et al., 2017; Oliveira et al., 2016).</p> <p>Inoltre, l'uso della realtà aumentata (o realtà virtuale) potrebbe fornire uno strumento più efficace per la formazione di nuovo personale, fornendo così la base di conoscenza necessaria per i compiti futuri.</p>

<b>Robot autonomi</b>	L'uso di robot potrebbe ottimizzare l'esecuzione di specifiche attività, aumentandone la velocità e la qualità (Oztemel and Gursev, 2020). Inoltre, l'interazione uomo-robot può consentire un'esecuzione più sicura delle attività di manutenzione (Koch et al., 2017).
<b>Produzione additiva</b>	La produzione additiva gioca un ruolo interessante nell'evoluzione della manutenzione, poiché la sua implementazione potrebbe consentire di creare pezzi di ricambio quando e dove sono necessari per ridurre i tempi di fermo macchina [Khajavi et al. 2014; Dilberoglu et al. 2017].
<b>Sicurezza informatica</b>	Essendo un tema importante nell'Industria 4.0, la <i>cyber-security</i> è essenziale nella realtà industriale, indipendentemente dalle specifiche attività analizzate e la sua implementazione e miglioramento continuo è fondamentale per garantire la sicurezza di ogni membro dell'organizzazione.

### 1.3.5 - DCS: Distributed Control System

Nell'ottica di sviluppo di un sistema di manutenzione predittiva, come descritto nei paragrafi precedenti, è quindi necessario focalizzarsi anche sul suo funzionamento in real-time per controllo, diagnostica e prognostica dei componenti di impianto. La soluzione sviluppata dovrà quindi essere implementata sui software gestionali di impianto per essere utilizzata con i dati provenienti dai sensori sul campo.

Nell'ambito della digitalizzazione delle industrie, i sistemi di controllo distribuiti (*Distributed Control System* – DCS) rappresentano una delle soluzioni più adottate per il controllo dei processi che riguardano i grandi impianti continui (ad es. le raffinerie, gli impianti chimici, le centrali di produzione di energia...).

Un DCS è un sistema di controllo automatico, suddiviso a sua volta in più sottosistemi, ognuno con il proprio compito ma con la capacità di interagire reciprocamente. Cioè, non vi è un solo computer che controlla tutto il sistema, ma diversi controllori suddivisi in base alla sezione di impianto. Questo consente di svolgere in modo integrato entrambe le funzioni che normalmente competono ai PLC ('Programmable Logic Computer') e agli SCADA ('Supervision Control and Data Acquisition'): ovvero rispettivamente quelle di elaborare i segnali provenienti dai sensori dell'impianto, acquisire e controllare i dati. Unire i due aspetti all'interno di una sola struttura, comporta un risparmio di costi, migliora l'operatività dell'impianto e facilita la formazione dei suoi operatori. Quest'ultima, infatti, smette di essere legata al singolo componente, bensì può essere erogata sotto forma di un unico percorso standard.

L'architettura DCS tipica prevede una struttura gerarchica a 5 livelli, che comprendono il sistema da controllare (livello 0, detto anche "campo") e quattro livelli di controllo:

- Livello 1 o "controllo diretto": costituito da controllori, controllori logici programmabili (PLC), *Burner management system* (BMS) e sistemi di spegnimento di emergenza (ESD).
- Livello 2: costituito dalla supervisione e dal comando degli operatori.

- Livello 3 o "controllo di produzione": costituito dai sistemi avanzati di controllo ottimo e storicizzazione.
- Livello 4: costituito dal computer centrale.

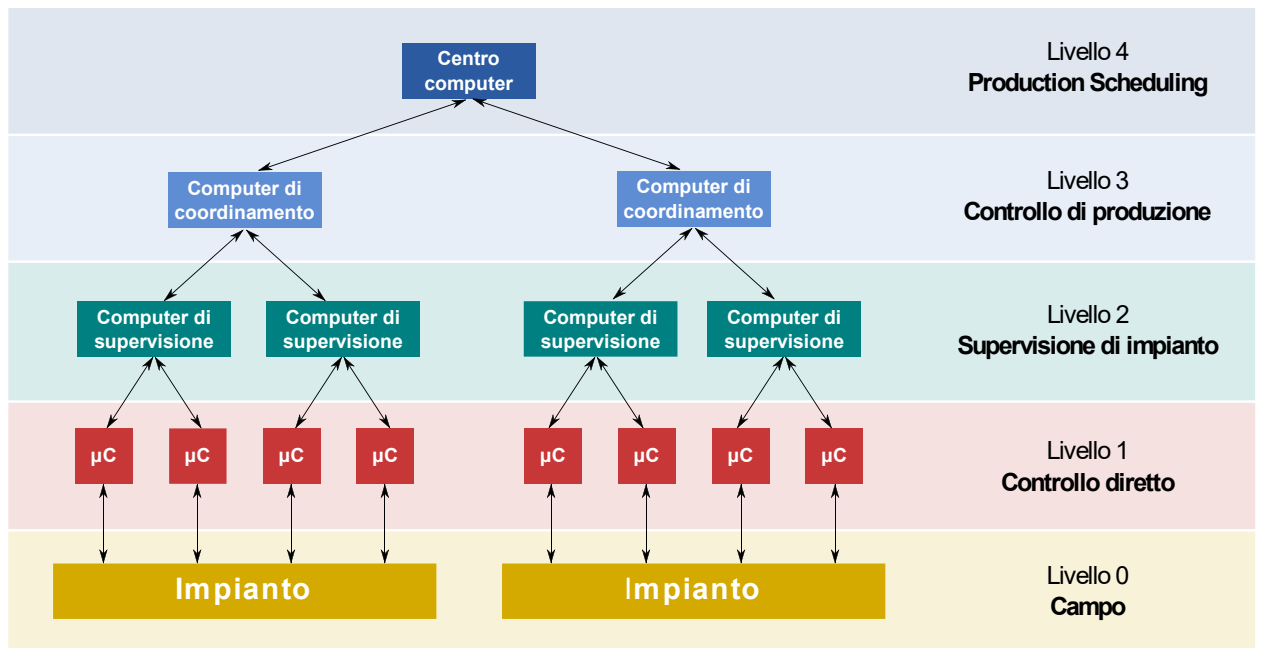


Figura 1.48 Struttura DCS.

### Livello 0: Campo

Con "campo" si intende l'insieme delle variabili di processo PV (*process value*) e delle variabili di controllo SP (*set point*) e OP (*output*) che rappresentano l'impianto oggetto del sistema di controllo. A livello di Campo abbiamo tutta l'interazione fisica con la strumentazione d'impianto che si vuole controllare, formata da attuatori (valvole, motori, serrande ecc.) e da sensori. Generalmente lo scambio di informazioni (acquisizione di misure o comando verso gli attuatori) tra Livello 1 e Livello 0 avviene tramite segnali elettrici.

### Livello 1: Controllori e PLC

Sono sistemi computerizzati in remoto, atti ad acquisire le variabili di processo dal campo e calcolare automaticamente tramite semplici logiche come ad esempio controllori PID o carte di controllo le variabili di controllo. L'acquisizione avviene tramite appositi moduli di acquisizione e pubblicati sul livello 2 tramite *control net* o rete seriale come ad esempio il modbus.

A questo livello avviene il cuore del sistema di controllo ed automazione di impianto o del sistema di sicurezza. Le regolazioni e le logiche che automatizzano l'impianto sono implementate nei DCS o in PLC.

La scelta tecnologica dipende come sempre dallo scopo dell'applicazione, quindi per un sistema di regolazione piccolo si tenderà a scegliere PLC general purpose, mentre per sistemi di sicurezza che devono garantire un certo livello di sicurezza integrata necessariamente bisognerà scegliere dei PLC che soddisfino questi requisiti. Il DCS è tipicamente utilizzato per sistemi di regolazione grandi ed integrati.

## **Livello 2: Supervisione e comando**

A questo punto entriamo al livello dell'operatore di controllo di impianto, A questo livello il DCS si occupa della lettura delle variabili di processo provenienti dal campo PV (process value) e del comando delle variabili di controllo SP (set point) e OP (output). L'operatore, tramite interfaccia uomo-macchina, ha il controllo e la supervisione del campo.

Generalmente il livello 2 è composto da uno SCADA o da un sistema HMI integrato al DCS. Gli elementi di un sistema di supervisione sono l'interfacciamento con il campo tramite:

- pagina grafica, composta di una parte statica rappresentante l'impianto e una parte dinamica rappresentante le misure di processo e lo stato degli attuatori. Pagine grafiche specifiche per i sistemi ESD e F&G possono essere tabellari o sotto forma di *logic diagram* semplificati
- gruppi operativi, pagine contenenti le misure di un gruppo di strumenti
- *trend*, pagine che mostrano gli andamenti delle misure rispetto al tempo in un determinato periodo
- *alarm logger*, pagina che riporta i principali allarmi di sistema e di processo con le rispettive priorità di allarme (low, high, emergency)
- *event logger*, pagina che riporta i principali eventi (allarmi e azioni da operatore)

## **Livello 3: Sistemi avanzati di controllo ottimo e storicizzazione**

Il livello 3 accede ai dati presentati sul livello 2 e li elabora per storicizzarli a lungo termine (con campionamento definito da utente), per generare strategie di controllo avanzate basate sulla storicizzazione o per sistemi di addestramento per operatori basati su simulazione dinamica di eventi imprevisi in impianto.

## **Livello 4: Office**

Raccolta dati per sistemi di forecasting remotizzati, tramite collegamento sicuro *read-only* con accesso al livello 3 via DMZ. Il livello office è usato per lo più a scopo statistico.



(Bolzern et al., 2008; Green and Perry, 2019; Ogata and Yang, 2002; Popovic and Bhatkar, 1990)

Nel capitolo “Caso studio” vedremo come i modelli diagnostici sono stati integrati nel DCS di impianto utilizzato.

## 1.4 - Domande di ricerca

In questo contesto, l'obiettivo principale del progetto di ricerca è quello di rivedere ed estendere le attuali conoscenze sulle metodologie prognostiche per le modalità di guasto dei componenti degli impianti industriali. Si intende inoltre definire una guida per le aziende che comprenda indicazioni su: vantaggi e svantaggi, problemi e opportunità, tecnologie abilitanti e una metodologia per l'analisi e l'applicazione dei metodi prognostici individuati.

Da questo punto di vista, le domande di ricerca possono essere riassunte come segue:

1. Quali sono i modelli diagnostici e prognostici adottati per componenti di impianti industriali? Quali metodologie adottare in funzione della tipologia di componente?
2. Quali passi devono essere seguiti per applicare prognostica in impianto? Su quali componenti dovrebbe essere focalizzata l'analisi?
3. Quali sono le tecnologie chiave per l'implementazione di queste metodologie?

Come descritto nei capitoli precedenti, la profonda analisi della letteratura riguardante sia le tecniche prognostiche che la manutenzione nel contesto dell'industria 4.0 ha già fornito una risposta parziale alle domande di ricerca qui indicate. L'applicazione di quanto appreso dalla letteratura ad un caso studio, permetterà di estendere quanto analizzato ed approfondire ulteriormente le domande di ricerca calandole in un reale contesto industriale.

## 2 - Metodi

In questo capitolo verrà presentata la metodologia utilizzata per rispondere alle 3 domande di ricerca.

Una prima analisi della letteratura, approfondita nel capitolo “Introduzione” ha permesso di rispondere in parte già alle domande di ricerca, si è infatti costruito sulla base di quanto studiato in letteratura un modello generale per l’approccio alla prognostica, che è stato poi rielaborato in occasione dell’applicazione pratica al caso studio e presentato in dettaglio nei paragrafi successivi.

### 2.1 - Approccio alla prognostica

Dall’analisi della letteratura e dall’esperienza maturata durante il progetto caso studio che approfondiremo nel prossimo capitolo, emerge che il corretto approccio alla prognostica segue le fasi riportate nella seguente figura:



Figura 2.1 Approccio alla prognostica.

Come mostrato in figura le fasi sono le seguenti:

1. **Analisi dei modi di guasto dei componenti:** grazie a questa fase è possibile individuare i componenti critici di impianto ed i relativi modi di guasto che necessitano maggiore attenzione e

che richiedono un maggior controllo. In questa fase si individuano inoltre i modi di guasto su cui è possibile sviluppare modelli prognostici e quali sono le variabili spia del modo di guasto da monitorare per effettuare prognostica;

2. **Sensorizzazione dei macchinari:** questo secondo passo suggerisce di installare nuovi sensori (se non già presenti in impianto) che permettano di monitorare tutte le variabili spia di interesse individuate al passo precedente. È inoltre necessario assicurarsi che le specifiche tecniche dei sensori (frequenza di campionamento dati, temperature di funzionamento, accuratezza ecc.) siano compatibili con le analisi da effettuare e le caratteristiche del macchinario da monitorare;
3. **Analisi e sviluppo del modello diagnostico:** in questa fase viene scelto e sviluppato il modello diagnostico più consono all'individuazione della classe di funzionamento del modo di guasto in analisi in funzione dei dati delle variabili spia individuate. Tale modello diagnostico oltre a classificare il comportamento del componente deve essere in grado di fornire ad output un parametro proxy delle prestazioni del macchinario in analisi.
4. **Analisi e sviluppo del modello di previsione della RUL:** In questa fase viene scelto il modello di analisi di serie temporali necessario alla prognostica del modo di guasto di interesse. Il modello costruito verrà applicato alla variabile proxy delle prestazioni del macchinario (output del modello diagnostico) in modo da prevedere quando queste saranno inferiori ad una soglia di accettabilità preimpostata.
5. **Schedulazione degli interventi manutentivi:** dalla conoscenza della vita utile residua del macchinario in analisi, output del modello di previsione della RUL, si procede a scrivere a calendario la data prevista dell'evento di guasto con informazioni circa la confidenza della previsione effettuata e la criticità del modo di guasto in base a quanto studiato nella prima fase. In questo modo si danno tutte le informazioni necessarie all'ingegneria di manutenzione per schedulare gli interventi manutentivi sul campo.

## 2.2 - Modello prognostico

Una volta definito l'approccio generale alla prognostica si è entrati più nel dettaglio della metodologia e delle fasi da seguire per costruire il modello prognostico, corrispondente alle fasi 3 e 4 descritte nel precedente paragrafo.

L'analisi della letteratura e l'applicazione dei modelli al caso studio hanno portato alla definizione della seguente metodologia per lo sviluppo di un sistema prognostico in real-time.

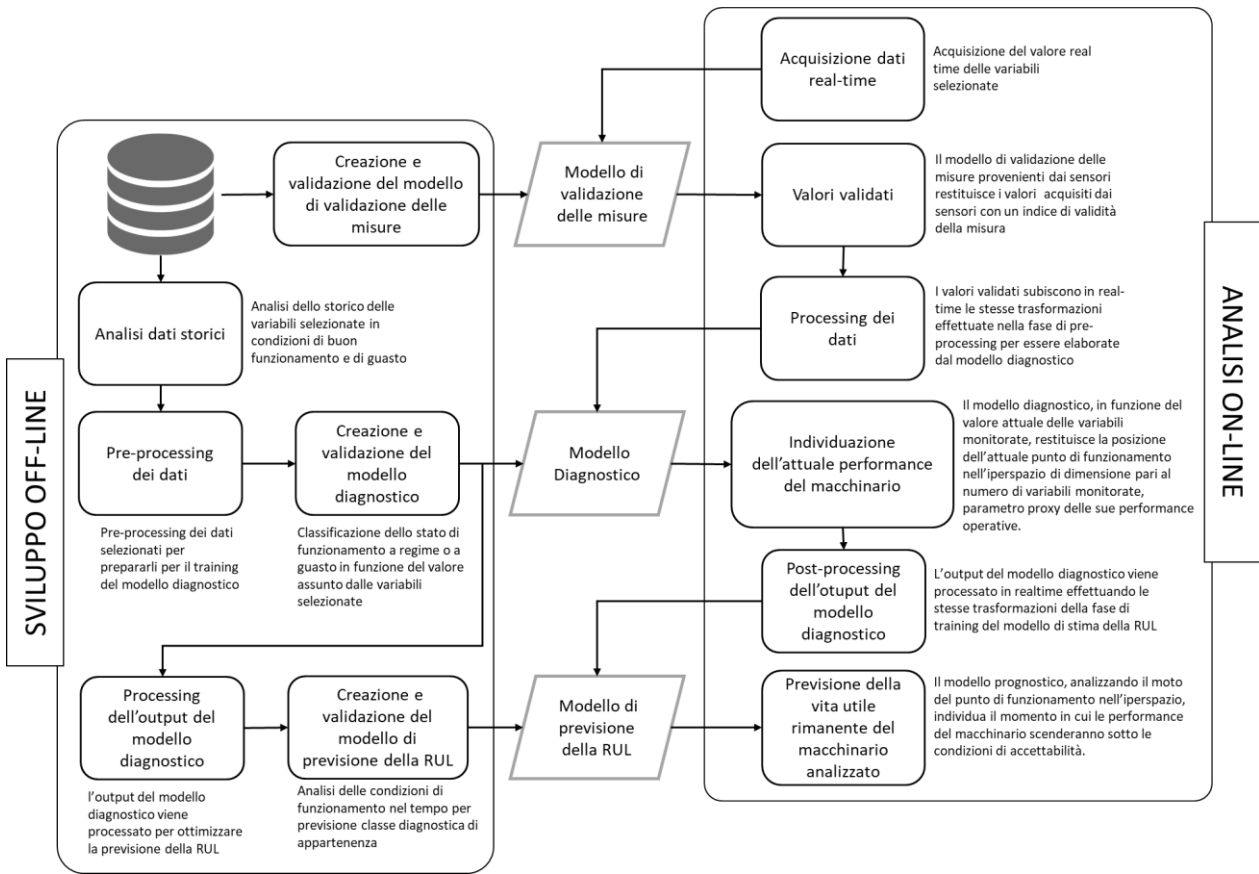


Figura 2.2 Framework del modello prognostico real-time.

Come mostrato in figura, la metodologia prevede uno sviluppo off-line dei tre modelli necessari alla prognostica:

1. Modello di validazione delle misure;
2. Modello diagnostico;
3. Modello di previsione della RUL;

ed una macro-fase di analisi on-line in tempo reale dei dati acquisiti dai sensori ed elaborati dai modelli addestrati.

A partire da un ricco database di impianto si effettua una prima analisi dei dati storici in esso contenuti allo scopo di selezionare le variabili spia del modo di guasto di interesse nei periodi sia di funzionamento corretto che in quelli di funzionamento anomalo connesso al particolare modo di guasto di interesse.

Successivamente si passa alla fase di preprocessing dei dati a database, allo scopo di:

1. Sincronizzare i dati per ottenere un set di training dei modelli coerente;

2. Trasformare i dati in modo da esaltare il loro contenuto informativo (ad esempio: filtro passa alto/basso, trasformata di Fourier per passare nel dominio delle frequenze, Wavelet ecc.)
3. Creare il set di dati necessario all'addestramento dei modelli.

Viene quindi selezionato, sviluppato, addestrato e validato sui dati storici il modello diagnostico che sarà utilizzato in real-time allo scopo di:

1. Individuare la classe di funzionamento del macchinario (regime o incipiente guasto);
2. Individuare lo stato attuale di salute del componente.

Se necessario al secondo output del modello diagnostico, viene effettuato una nuova trasformazione al fine di ottimizzare il potere previsionale del modello di stima della RUL. A questo punto si passa alla selezione, sviluppo ed addestramento anche di questo modello sui dati presenti a storico.

Parallelamente, nella fase di sviluppo e validazione off-line dei modelli necessari alla prognostica, i dati storici provenienti dai sensori vengono utilizzati allo scopo di sviluppare il modello di validazione delle misure provenienti dai sensori, necessario ad assicurare che tutti i dati processati dai modelli siano da considerarsi affidabili e che non ci siano quindi guasti/malfunzionamenti o starature ai sensori.

Per quanto riguarda la fase di analisi on-line del modello prognostico costruito, si procede all'acquisizione real-time dei dati delle variabili spia del guasto provenienti dai sensori. Questi vengono dapprima validati dal modello di validazione delle misure e successivamente processati e trasformati, se necessario, prima di essere elaborati dal modello diagnostico addestrato nella fase di sviluppo off-line.

Una volta processati ed elaborati dal modello diagnostico si ottengono le seguenti informazioni: classe di funzionamento del componente analizzato (regime o incipiente guasto) e l'indice delle performance attuali del componente. A questo punto si effettua una nuova trasformazione di quest'ultimo output identica a quella effettuata nella fase di addestramento off-line del modello di stima della RUL. L'ultima fase è quindi quella di elaborazione della variabile proxy delle performance del macchinario per ottenere la previsione del suo futuro valore ed impostando una soglia di minima accettabilità delle performance del componente, ottenere la data prevista di guasto del componente.

Mettendo insieme le informazioni provenienti dal modello diagnostico (tipo di guasto in cui si incorre) con quelle del modello di stima della RUL si ottiene così un completo modello prognostico, che permette la schedulazione agevolata degli interventi manutentivi di impianto.

## 2.3 - Modello di stima della RUL in assenza di dati di guasto

Come descritto nel precedente paragrafo, lo sviluppo di un sistema di manutenzione predittiva necessita di dati relativi al comportamento dell'asset analizzato durante la sua evoluzione verso il fenomeno di guasto per poterlo analizzare e prevedere. Molti dei componenti critici degli impianti industriali, hanno una elevata affidabilità e quindi non presentano dati storici di guasto su cui potersi basare per prevedere la RUL. In tal caso si trovano in letteratura tecniche diagnostiche basate sullo *Statistical Process Control* (SPC) come le carte di controllo che si limitano a segnalare comportamenti anomali delle variabili analizzate.

Anche in questo contesto si è sviluppato un modello concettuale di previsione della vita utile del componente analizzato (RUL) sulla base di due modelli analitici:

1. Modello di *state detection*: ha il compito di elaborare le variabili connesse alle prestazioni del macchinario in analisi e distillare un indice di salute del componente;
2. Modello di previsione della RUL: ha il compito di prevedere l'andamento futuro del parametro distillato ed indicare quando questo scenderà sotto la soglia di accettabilità.

Il framework ideato segue le stesse fasi del modello prognostico presentato al paragrafo precedente con la differenza che:

- Non si parla più di modello diagnostico ma di *state detection*, in quanto non avendo a disposizione dati relativi al funzionamento antecedente al fenomeno di guasto del macchinario, non possiamo a priori associare un abbassamento delle prestazioni ad un particolare modo di guasto, se non dopo un'attenta analisi da parte dell'ingegneria della manutenzione degli andamenti delle variabili in input del modello stesso;
- Il modello non sarà un modello di classificazione del funzionamento del componente (regime, anomalia) ma bensì un modello addestrato sui soli dati di regime ed in grado di indicare una variazione delle condizioni di funzionamento da quelle di addestramento (come ad esempio le carte di controllo multivariate o carte di Hotelling o lo One-Class SVM);

Nella figura si riporta il framework sviluppato per la previsione della RUL in caso di mancanza di dati relativi al guasto del componente.

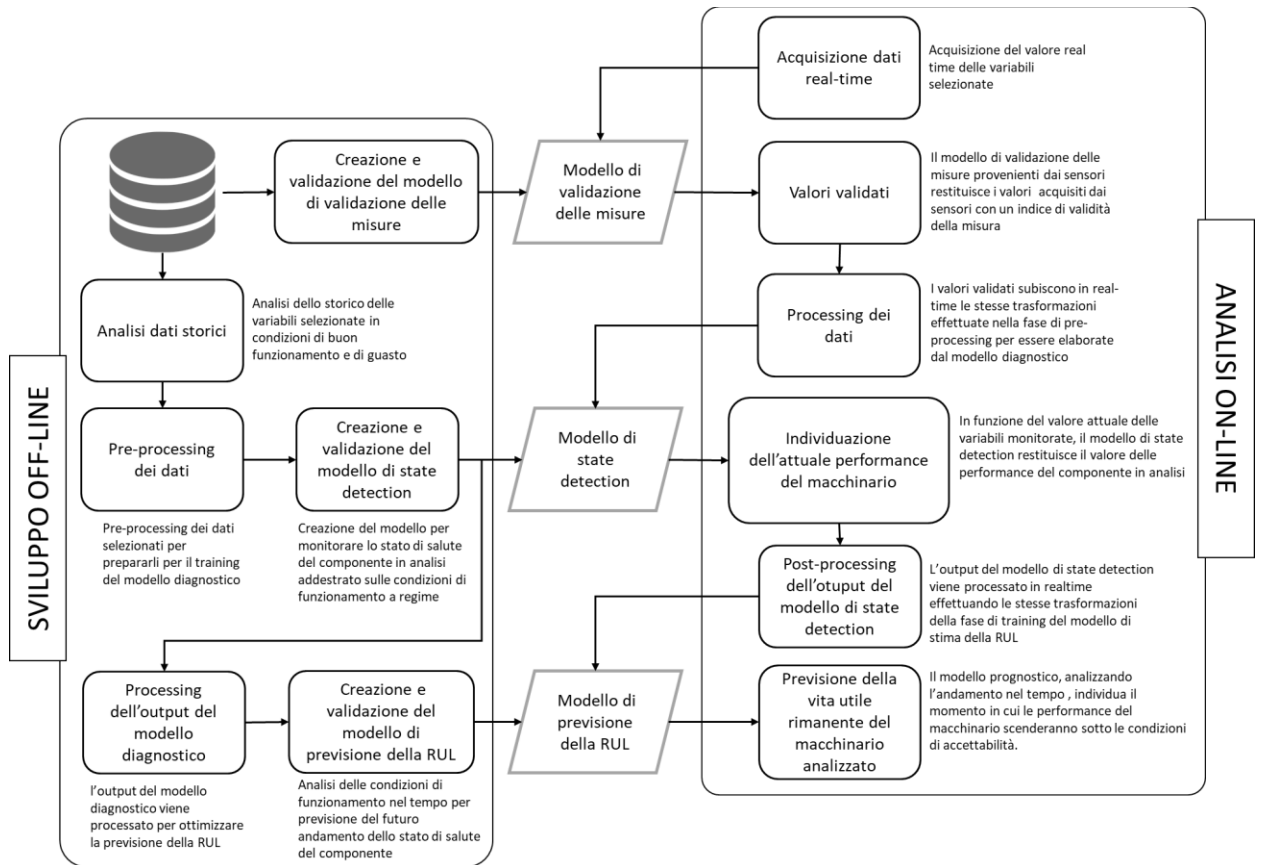


Figura 2.3 Framework del modello di stima della RUL in assenza di dati di guasto.

## 2.4 - Architettura informatica

Una volta costruita la metodologia real-time di sviluppo ed analisi dei modelli presentati, ci siamo chiesti a che livelli dell'architettura informatica di impianto debbano essere rilasciati per consentire il loro funzionamento efficace.

Come descritto nel paragrafo 1.3 -, i DCS di impianto sono tipicamente costruiti su 5 differenti livelli, dall'acquisizione delle misure di interesse sul campo all'analisi di informazioni aggregate al livello più alto. Più aumenta il livello e minore è il dettaglio sui dati acquisiti dai sensori, inoltre, maggiore è il livello e maggiore è la potenza di calcolo a disposizione per effettuare analisi sui dati.

Per definire come suddividere le metodologie presentate sui vari livelli di impianto, è necessario suddividere le caratteristiche prognostiche dei possibili modi di guasto in due categorie:

1. Modi di guasto che richiedono un monitoraggio hard real-time
2. Modi di guasto che richiedono un monitoraggio soft real-time

Appartengono alla prima categoria quei modi di guasto o fenomeni di perdita di prestazioni del macchinario che hanno una rapida evoluzione nel tempo e che necessitano di un intervento immediato non appena diagnosticati da parte dell'automazione di impianto.

Appartengono invece alla seconda categoria quei modi di guasto, tipici dell'usura e/o corrosione di componenti meccanici, che presentano una fenomenologia di degrado di lunga durata e che necessitano di interventi manutentivi da parte degli operatori di manutenzione.

Per l'una e l'altra categoria, è necessario implementare i modelli a diversi livelli del DCS per garantire l'efficacia degli interventi, automatici o tramite personale dedicato.

Una ulteriore riflessione necessaria riguarda invece la tipologia di misura acquisita, che a sua volta determina la necessità o meno di spingersi ad operare a livelli più bassi del DCS. In particolare, anche in questo caso le misure possono essere suddivise in due macro-famiglie:

1. Misure che richiedono un pre-processing sui dati *raw* per poterne cogliere il contenuto informativo;
2. Misure che non richiedono un pre-processing sui dati *raw*.

Appartengono alla prima categoria tutte le misure il cui contenuto informativo riguardo allo stato di salute del macchinario sono visibili solo se analizzati ad alta frequenza di campionamento ed esplicate al meglio dopo una trasformazione dei dati. Ad esempio, il valore grezzo così come acquisito da un sensore di vibrazione non ha un elevato contenuto informativo quanto una sua trasformazione nel dominio delle frequenze (ad esempio utilizzando una trasformazione FFT o Wavelet) e quindi è necessario trasformarle a basso livello del DCS (ad esempio livello 1 PLC e sistemi di controllo) così da inserire tra le variabili spia del modo di guasto il loro contenuto di ampiezza e frequenza. Solitamente queste misure una volta effettuata la trasformazione ad alta frequenza non perdono di contenuto informativo se storicizzate ad un tempo di campionamento relativamente alto (secondi, minuti o addirittura ore) poiché una loro variazione sostanziale avrà tempi paragonabili al processo di usura dovuto al fenomeno di guasto.

Appartengono invece alla seconda categoria tutte le misure che sono informative già così come sono acquisite dal sensore. Tipicamente queste grandezze presentano ad alta frequenza solo delle oscillazioni intorno al valore atteso e le loro variazioni sostanziali sono apprezzabili ad intervalli di tempo relativamente lunghi (secondi, minuti, ore). Per queste grandezze la fase di pre-processing, se necessaria, non ha bisogno di essere effettuata a livello 1 del DCS ma può essere spostata a livelli più alti ed effettuata direttamente sui dati storicizzati, dato che il campionamento dovuto alla storicizzazione del dato non inficia il loro contenuto informativo.



Nel caso si abbia a che fare con misure appartenenti alla prima categoria è necessario prevedere a monte della costruzione dei modelli diagnostici e/o prognostici una fase di trasformazione dei dati al livello 1 del DCS per poter poi analizzare i dati storicizzati con il giusto contenuto informativo.

Nella tabella seguente si riportano: il livello di DCS in cui sono rilasciati i modelli sviluppati e dove avvengono le fasi di analisi on-line descritte in Figura 2.2 Framework del modello prognostico real-time. e Figura 2.3 Framework del modello di stima della RUL in assenza di dati di guasto. suddivise per modelli soft real-time ed hard real-time. In entrambi i casi le fasi di sviluppo degli algoritmi possono essere effettuate anche separatamente su PC dedicati o a livello 3 del DCS dove si ha una elevata potenza di calcolo. La differenza è che nel caso in cui si costruiscano modelli hard real-time che necessitano di misure con campionamento più alto di quello di storicizzazione, è necessario prevedere una fase di acquisizione dei dati raw al posto della prima fase di estrazione dei dati dal database di impianto.

**Tabella 2.1 .Livelli di rilascio dei modelli e di analisi on-line dei dati**

	FASI DELLA METODOLOGIA	LIVELLO DCS	
		SOFT REAL-TIME	HARD REAL-TIME
MODELLI	Modello di validazione delle misure	Livello 3: Controllo di produzione	Livello 1: Controllo diretto
	Modello diagnostico/state detection	Livello 3: Controllo di produzione	Livello 1: Controllo diretto
	Modello di previsione della RUL	Livello 3: Controllo di produzione	NA
ANALISI ON-LINE	Acquisizione dati real-time	Livello 3: Controllo di produzione	Livello 1: Controllo diretto
	Valori validati	Livello 3: Controllo di produzione	Livello 1: Controllo diretto
	Pre-processing dei dati	Livello 3: Controllo di produzione	Livello 1: Controllo diretto
	Individuazione dell'attuale performance del macchinario	Livello 3: Controllo di produzione	Livello 1: Controllo diretto
	Post-processing dell'output del modello di state detection	Livello 3: Controllo di produzione	NA
	Previsione della vita utile rimanente del macchinario analizzato	Livello 3: Controllo di produzione	NA

Come descritto in tabella, nel caso in cui si abbia a che fare con modi di guasto o perdite di prestazione che necessitano di modelli hard real-time la fase prognostica è inutile da implementare in quanto non si necessita di una schedulazione di un intervento manutentivo ma si agisce direttamente a livello di controllo di impianto, regolando il funzionamento del macchinario o bloccandolo.

Per implementare modelli diagnostici e/o di *state detection* hard real-time è necessario munirsi di PLC di ultima generazione che permettono di implementare modelli analitici complessi come modelli di *machine learning* o statistici avanzati. Ad esempio, i PLC Siemens SIMATIC STEP 7 supportano modelli sviluppati in Simulink PLC Coder, sfruttando quindi la potenza di MATLAB nell'implementazione di modelli analitici avanzati. In entrambi i casi, la fase di re-training dei modelli, che può essere necessaria quando si hanno a disposizione nuovi dati utili all'addestramento del modello diagnostico o di *state detection*, è comunque da prevedere su PC dedicati o di alto livello del DCS, che rilasceranno poi i parametri dei modelli aggiornati sui PLC.

## 2.5 - Modelli utilizzati nel caso studio

In questo capitolo si riporta la teoria di tutti i modelli costruiti e validati per il caso studio analizzato, la cui implementazione in impianto sarà poi affrontata nel capitolo 3 -.

### 2.5.1 - Validazione delle misure acquisite dai sensori

Prima ancora di passare in rassegna le problematiche e l'analisi dei modi di guasto e delle criticità dei componenti di impianto, dato che per l'implementazione di modelli diagnostici e prognostici ci baseremo in gran parte sui dati real-time provenienti dai sensori di impianto, si rende necessario costruire un modello analitico che permetta di validare le misure dei sensori presenti sul campo.

Tutto questo permette di ottenere due importanti risultati:

1. Definire preventivamente malfunzionamenti dei sensori;
2. Evitare di effettuare analisi diagnostiche e prognostiche su misure non valide, evitando quindi il problema del "Garbage In Garbage Out" (GIGO), ovvero dell'elaborazione di dati "spazzatura" che chiaramente forniranno in uscita informazioni fuorvianti.

Solitamente, nei punti critici dell'impianto si prevede l'utilizzo di sensori ridondati in modo tale che la misura sia calcolata sfruttando i dati provenienti dalle 3 fonti. Nel caso in cui uno dei sensori fornisca una misura significativamente differente da quella che risulta dagli altri 2, allora è previsto un algoritmo che esclude dal calcolo tale dispositivo, che si potrebbe essere guastato, passando ad una logica di misura su 2 valori anziché su 3.

Il problema sorge però nel momento in cui devo valutare la bontà di questi 2 sensori rimasti, dal momento che, in caso forniscano dati discordanti, non potrò mai sapere quale dei 2 stia effettivamente

dicendo il vero. La probabilità però che si guastino contemporaneamente 2 sensori su 3 con modi di guasto indipendenti è in pratica molto remota. La probabilità invece che ci sia un modo di guasto comune ai 3 sensori può non esserlo.

Decisamente più complicato è il caso in cui i sensori sono ridondati solo una volta, ovvero che da progetto ci siano 2 soli sensori con lo stesso scopo, oppure quando non è proprio stata prevista la ridondanza. In questi casi risulta evidentemente molto più complicato e allo stesso tempo critico capire la bontà della misura.

### 2.5.1.1 Sensori ridondati

In letteratura sono presenti varie tecniche che vanno dall'applicazione di principi statistici basati sullo storico dei dati, metodi che sfruttano modelli predittivi e controllano quanto la misura reale va a discordarsi da quella prevista, o anche tecniche che sfruttano le potenzialità delle reti neurali e del *machine learning* (Dorr et al., 1997; Sharma et al., 2010). Noi abbiamo raccolto la conoscenza presente ma abbiamo deciso di rivisitarla per creare un metodo che fosse computazionalmente non complesso e potesse avere una buona efficacia di riconoscimento dei guasti.

Prima di procedere con la spiegazione del metodo che intendiamo applicare per riconoscere le anomalie ai sensori ridondati, si osservi che:

- tra questi sensori, la probabilità di subire guasti per cause indipendenti tra loro è bassa, mentre è molto più probabile che ci sia un modo di guasto comune a tutti e 3
- i sensori, che sono identici, misurano lo stesso fenomeno e, perciò, si può supporre che il loro valore di output dipenda solo dalla grandezza osservata, dalla taratura del dispositivo e dal suo segnale di disturbo casuale.

La conclusione, quindi, è che ci si aspetta che due dispositivi funzionanti diano luogo in media allo stesso valore.

A causa delle osservazioni appena fatte, il nostro metodo di riconoscimento anticipato delle anomalie ai sensori prevede di studiare la differenza tra i valori rilevati, verificando che questa si mantenga statisticamente inalterata nel tempo.

Un possibile guasto viene identificato ad ogni istante  $t$  in cui la differenza tra due sensori esce fuori da un range stabilito. Trattandosi di variabili continue a media e varianza stazionarie, il range viene imposto a priori uguale alla media della misura più 3 volte la deviazione standard di questa misura, calcolate

durante un regime di funzionamento senza anomalie. Questa scelta aiuta a contenere gli errori di I specie, ovvero il rischio di rifiutare l'ipotesi nulla anche quando è vera.

In ogni istante, devono essere verificate le relazioni seguenti, dove il valore a destra dell'equazione rappresenta il limite di controllo superiore (UCL):

$$(A - B)_t \leq \overline{A - B} + 3s_{A-B}$$

$$(A - C)_t \leq \overline{A - C} + 3s_{A-C}$$

$$(B - C)_t \leq \overline{B - C} + 3s_{B-C}$$

Quando si hanno dei superamenti di soglia, l'obiettivo diventa quello di capire quale/i siano i dispositivi che in quell'istante stanno fornendo la lettura più verosimile per la grandezza osservata. Quindi, per mezzo di una tabella della verità (Tabella 2.2), abbiamo distinto delle categorie di situazioni degradate, dove ciascuna indica il valore da prendere come valore finale.

**Tabella 2.2** La tabella della verità mostra 4 categorie di situazioni degradate, che si hanno in base a quante coppie di sensori superano la soglia. Per ogni caso, è riportato il valore da preferire come finale e lo stato di qualità di tale valore.

$(A - B)_t > \text{UCL}$	$(A - C)_t > \text{UCL}$	$(B - C)_t > \text{UCL}$	Val_Finale	Categoria	Val_Verosimile	Qualità
X	Y	Z	X v Y v Z			
F	F	F	F	1	mediana	Healthy (h)
V	F	F	V	3	media pesata	h+warning
V	V	F	V	2	media tra B e C	h+allarme
V	F	V	V	2	media tra A e C	h+allarme
F	V	F	V	3	media pesata	h+warning
F	V	V	V	2	media tra A e B	h+allarme
F	F	V	V	3	media pesata	h+warning
V	V	V	V	4	no informazioni	No Output

Se nell'istante di valutazione:

1. Tutte e le 3 coppie di differenze **non** superano il limite imposto, allora si ritiene che la mediana tra le 3 pressioni sia il valore più verosimile di output.

2. Due coppie di differenze sono al di sopra della soglia, allora la loro intersezione identifica chiaramente il sensore difettoso e la pressione viene calcolata come media tra i 2 sensori “sopravvissuti”.
3. Una sola coppia esce fuori dal range, allora si fa una media pesata tra le tre pressioni, assegnando peso maggiore al sensore che non appartiene alla coppia fuori controllo: in modo empirico prendiamo 0,6 per il peso più grande e 0,2 per gli altri due.
4. Tutte le coppie superano contemporaneamente il limite superiore, allora non disponiamo di informazioni sufficienti per identificare il sensore che sta avendo un comportamento diverso da quello dei “fratelli”.

Un esempio del caso due può essere quando sia la differenza (A – B) che (A – C) eccedono il valor limite. Il sensore A, dato dalla loro intersezione, probabilmente si è guastato e, perciò, esso viene escluso dal calcolo della pressione finale. Supponiamo, invece, che solo la coppia AC stia registrando un valore fuori controllo, allora la nostra tecnica suggerisce di calcolare il valore acquisito come segue:

$$\text{Valore}_f = 0,2 * \text{Sensore}_A + 0,6 * \text{Sensore}_B + 0,2 * \text{Sensore}_C$$

Il modello di monitoraggio proposto, oltre a specificare quale valore di output debba essere considerato il più rappresentativo per la grandezza misurata, gli attribuisce un certo livello di qualità. In questo modo, l’utente può essere informato tramite misurazioni on-line sia del valore che della bontà del valore ottenuto, il quale in alcuni casi sarà anche corredato da uno ‘warning’ o da un allarme.

In particolare, ad ogni categoria abbiamo assegnato i seguenti stati di qualità (e aggiunto eventuali notifiche per gli operatori):

1. Healthy: la misura risultante può essere ritenuta affidabile. Infatti, tutti e 3 i sensori stanno restituendo dei valori all’interno della regione di controllo;
2. Healthy + Error: la misura è ancora buona, ma è necessario produrre un allarme che indichi quale sensore è difettoso e la perdita di ridondanza che ne comporta;
3. Healthy + Warning: questo caso meriterebbe maggior attenzione dal punto di vista teorico, tuttavia dando un peso maggiore al sensore che non fa parte della coppia fuori soglia, si può continuare a ritenere l’output buono con l’aggiunta di una segnalazione sul fatto che 2 dispositivi su 3 stanno avendo un comportamento stranamente diverso tra loro;
4. No Output: tutti e 3 i sensori presentano un comportamento anomalo. Non vi sono informazioni sufficienti per identificare quale sia la misura corretta.

Prima di concludere la trattazione sulla tecnica di diagnostica utilizzata, è doverosa una precisazione. Le casistiche affrontate dal nostro metodo sono tutte riconducibili ad una situazione in cui ogni sensore sta producendo una risposta per la misura. Il modello, perciò, non tiene conto del fatto che i dispositivi potrebbero non funzionare di per sé stessi nel momento in cui si trovano sconnessi dal loro *connector* al campo. Il risultato è che il sensore sconnesso non fornirà ovviamente alcun dato, ma mancherà proprio. Si aggiungono, allora, altri 3 sotto casi: Il primo si ha quando un solo sensore viene meno e, quindi, resta solo una correlazione tra 2. Se quest'ultima è all'interno dell'intervallo di accettazione dell'ipotesi nulla, si può prendere la media tra i 2 sensori come valore di pressione finale. Se invece, anche la correlazione rimasta va al di fuori del limite di controllo, bisognerà considerare inaffidabile il dato. Infine, l'ultimo caso che vale la pena considerare, è quello in cui mancano 2 sensori su 3. Possiamo decidere di restituire comunque il valore fornito dal terzo sensore, ma la misura sarà del tutto dubbia. Infatti, non avendo abbastanza elementi né per rifiutare né per accettare l'ipotesi nulla, ci troviamo in una situazione di incertezza totale.

## 2.5.2 - Modello diagnostico SVM

I modelli SVM fanno parte dei modelli supervisionati classificatori e regressivi machine learning. La teoria alla base di questi modelli è già stata approfondita all'interno del capitolo 1.2.5.3.1.

È stato scelto di utilizzare questa famiglia di modelli machine learning grazie principalmente a due caratteristiche:

1. versatilità di utilizzo: La qualità previsionale e classificatoria di questi modelli non dipende dalla tipologia di guasto analizzata ma solo dalla quantità e qualità dei dati di addestramento a disposizione.;
2. miglioramento continuo: Essendo la loro accuratezza strettamente connessa alla quantità e alla qualità dei dati a disposizione, incrementando la base dati a disposizione del loro addestramento, incrementerà conseguentemente anche il loro potere classificatorio.

Tra i vari modelli machine learning ci siamo focalizzati sulle Support Vector Machine (SVM) in quanto una volta addestrato fornisce due tipi di output:

1. Classificazione del comportamento attuale del componente;
2. Distanza del punto di funzionamento attuale del componente dalla superficie di separazione delle classi;

Quest'ultimo output avrà quindi valore maggiore quanto più il comportamento del componente è situato all'interno della regione della classe di appartenenza e quindi quanto più saremo certi che il macchinario sia effettivamente nella classe predetta.

Di conseguenza questo parametro può essere considerato come la variabile proxy dello stato di salute del componente, sul quale costruire un modello prognostico attraverso l'analisi del suo andamento nel tempo.

### 2.5.3 - Modello diagnostico - Riconoscitore di forme d'onda

Questo modello statistico è stato utilizzato all'interno del caso studio per riconoscere le forme d'onda tipiche della pressione in ingresso al compressore che precedono l'insorgenza del fenomeno del pompaggio. L'applicazione ed i risultati della validazione saranno approfonditi nel capitolo 3.7.2 -.

L'algoritmo deve essere in grado di individuare, dato un segnale uniformemente campionato, tutte le porzioni temporali del segnale "somiglianti" a delle forme campioni preventivamente inserite in input (chiamate d'ora in avanti *firme*). Tale somiglianza:

1. non deve dipendere (o deve dipendere il meno possibile) dalla lunghezza in campioni della porzione di segnale di confronto,
2. non deve dipendere (o deve dipendere il meno possibile) dall'ampiezza della porzione di segnale di confronto,
3. deve essere definita in maniera semplice e robusta.

I primi due punti sono soggetti a volute restrizioni e non valgono in assoluto. In particolare, dipendentemente dal fenomeno fisico, potrebbe essere necessario riconoscere come "somiglianti" una firma e una porzione di segnale solo se quest'ultima ha un'ampiezza totale (picco-picco), o una durata, contenuta entro certi margini, tipici del fenomeno fisico di interesse. L'algoritmo, quindi, consente di inserire questi margini come input, sia in termini di durata che di ampiezza.

Per quanto riguarda la definizione di "somiglianza", l'algoritmo definisce una porzione di segnale somigliante ad una firma se la distanza quadratica media delle due, opportunamente rinormalizzate, è inferiore ad un valore fissato in input dall'utente. Data quindi questa definizione di "somiglianza", l'algoritmo seleziona tutte le porzioni di segnale in input e le confronta con tutte le firme in input, individuando e segnalando le porzioni che risultano somiglianti ad una firma.

### 2.5.3.1 Definizione della distanza

Siano  $v_1 \in \mathbb{R}^n$  e  $v_2 \in \mathbb{R}^m$  due vettori reali, genericamente di differente lunghezza  $n$  e  $m$ , rappresentanti rispettivamente la firma e la porzione di segnale da confrontare. L'insieme  $L$  di tutti i possibili valori  $n/m$  è un insieme discreto finito, e fa parte degli input dell'algoritmo. La prima operazione effettuata dall'algoritmo è un ricampionamento tramite interpolazione (che per semplicità è stata scelta lineare) del vettore  $v_1$  per ottenere un vettore  $v_1' \in \mathbb{R}^m$  con uguale numero di elementi di  $v_2$ . Successivamente, vengono definiti due nuovi vettori  $w_1, w_2 \in \mathbb{R}^m$ :

$$w_{1i} = \frac{v_{1i}' - \min(v_1')}{\max(v_1') - \min(v_1')}$$

$$w_{2i} = \frac{v_{2i} - \min(v_2)}{\max(v_2) - \min(v_2)}$$

$$i = 1, \dots, m$$

Si ha dunque che  $0 \leq w_{1i} \leq 1$  e  $0 \leq w_{2i} \leq 1$ . Si può definire abbastanza facilmente la distanza quadratica  $\tilde{d}$  tra  $w_1$  e  $w_2$  come segue:

$$\tilde{d}(w_1, w_2) = \sqrt{\frac{1}{m} \sum_{i=1}^m (w_{1i} - w_{2i})^2}$$

Si definisce poi la distanza tra i vettori originali  $v_1$  e  $v_2$ :

$$d(v_1, v_2) = \tilde{d}(w_1, w_2)$$

dove i vettori  $w_1$  e  $w_2$  sono costruiti da  $v_1$  e da  $v_2$  come descritto in precedenza. Date le precedenti relazioni, si può dimostrare che  $0 \leq d(v_1, v_2) \leq 1$ . La distanza  $d$  così costruita risulta perciò assoluta (senza unità di misura), con valori compresi tra 0 e 1, ed assume il valore 0 solo se i vettori  $v_1$  e  $v_2$  sono uno il ricampionamento e la riscalatura dell'altro.

L'algoritmo è stato costruito in MATLAB e la sua implementazione completa è riportata in appendice B.



### 2.5.4 - Analisi di serie temporali ARIMA

Il metodo ARIMA (AutoRegressive Integrated Moving Average) è un modello che dati in ingresso 3 parametri p,d,q è in grado di modellare l'andamento di una serie temporale e di prevedere ciò che accadrà i successivi istanti temporali. I 3 parametri stanno ad indicare:

- p: ordine del modello autoregressivo;
- d: grado del primo differenziale;
- q: ordine del modello a media mobile.

Per il calcolo dei coefficienti si procede con l'analisi dell'autocorrelazione e della autocorrelazione parziale (ACF e PACF). Il numero di lag significativi della ACF identificherà il valore di p, quello della PACF (misura la correlazione della grandezza a tempi diversi al netto della correlazione con gli istanti intermedi) mi fornirà il valore di q, e la D mi indicherà il grado di non stazionarietà della funzione, zero quando è stazionaria, uno quando a trend lineare, ecc.

$$ACF = \rho_{lag} = \frac{E[(X_t - \mu(t))(X_{t+lag} - \mu(t))]}{\sqrt{E[(X_t - \mu(t))^2] * E[(X_{t+lag} - \mu(t))^2]}}$$

$$PACF = \rho_{lag\ lag} = \frac{|R_{lag}^*|}{|R_{lag}|}$$

$$con |R_{lag}^*| = \begin{bmatrix} 1 & \rho_1 & \dots & \rho_1 \\ \rho_1 & 1 & \dots & \rho_2 \\ \dots & \dots & \dots & \dots \\ \rho_{lag-1} & \rho_{lag-2} & \dots & \rho_{lag} \end{bmatrix}$$

$$e |R_{lag}| = \begin{bmatrix} 1 & \rho_1 & \dots & \rho_{lag-1} \\ \rho_1 & 1 & \dots & \rho_{lag-2} \\ \dots & \dots & \dots & \dots \\ \rho_{lag-1} & \rho_{lag-2} & \dots & 1 \end{bmatrix}$$

L'utilizzo di questa tecnica previsionale in campo ingegneristico o industriale non è innovativo ma esistono ancora pochi casi di studio dato che la loro applicazione è circoscritta a fenomeni ad andamento lento che usualmente sono di interesse inferiore. Ad esempio, quando parliamo di diagnostica prognostica e manutenzione di cuscinetti ci riferiamo sempre ad elementi alto-rotanti, con velocità dell'ordine delle centinaia o migliaia di rpm, per i quali le analisi migliori sono quelle riportate anche in questa tesi. Per cuscinetti invece con velocità di rotazione basse (<10rpm), come riportato nel lavoro dell'Università di Wollongong (Australia) (Kosasih et al., 2014), le energie in gioco sono talmente basse

che con i metodi classici non si sarebbe in grado di rilevare alcun guasto. Proprio in questi casi il modello ARIMA risulta alquanto efficace.

Una valida alternativa per la scelta dei coefficienti  $p$ ,  $D$  e  $q$  risulta essere la valutazione dei parametri AIC e BIC del modello statistico costruito.

Il criterio d'informazione di Akaike (Akaike's information criterion, AIC) è un metodo per la stima e il confronto tra diversi modelli statistici e fornisce una misura della qualità del modello analizzato tenendo conto sia della bontà di adattamento che della complessità del modello. Si basa sul concetto di entropia dell'informazione e ha come obiettivo quello di preferire i modelli con l'AIC più basso. La formula per il calcolo dell'AIC, che fornisce una visione della relazione tra la massima verosimiglianza del modello e la penalità per la complessità, è:

$$AIC = -2(\log L) + 2(NumParam)$$

Il Criterio di informazione Bayesiano (Bayesian information criterion, BIC) è ampiamente utilizzato per identificare il modello di serie storiche e regressione lineare più rappresentativo della realtà. Può, tuttavia, essere applicato a qualsiasi insieme di modelli basati sulla massima verosimiglianza. Per la statistica è un criterio per la selezione di un modello tra una classe di modelli parametrici con un diverso numero di parametri. Stimando i parametri del modello mediante il metodo della massima verosimiglianza, è possibile aumentarla attraverso l'aggiunta di parametri, causando tuttavia un incremento della complessità del modello. Il BIC risolve questo problema introducendo un termine di penalizzazione per il numero di parametri utilizzati. Come AIC, BIC utilizza il valore ottimale della funzione loglikelihood e penalizza i modelli più complessi, cioè modelli con parametri aggiuntivi. La penalizzazione del BIC è una funzione della dimensione del campione, ed è quindi tipicamente più severa di quella dell'AIC. La formula per il calcolo del BIC è:

$$BIC = -2(\log L) + NumParam * \log(NumObs)$$

I passi per la costruzione e l'utilizzo di un modello di tipo ARIMA sono i seguenti:

- step 1: trasformazione della time-series non stazionaria in una stazionaria con l'utilizzo di tecniche differenziali (il coefficiente  $D$  è inserito a questo scopo);
- step 2: scelta dei parametri  $p$  e  $q$ ;
- step 3: stima dei coefficienti attraverso una regressione ai minimi quadrati;
- step 4: forecast dei successivi istanti temporali;
- Step 5: rifinitura del modello attraverso successive iterazioni degli step 2-4.

### 2.5.4.1 Struttura dell'algoritmo prognostico per il suo funzionamento real-time

L'algoritmo predittivo ARIMA dell'evento di guasto per segnali lento varianti è stato suddiviso nelle seguenti fasi:

- Addestramento (ARIMA\_training): in questa fase attivabile solo manualmente dall'operatore, vengono definiti i criteri di attivazione dell'algoritmo di predizione ARIMA e tutti i parametri necessari all'attivazione dell'algoritmo; In questa fase si addestra il modello di state detection sulla base di una semplice carta di controllo;
- Inizializzazione (ARIMA\_init): in questa fase, che avviene online richiamata dal DCS di impianto, viene inizializzato l'algoritmo ARIMA mediante il calcolo dei suoi parametri necessari per la previsione (p,D,q).
- Realtime (ARIMA\_realtime): in questa fase, richiamata online direttamente dal DCS di impianto, la routine controlla che siano state effettuate le fasi di addestramento e di inizializzazione ed effettua la previsione, restituendo come output l'istante temporale in cui si ha la probabilità più alta di avere l'evento di guasto e i due istanti temporali, corrispondenti al limite superiore ed inferiore del prescelto intervallo di confidenza, di avvenimento del medesimo evento di guasto.

La fase di addestramento è l'unica che deve essere attivata manualmente; le altre, sono attivate automaticamente dal DCS di impianto. In particolare, la chiamata della corretta routine diagnostica avverrà leggendo il valore della variabile varMAT che dà anche indicazioni sul risultato dell'analisi. Questa variabile assume i seguenti valori:

- varMAT=1 → diagnosi possibile, output valido → Il DCS richiama ARIMA\_realtime;
- varMAT=2 → diagnosi possibile, previsione inaffidabile → Il DCS richiama ARIMA\_realtime;
- varMAT=3 → Necessaria inizializzazione (pochi valori storici) → Il DCS richiama ARIMA\_init;
- varMAT=4 → Necessario training manuale offline → Il DCS crea un allarme per gli operatori

Il valore iniziale della variabile varMAT è 1. La routine ARIMA\_realtime è quella che controlla la necessità di riaddestramento o inizializzazione del modello.

La logica di costruzione dell'algoritmo prevede che questo venga chiamato solo nel caso in cui la variabile monitorata presenti un valore anomalo secondo i limiti identificati in fase di addestramento, verificati da una funzione implementata direttamente sul DCS di impianto.

Tutte le routine utilizzano una specifica cartella con nominativo univoco per ogni variabile monitorata, all'interno della quale salvare tutti i parametri e lo storico dei dati anomali necessari al corretto funzionamento delle routine.

### 2.5.4.1.1 ARIMA training

La logica dell'algoritmo è rappresentata nel seguente diagramma a blocchi:

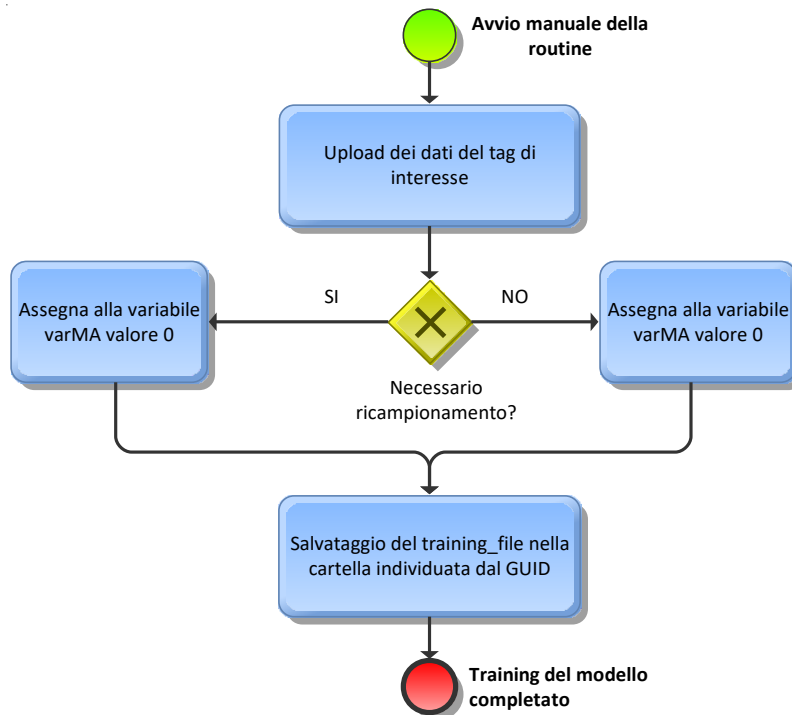


Figura 2.4 Logica dell'algoritmo ARIMA training.

L'algoritmo viene avviato manualmente da un operatore che dovrà indicare i seguenti input:

- GUID= identificativo univoco della directory di salvataggio della variabile monitorata
- tag = nome della variabile da monitorare
- path= directory contenente lo storico del tag da monitorare
- datego1= estremo sinistro dell'intervallo di buon funzionamento
- datego2= estremo destro dell'intervallo di buon funzionamento
- threshold= valore di soglia di allarme
- ndata= numero di dati necessari all'inizializzazione
- estpoint= numero di punti di previsione dell'algoritmo ARIMA
- fail\_duration= intervallo temporale dall'inizio dell'anomalia nella misura monitorata fino al valore di soglia (evento di guasto)

- `sampling_time`= tempo di campionamento dei dati acquisiti

L'algoritmo provvede quindi all'upload dei dati storici nel periodo fra le date di buon funzionamento.

Successivamente l'algoritmo calcola il valore di media e deviazione standard nel periodo di buon funzionamento individuati ed indicati dall'operatore come input attraverso i parametri " `datego1`" e " `datego2`". In questa fase viene quindi costruita la carta di controllo per l'attivazione della previsione della RUL.

Se al posto della semplice carta di controllo monovariata si decida di utilizzare un qualsiasi altro modello diagnostico e/o di state detection, queste prime due fasi dell'algoritmo saranno svolte da quest'ultimo modello lasciando invariate le successive.

La successiva analisi riguarda la valutazione di un eventuale ricampionamento dei dati e l'individuazione del relativo intervallo temporale tra dati successivi. La necessità di ricampionare i dati, che potrebbero essere acquisiti con un tempo di campionamento troppo ristretto per il tipo di analisi, è dovuta principalmente a tre fattori:

1. Limite minimo di campioni per la costruzione di un modello ARIMA consistente;
2. Limite sul rapporto tra numero di punti di previsione e numero di punti utilizzati per la costruzione di un modello ARIMA consistente;
3. Limite superiore sul numero di punti da predire funzione delle caratteristiche di memoria fisica del pc utilizzato per l'analisi.

In particolare, per ottenere un modello ARIMA consistente sono necessari almeno 10 punti per l'addestramento (se ne consigliano almeno 100), ed il rapporto tra numero di punti predetti e numero di punti per l'addestramento del modello dovrà essere contenuto al fine di ottenere una buona confidenza sulla previsione dei risultati (si consiglia massimo 10). Infine, un numero eccessivo di punti di previsione, oltre ad avere un costo computazionale molto elevato, trova un limite nelle caratteristiche di memoria fisica del computer utilizzato (es: RAM pari a 16267 MB, numero massimo di punti di previsione pari a circa 30 000).

Alla luce di quanto detto, per ottenere una previsione consistente e che raggiunga con un buon anticipo l'evento di guasto, si è introdotto il controllo sul corretto tempo di campionamento in funzione dei seguenti parametri di input:

- `ndata`= numero di dati necessari all'inizializzazione

- *estpoint*= numero di punti di previsione dell'aritmo ARIMA
- *fail\_duration*= intervallo temporale dall'inizio dell'anomalia nella misura monitorata fino al valore di soglia (evento di guasto)
- *sampling\_time*= tempo di campionamento dei dati acquisiti

L'aritmo calcola il *sampling\_time* secondo la seguente formula:

$$ma\_time = \frac{fail\_duration}{(ndata + estpoint)}$$

E lo confronta con il *sampling\_time* dei dati di acquisizione. Se risulta maggiore o uguale a 2 volte il *sampling\_time*, l'aritmo attiva il ricampionamento dei dati acquisiti, che avviene nell'ARIMA\_training, ponendo la variabile *varMA* pari ad 1 (pari a 0 se non è necessario il ricampionamento). Il ricampionamento avviene facendo la media dei dati all'interno della finestra di ricampionamento, pertanto l'aritmo calcola anche il numero di dati acquisiti all'interno di tale finestra secondo la seguente relazione:

$$MAlag = round\left(\frac{ma\_time}{sampling\_time}\right)$$

L'aritmo restituisce ad output e comunica al DCS di impianto i seguenti parametri:

- *meango*= valore medio dei valori acquisiti nella finestra di buon funzionamento
- *stdevgo*=deviazione standard dei valori acquisiti nella finestra di buon funzionamento

necessari al funzionamento della carta di controllo costruita sull' AF analytics, per l'individuazione di valori anomali della variabile monitorata.

Gli ulteriori parametri necessari al funzionamento delle altre fasi di analisi (ARIMA\_init e ARIMA\_realtime) sono invece salvati su un file .m chiamato "training\_file" nella directory identificata dal parametro GUID di input. Tali parametri sono:

- *p\_ndata*= numero di dati necessari all'inizializzazione impostati al momento del training
- *p\_estpoint*= numero di punti di previsione dell'aritmo ARIMA impostati al momento del training
- *p\_threshold*= valore di soglia di allarme impostato al momento del training
- *sampling\_time*= tempo di campionamento o ricampionamento dei dati

- varMA= booleano di attivazione del ricampionamento (se pari a 1 ricampionamento necessario, se zero non necessario)
- MAlag= numero di dati nell'intervallo di ricampionamento (se varMA=0 allora MAlag=0)

### 2.5.4.1.2 ARIMA init

La logica dell'algorithmo è rappresentata nel seguente diagramma a blocchi:

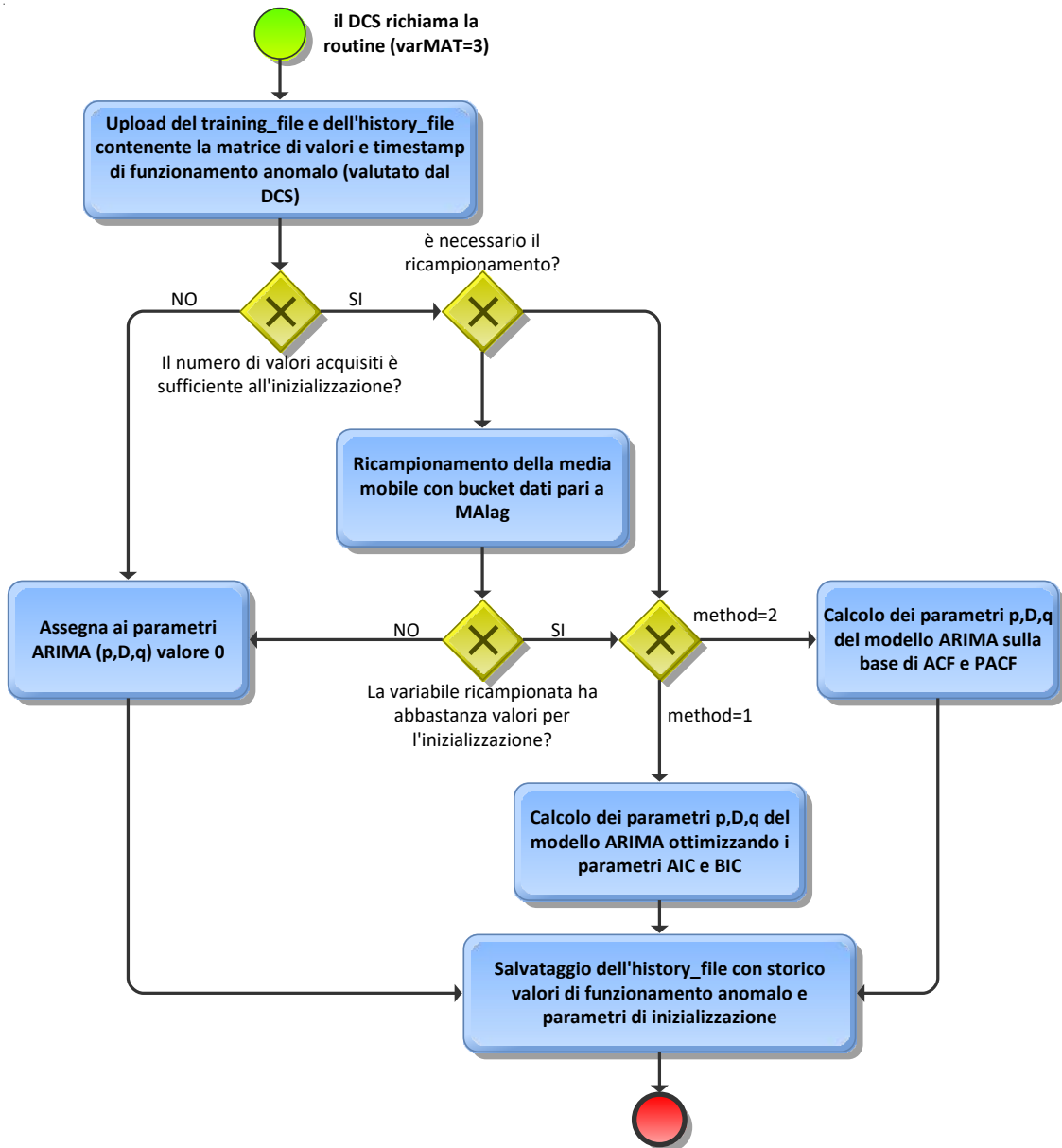


Figura 2.5 Logica dell'algorithmo ARIMA init.

L'algorithmo viene richiamato dal DCS di impianto, quando la variabile varMAT assume valore pari a 3, con i seguenti parametri di input:

- GUID = identificativo univoco della directory di salvataggio della variabile monitorata;
- method = booleano (1 o 2) per la selezione del metodo di calcolo dei parametri p e q del modello.  
1 per calcolo tramite ACF e PACF, 2 per ottimizzazione AIC-BIC;
- limit = numero intero pari al limite superiore per i parametri p e q.

La routine procede effettuando l'upload del training\_file generato dall' ARIMA\_trainig e dell'history\_file generato dall'ARIMA\_realtime. Quest'ultimo contiene i dati storici anomali con il relativo timestamp. Viene quindi verificato che la quantità di dati salvati all'interno dell'history\_file sia sufficiente alla costruzione dell'algorithm ARIMA. La definizione del numero minimo di dati per l'addestramento del modello è un parametro che viene impostato dall'operatore in fase di training del modello (parametro "ndata" del "training\_file"). Nel caso in cui i dati salvati nell'hystory\_file siano insufficienti alla costruzione del modello ARIMA, la routine assegna ai parametri di inizializzazione p, D e q valore pari a 0, valori non plausibili per la costruzione del modello. Al contrario, se il numero di dati è maggiore o uguale a quanto richiesto in fase di training, l'algorithm, in funzione del valore della variabile "varMA" (se pari a 0 ricampionamento non necessario, se pari ad 1 ricampionamento attivo) procede o meno al ricampionamento dei dati facendone la media nelle finestre di numerosità pari a "MAlag", variabile definita in fase di training del modello. Se ricampionati, i dati risulteranno di numerosità inferiore e quindi è possibile che non raggiungano il valore minimo per la costruzione del modello ARIMA. Di conseguenza se il valore è minore od uguale al parametro "ndata" assegnerà valore zero ai parametri di inizializzazione (p,D,q), altrimenti procederà con il calcolo di questi parametri. Se la variabile di input method è pari ad 1, i parametri p e q saranno calcolati in funzione dell'autocorrelazione e dell'autocorrelazione parziale dei dati di input, se method è pari a 2 utilizza l'ottimizzazione dei parametri AIC e BIC come descritto nel paragrafo 2.5.4 -.

La routine infine, salva sul file .m chiamato "history\_file" nella directory identificata dal parametro GUID di input, tutti i parametri e dati necessari alla routine ARIMA\_realtime. In particolare, salva:

- ndata= numero di dati necessari all'inizializzazione impostati al momento del training;
- val= matrice di dati e timestamp anomali inviati dall'AF;
- val\_ma= matrice di dati e timestamp anomali inviati dall'AF ricampionati. Se il ricampionamento non è necessario è uguale a val;
- p= ordine del modello autoregressivo;
- D= grado del primo differenziale;
- Q= ordine del modello a media mobile.;



### 2.5.4.1.3 ARIMA realtime

La logica dell'algoritmo è rappresentata nel seguente diagramma a blocchi:

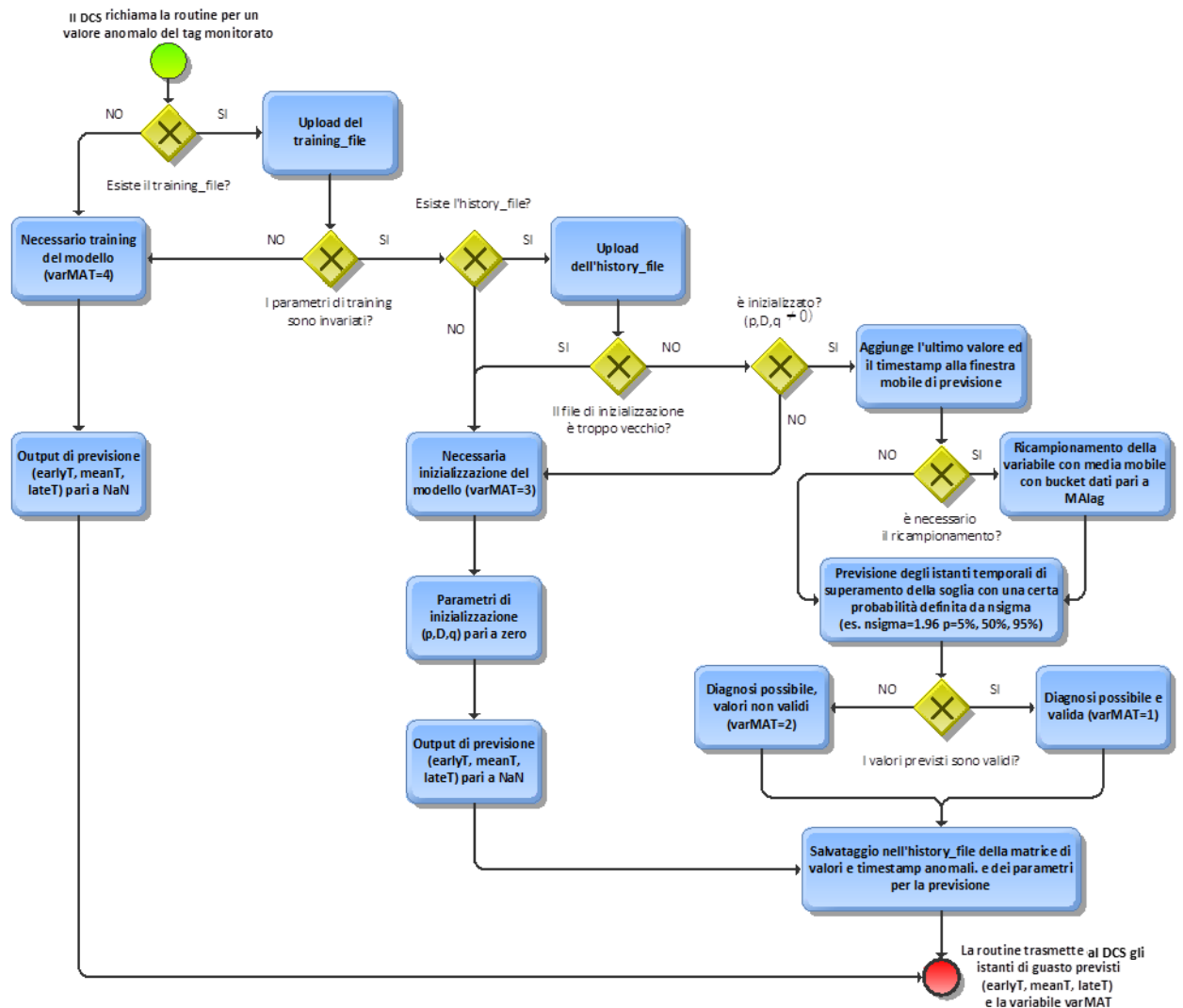


Figura 2.6 Logica dell'algoritmo ARIMA real-time.

L'algoritmo viene richiamato dal DCS di impianto, ogni volta che la variabile monitorata assume un valore anomalo, valutato tramite carta di controllo implementata sul DCS stesso. I parametri di input sono i seguenti:

- GUID= identificativo univoco della directory di salvataggio della variabile monitorata
- newval = valore anomalo attuale della variabile da diagnosticare
- estpoint= numero di punti di previsione dell'algoritmo ARIMA
- threshold= valore di soglia di allarme
- ndata= numero di dati necessari all'inizializzazione

- nsigma= numero di deviazioni standard intorno al valore medio che determinano i punti estremi di previsione

Gli output della routine, trasmessi al DCS, sono i seguenti:

- earlyT= istante temporale di superamento del livello di soglia corrispondente al percentile inferiore individuato da nsigma;
- meanT= istante temporale atteso di superamento del valore di soglia;
- lateT= istante temporale di superamento del livello di soglia corrispondente al percentile superiore individuato da nsigma;
- varMAT= variabile di attivazione della corretta routine.

La routine effettua per prima cosa l'upload del training\_file, se esistente, e valuta se i parametri: estpoint, threshold e ndata sono gli stessi utilizzati per l'addestramento del modello ARIMA. Nel caso in cui il modello non sia stato addestrato (il training\_file non esiste) o siano cambiati i parametri di input assegna valore pari a NaN (not a number) ai tre istanti di previsione e valore pari a 4 alla variabile varMAT (necessario training del modello).

Nel caso in cui il modello è addestrato (training\_file presente), la routine controlla se è presente il file di inizializzazione (history\_file), se questo è sufficientemente nuovo e se è inizializzato (parametri di inizializzazione p,D e q diversi da zero). In particolare, se non inizializzato (mancanza del file o di parametri di inizializzazione), o se la differenza tra il timestamp dell'ultimo dato salvato nell'history\_file e il nuovo dato inviato dall'AF è maggiore di 10 volte il sampling\_time, la routine procede ad assegnare valore pari a 3 alla variabile varMAT (necessaria inizializzazione), NaN (not a number) ai tre istanti di previsione e zero ai parametri di inizializzazione (p,D,q). Procede inoltre al salvataggio del nuovo dato acquisito nell'history\_file.

Nel caso in cui il modello ARIMA risulti inizializzato, la routine aggiunge l'ultimo valore inviato dall'AF alla finestra mobile di previsione (ed elimina quindi il primo della finestra); procede poi se necessario (varMA=1) al ricampionamento dei dati della finestra mobile ed effettua la previsione dei 3 istanti temporali di superamento della soglia (earlyT, meanT e lateT) con probabilità individuata dal parametro di input nsigma. Ad esempio:

**Tabella 2.3 esempio degli intervalli di confidenza in funzione del parametro nsigma**

nsigma	percentile earlyT	percentile meanT	percentile lateT	Intervallo di confidenza
--------	-------------------	------------------	------------------	--------------------------

1	15.87%	50%	84.13%	68.26%
2	2.27%	50%	97.73%	95.46%
3	0.13%	50%	99.87%	99.74%

Infine, se i valori predetti sono tutti diversi da NaN, la routine assegna a varMAT valore 1 (diagnosi possibile e valida), altrimenti valore pari a 2 (diagnosi possibile, valori non validi). Quest'ultimo caso si verifica quando l'ultimo punto predetto dal modello ARIMA non supera la soglia di allarme impostata (threshold); seppur il modello è addestrato e inizializzato correttamente non siamo ancora in grado di prevedere il superamento della soglia.

La routine infine, salva sul file .m chiamato "history\_file" nella directory identificata dal parametro GUID di input, il nuovo vettore di dati della finestra mobile aggiornato, oltre ai parametri necessari al modello ARIMA. In particolare, salva:

- ndata= numero di dati necessari all'inizializzazione impostati al momento del training;
- val= matrice di dati e timestamp anomali inviati dall'AF;
- val\_ma= matrice di dati e timestamp anomali inviati dall'AF ricampionati. Se il ricampionamento non è necessario è uguale a val;
- p= ordine del modello autoregressivo;
- D= grado del primo differenziale;
- Q= ordine del modello a media mobile.

In appendice sono riportati gli algoritmi descritti in questo capitolo sviluppati in MATLAB.

### 2.5.5 - Modello di state detection – Carte di controllo multivariate

Le carte di controllo multivariate furono introdotte nel 1947 da Harold Hotelling, consentono di aggregare informazioni relative ad alcune variabili di processo/prodotto su un diagramma utilizzando la cosiddetta statistica  $T^2$ .

La carta di controllo  $T^2$  è basata sulla distribuzione normale multivariata e sulla distanza di Mahalanobis, ovvero la distanza del set di variabili acquisite dalla media della distribuzione gaussiana multivariata. I valori del vettore medio e della matrice di covarianza della popolazione sono stimati a partire dal set di dati storici delle variabili di controllo selezionate che è il campione della popolazione della quale vogliamo

stimare la funzione di distribuzione di probabilità multivariata. La statistica  $T^2$  di Hotelling è definita come segue;

siano:

$$\mathbf{x}_1, \dots, \mathbf{x}_n$$

vettori colonna di numeri reali dove ogni colonna rappresenta lo storico di una variabile di controllo selezionata e:

$$\bar{x} = (x_1 + \dots + x_n)/n$$

Le loro medie. Sia:  $\bar{x}$

$$W = \sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x})' / (n - 1)$$

La matrice delle loro varianze e sia  $\mu$  il vettore colonna delle medie stimate della popolazione, si ha:

$$T^2 = (\bar{x} - \mu)' W^{-1} (\bar{x} - \mu)$$

(Jolliffe, 2003; Krzanowski, 2000; Seber, 2009)

La statistica  $T^2$  risulta essere la singola variabile di controllo proxy delle performance del componente analizzato da utilizzare per diagnosticare significative variazioni delle prestazioni e da analizzare tramite il modello di analisi di serie temporali per prevedere quando le prestazioni scenderanno sotto la soglia di accettabilità.

Essendo  $T^2$  la distanza del set di variabili dalla media della gaussiana multivariata costruita sui dati storici di buon funzionamento, ad un incremento del parametro  $T^2$  corrisponde un abbassamento delle prestazioni del componente stesso.

### 2.5.6 - Modello di state detection – One class SVM

Lo one-class learning o SVM non supervisionata, mira a separare i dati dall'origine nello spazio  $n$  dimensionale, con  $n$  pari al numero di variabili di controllo. È un algoritmo utilizzato per il rilevamento di valori anomali. L'algoritmo viene addestrato sul set di dati storici delle variabili di controllo selezionate relative al buon funzionamento del componente analizzato con l'obiettivo di minimizzare la doppia espressione:

$$0,5 \sum_{jk} \alpha_j \alpha_k G(x_j, x_k)$$

Rispetto ad  $\alpha_1, \dots, \alpha_n$  soggetto a:

$$\sum \alpha_j = \nu$$

Con  $0 \leq \alpha_j \leq 1$  per ogni  $j = 1, \dots, n$ . Il valore di  $G(x_j, x_k)$  è un elemento  $(j, k)$  della matrice di Gram.

Un piccolo valore di  $\nu$  porta a un minor numero di vettori di supporto e, quindi, a un confine decisionale liscio e grezzo. Un grande valore di  $\nu$  porta a un maggior numero di vettori di supporto e, quindi, a un contorno decisionale sinuoso e flessibile. Il valore ottimale di  $\nu$  dovrebbe essere abbastanza grande da catturare la complessità dei dati e abbastanza piccolo da evitare il sovrallenamento. Inoltre,  $0 < \nu \leq 1$ . (Schölkopf et al., 2001)

Una volta addestrato sui dati storici delle variabili di controllo, il modello attribuisce ad ogni set di nuovi dati il valore di verosimiglianza di appartenenza alla popolazione di addestramento. Il valore di verosimiglianza è quindi la singola variabile di controllo proxy delle performance del componente controllato da utilizzare per diagnosticare significative variazioni delle prestazioni e da analizzare tramite il modello di analisi di serie temporali per prevedere quando le prestazioni scenderanno sotto la soglia di accettabilità.

Il valore di verosimiglianza sarà tanto più elevato quanto più il nuovo set di dati elaborato dal modello addestrato è vicino allo storico dati utilizzati per l'addestramento. Siccome l'addestramento è fatto sui dati relativi alle condizioni di buon funzionamento del macchinario, maggiore è il valore di verosimiglianza e migliori saranno le performance del componente.



## 3 - Caso studio

L'applicazione della metodologia sviluppata ad un caso studio ha permesso di affinarla, di testarla e validarla utilizzando le tecniche teorizzate.

Il caso studio è un impianto di produzione di energia geotermica di ENEL Greenpower S.p.A. con la quale ho collaborato per 2 anni durante il periodo di dottorato. Il caso studio si sviluppa nell'ambito del progetto strategico di ricerca e sviluppo POR FESR 2014-2020 cofinanziato dalla Regione Toscana di iniziativa di ENEL GREEN POWER S.P.A. dal titolo: "Sviluppo, prototipazione e dimostrazione di un sistema avanzato di tele-diagnostica per impianti geotermici (SmartGeo)". In tale progetto prendono parte, oltre al dipartimento di ingegneria industriale dell'Università degli studi di Firenze, altri enti tra i quali:

- Scuola Superiore di Studi Universitari e di Perfezionamento Sant'Anna;
- I.S.E – Società a Responsabilità Limitata;
- Università degli Studi di Firenze;
- S.D.I – Automazione Industriale – S.P.A.

L'obiettivo di questo progetto è quello di sviluppare un nuovo sistema avanzato di diagnostica e prognostica dei sistemi di un impianto geotermico al fine di estendere il monitoraggio a tutti i componenti in questione per riconoscere in anticipo, per mezzo di misurazioni on-line dello stato di funzionamento delle macchine, l'imminente incorrere dei guasti, così da supportare i responsabili di esercizio nell'ottimizzazione degli interventi di manutenzione.

I modelli sviluppati all'interno del progetto sono stati suddivisi in funzione del tipo di componente e del guasto sotto analisi in:

1. Modelli hard-realtime: sono modelli che diagnosticano modi di guasto e anomalie di funzionamento con una evoluzione temporale molto rapida e che richiedono l'intervento di automazione di impianto per essere scongiurati;
2. Modelli soft-realtime: sono modelli che prognosticano modi di guasto ed anomalie di funzionamento con una evoluzione temporale tipica dell'usura meccanica. La previsione di tali guasti necessita l'intervento di operatori di manutenzione per la sostituzione delle componenti usurate.

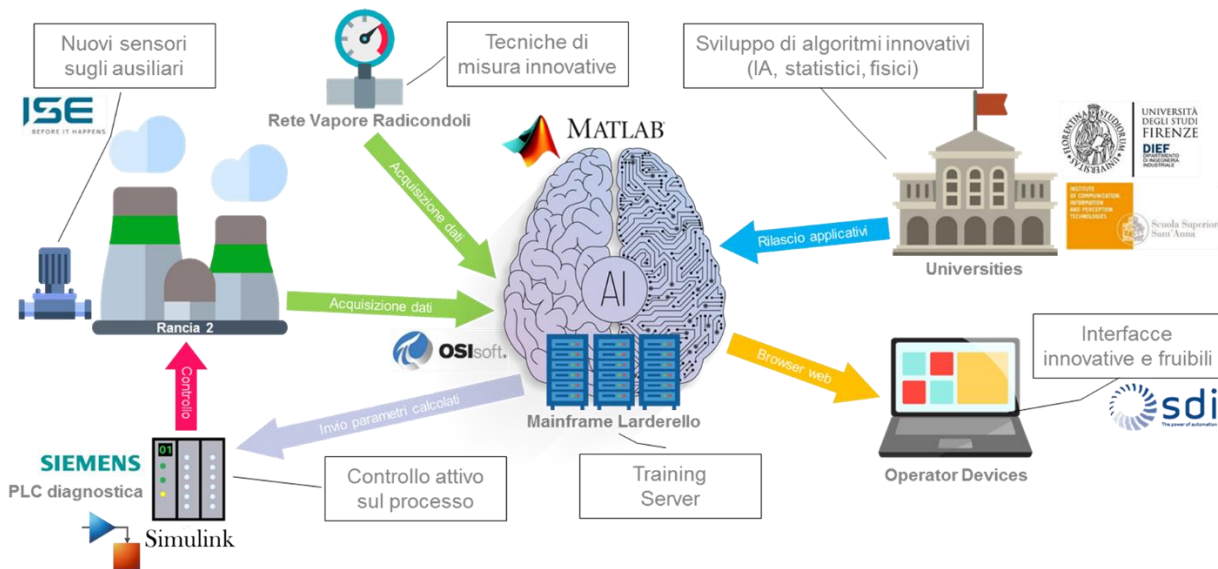


Figura 3.1 Struttura del progetto SmartGEO.

Come mostrato in figura, i partner di progetto hanno avuto differenti ruoli all'interno del progetto; le due università coinvolte si sono occupate dello sviluppo in MATLAB di algoritmi innovativi per la diagnostica e la prognostica di modi di guasto dei componenti di impianto; il partner SDI Automation si è invece occupato dello sviluppo dell'interfaccia di visualizzazione dei risultati degli algoritmi sviluppati ed il partner ISE della scelta ed installazione dei nuovi sensori necessari alla diagnostica e prognostica.

Dal punto di vista dell'architettura informatica implementata, nel caso di modelli hard-realtime, tutti i modelli diagnostici e prognostici sviluppati dalle università, vengono rilasciati nel mainframe di impianto, dove si ha la potenza di calcolo necessaria ad effettuare un addestramento dei modelli ML anche molto complessi in tempi brevi. Il risultato dell'addestramento, ovvero i parametri calcolati del modello appena addestrato vengono poi inviati al PLC diagnostico, che ha la funzione di monitorare i parametri del macchinario in real-time utilizzando il modello ML addestrato. Per il progetto è stato affiancato ai PLC di impianto anche questo PLC diagnostico, che rispetto agli altri ha la possibilità di leggere il linguaggio MATLAB Simulink permettendo quindi di effettuare analisi diagnostiche avanzate sui dati raw provenienti dai sensori ed intervenire rapidamente in caso di anomalie. Questa architettura è utilizzata per tutti i modelli diagnostici hard-realtime. Gli output del PLC diagnostico sono poi inviati al mainframe per la loro visualizzazione da parte degli operatori di impianto.

Per quanto riguarda i modelli diagnostici soft-realtime, anch'essi saranno rilasciati sul mainframe di impianto ma le analisi diagnostiche e prognostiche sono anch'esse elaborate sui mainframe.

L'interfaccia è stata realizzata su pagina web per poter essere visualizzata da tutti gli operatori di impianto e da tutti i tipi di dispositivo (PC, table, smartphone ecc.).



La mia partecipazione a questo progetto è stata l'occasione giusta per testare e validare quanto analizzato in letteratura.

### 3.1 - Gli impianti geotermici

L'energia geotermica è una tra le fonti di energia alternativa più comuni nel territorio italiano, in particolar modo in quello toscano. Se si prendono i numeri forniti dall'UGI (Unione Geotermica Italiana) al 2015, circa il 50% dei 11.500 GWh di energia geotermica generata in Europa è stato prodotto in Italia: questo valore supera perfino la quota prodotta in Islanda, paese noto nel mondo per essere in larga parte a territorio vulcanico. La maggioranza delle centrali geotermiche sul suolo italiano è concentrato nell'area toscana, nelle zone di Larderello, Radicondoli e del Monte Amiata, che da soli producono quasi il 10% di tutta l'energia geotermica generata nel mondo.

Tale ricchezza del sottosuolo fa sì che l'importanza delle centrali geotermiche per il sostentamento energetico "green" del nostro paese sia rilevante, e per questo è necessario che esse funzionino efficientemente la maggior parte del tempo possibile.

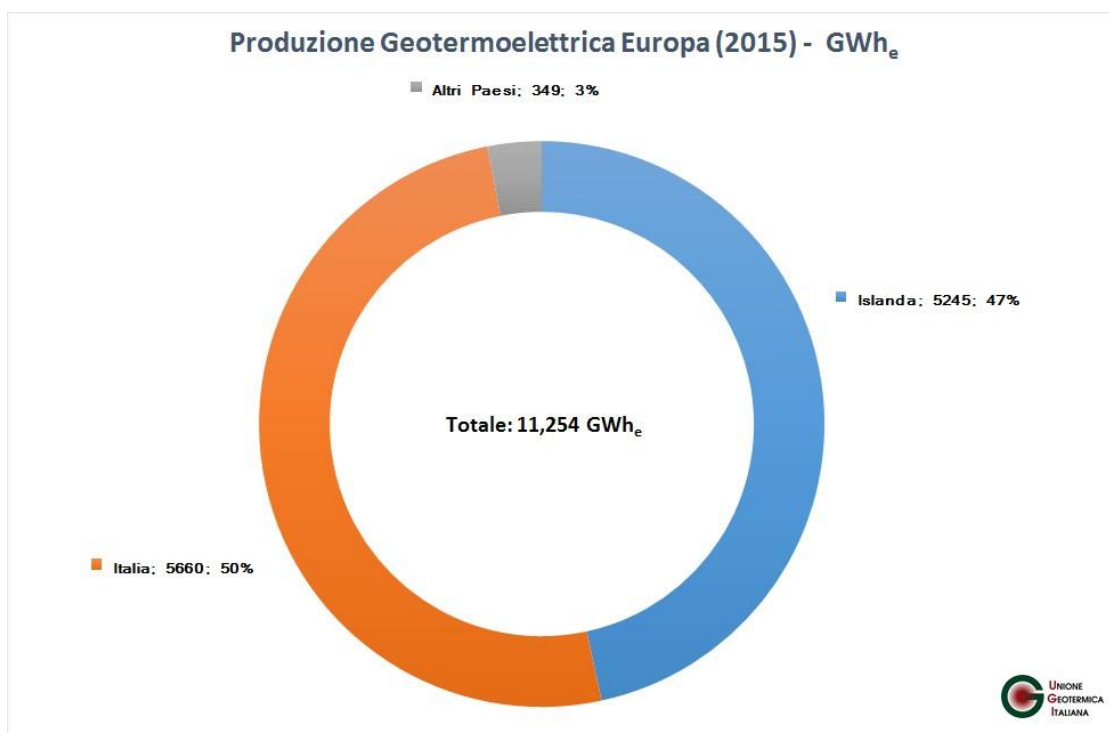


Figura 3.2 percentuale di produzione di energia elettrica da fonti geotermiche in Europa nel 2015.

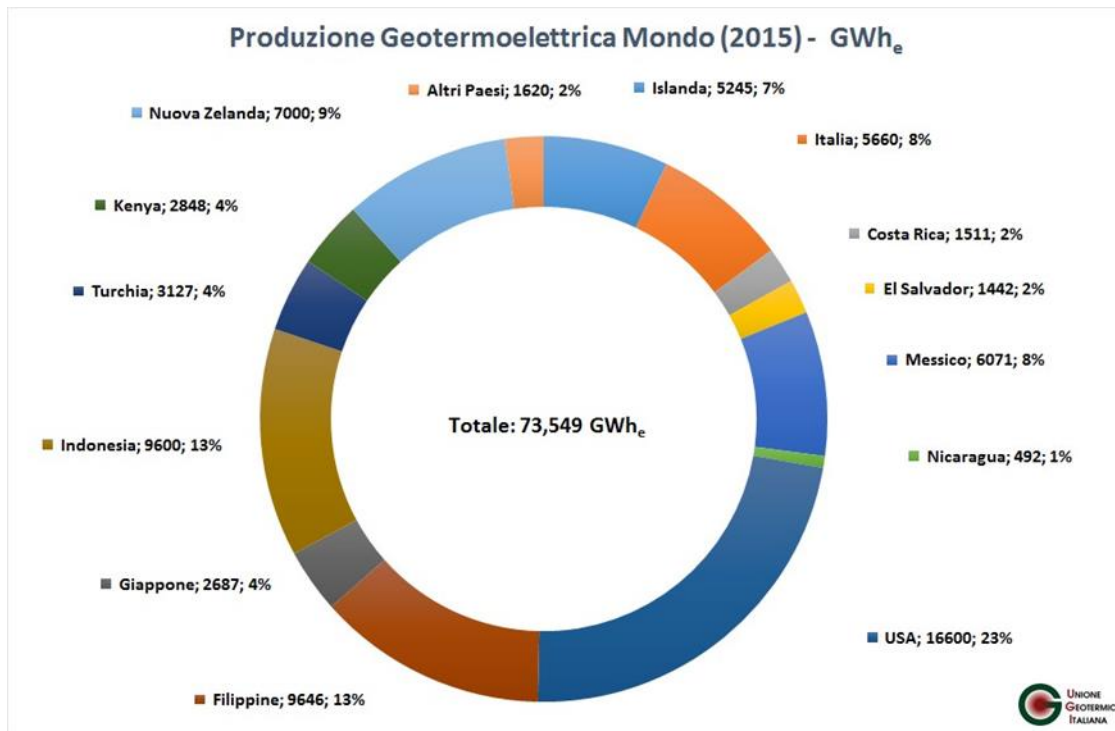


Figura 3.3 percentuale di produzione di energia elettrica da fonti geotermiche nel mondo nel 2015.

### 3.1.1 - Classificazione dei sistemi geotermici

L'energia geotermica è una forma di energia che deriva dal calore presente negli strati più profondi della crosta terrestre. Infatti, penetrando in profondità nella superficie terrestre, la temperatura diventa gradualmente più elevata, aumentando mediamente di circa 30 °C per km. I giacimenti di questa energia sono però dispersi e a profondità così elevate da impedirne lo sfruttamento. Per estrarre e usare il calore imprigionato nella Terra, è necessario individuare le zone con anomalia termica positiva dove il calore terrestre è concentrato: queste zone sono dette serbatoi o giacimenti geotermici.

I giacimenti geotermici possono essere di 4 tipi:

- Idrotermali,
- Petrotermici,
- Geopressurizzati,
- Magmatici.

Attualmente le uniche sorgenti sfruttate a livello industriale sono quelle idrotermali, costituite da formazioni rocciose permeabili (serbatoi geotermici), caratterizzate da elevata temperatura, e all'interno

delle quali circola acqua. In base ai valori di pressione e temperatura presenti all'interno del serbatoio, l'acqua può essere in fase liquida o in fase di vapore (Dickson and Fanelli, 2004).

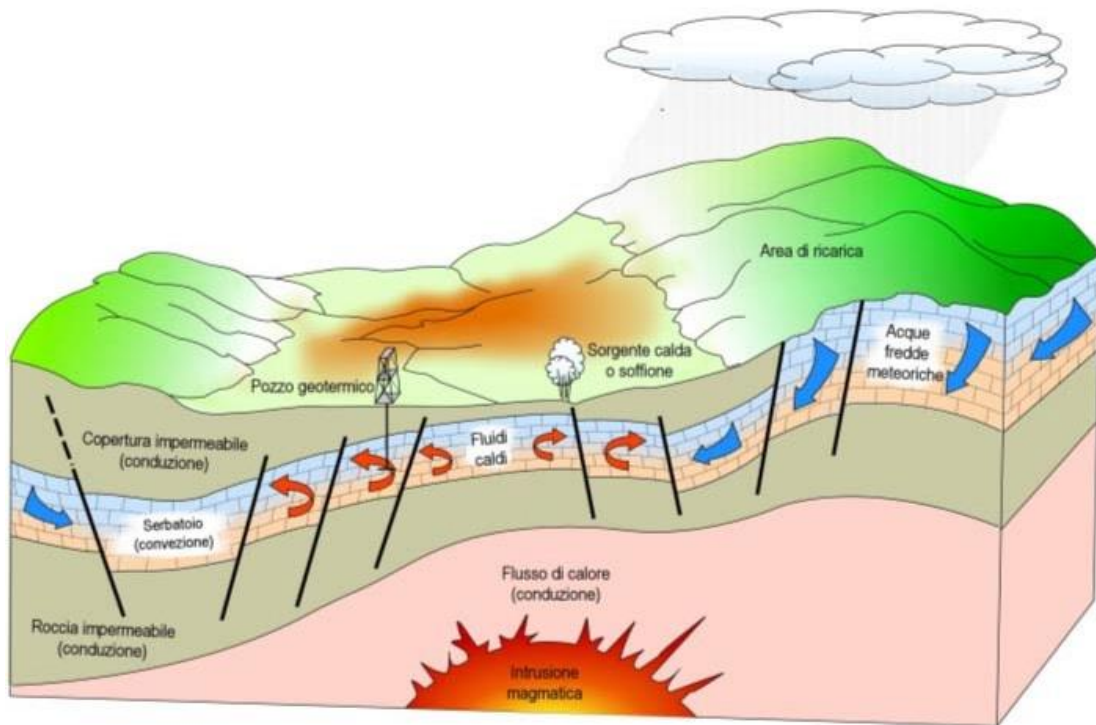


Figura 3.4 sorgente idrotermale.

### 3.1.2 - - Tecnologie di generazione geotermoelettriche

L'energia geotermica può essere utilizzata per la produzione di calore, ma il suo impiego prevalente riguarda la produzione di energia elettrica: il flusso di vapore proveniente dal sottosuolo produce una forza tale da far muovere una turbina, la cui energia meccanica viene trasformata in elettricità tramite un alternatore. La seguente figura illustra i possibili cicli di generazione di energia elettrica per i sistemi idrotermali:

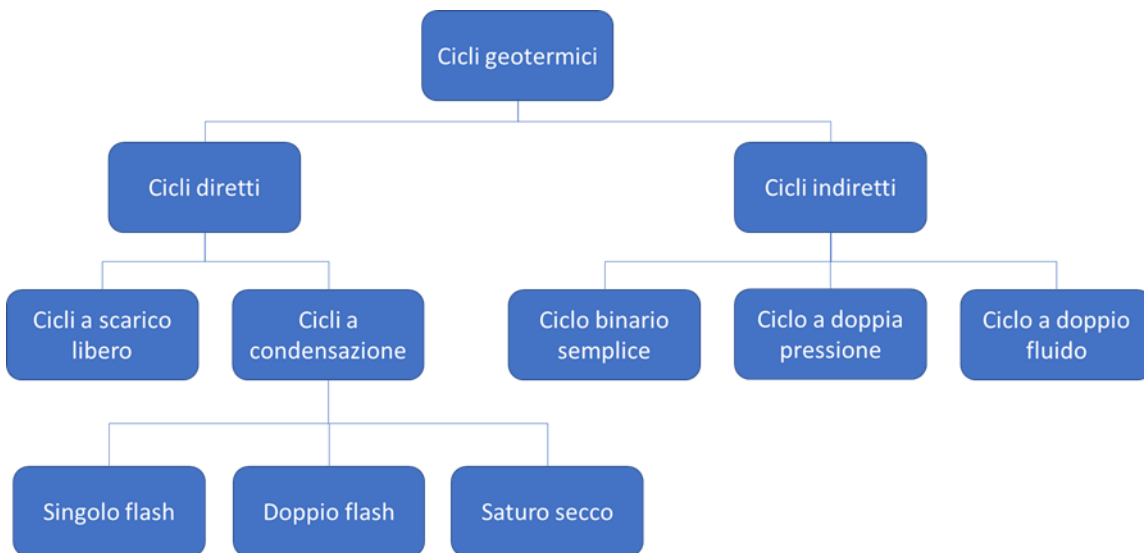


Figura 3.5 cicli di generazione di energia elettrica per i sistemi idrotermali.

### 3.1.3 - - Ciclo a condensazione a singolo flash

Il ciclo a condensazione a singolo flash è uno dei più efficienti meccanismi di generazione di energia elettrica da sorgente geotermica, e dunque uno dei più comuni. La figura seguente ne schematizza il funzionamento:

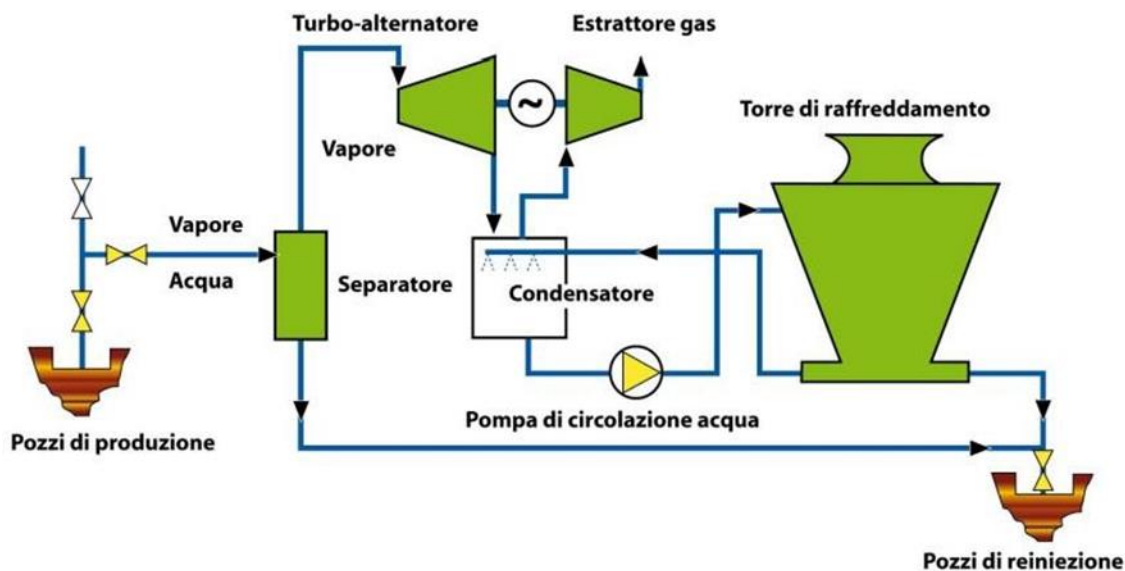


Figura 3.6 tipico schema di funzionamento del ciclo a condensazione a singolo flash i.

Il ciclo parte dall'individuazione di un serbatoio geotermico, al quale è possibile accedere tramite dei "pozzi di produzione", ottenuti attraverso opportune perforazioni, da cui estrarre il vapore.

Il vapore in questione, al momento dell'estrazione, risulta essere in condizioni non consone all'utilizzo in impianti di produzione, in quanto, date le caratteristiche del sottosuolo, ha una forte componente acida che, se condensata, corroderebbe tutta la componentistica di impianto con la quale dovesse entrare in contatto, rendendo necessarie continue e costose azioni manutentive. Per questo motivo, i pozzi di produzione sono dotati, o con della componentistica di superficie o con soluzioni che intervengono direttamente nel sottosuolo, di sistemi progettati per "lavare" questo vapore, così da ridurre la componente acida e riportarlo a condizioni di pH non aggressive per l'impianto stesso. In pratica, il lavaggio consiste nell'apporto di una soluzione a base di acqua e soda caustica che reagisce con il cloro presente nel vapore; ciò porta alla generazione di sali che si disciolgono nell'acqua, che naturalmente condensa nel processo di lavaggio o nei vapordotti.

Una volta estratto e "lavato", una fitta rete di chilometri di vapordotti porta il vapore ai vari siti produttivi. Tali vapordotti sono provvisti di scarichi per permettere alla condensa di essere rimossa, dato che durante il tragitto si può avere una parziale condensazione del vapore geotermico. Tale condensa viene poi raccolta nei bifasedotti, che sono delle tubazioni più piccole rispetto a quelle per il solo vapore, che si dispiegano parallelamente a quest'ultime in tutto il territorio, con il compito di trasportare l'acqua ai pozzi di reiniezione.

Raggiunto l'impianto, il primo elemento che si trova è un separatore ciclonico, che ha il compito innanzitutto di completare la separazione della componente liquida da quella aeriforme, formatasi per condensazione nel vapordotto. Inoltre, se necessario, completa la fase di lavaggio del vapore mediante degli appositi spruzzatori di soluzione acqua-soda. Il separatore è importante, in quanto alcuni pozzi di produzione estraggono vapore surriscaldato che non ha necessità di essere lavato direttamente in loco, dato che per l'alta temperatura non c'è rischio di ottenere la condensa, corrosiva per le tubazioni. Per questo, si prevede il suddetto componente anche per lavare il vapore, prima che entri in impianto e danneggi le apparecchiature.



**Figura 3.7 separatore ciclonico.**

Ora che il fluido è costituito dalla sola componente gassosa e con un pH accettabile, può evolvere in turbina, dopo una serie di valvole di regolazione e controllo (ridondate per questioni operative, di sicurezza e affidabilità del sistema). Il vapore passa quindi dalla temperatura di circa 190 °C e da 10 bar di pressione a circa 130 °C e 95 mbar (valori indicativi). La turbina è collegata all'alternatore per consentire la generazione di energia elettrica alla frequenza della rete.

A valle della turbina si ha il condensatore, che è mantenuto ad una temperatura di 40 °C, per raffreddare il vapore, ed a 95 mbar (valori indicativi), in modo tale da permettere un maggior salto di pressione alla turbina, che, in sostanza, si traduce in una maggiore e più efficiente produzione di energia. Il condensatore ha il compito di separare il vapore, che si raffredderà e diventerà acqua, dai gas incondensabili, presenti nel fluido in percentuale intorno al 5% (valore variabile in base all'area geotermica). Il raffreddamento avviene tramite l'utilizzo di acqua, proveniente dalle torri di raffreddamento, che viene direttamente spruzzata al suo interno. Dal condensatore si hanno due uscite principali.



**Figura 3.8 condensatore.**

Dalla prima, situata nella parte alta, si ha lo sfogo degli incondensabili che vengono immessi nella tubazione apposita grazie all'azione del compressore di aspirazione, collegato all'asse di rotazione della turbina attraverso un moltiplicatore.

Il compressore riporta i gas incondensabili in condizioni di pressione atmosferica e ad una temperatura di circa 170 °C, condizioni che permettono ai gas di essere efficientemente trattati dal sistema successivo, ovvero il sistema AMIS. L'AMIS (Abbattimento del Mercurio e dell'Idrogeno Solforato) è un impianto non funzionale alla generazione di energia elettrica, ma bensì obbligatorio ai fini di legge, per il rispetto di norme che limitano le emissioni in aria di sostanze nocive. Infatti, i gas incondensabili in uscita dal compressore, hanno forte presenza di mercurio e acido solfidrico ( $H_2S$ ), sostanze dannose per l'ambiente. A seguito del trattamento, che consiste in un filtro ai carboni attivi, che trattiene il mercurio, e in una reazione di ossidazione catalitica, che trasforma in  $SO_2$  l' $H_2S$ , i gas possono essere inviati alle torri di raffreddamento, dove verranno espulsi in aria.

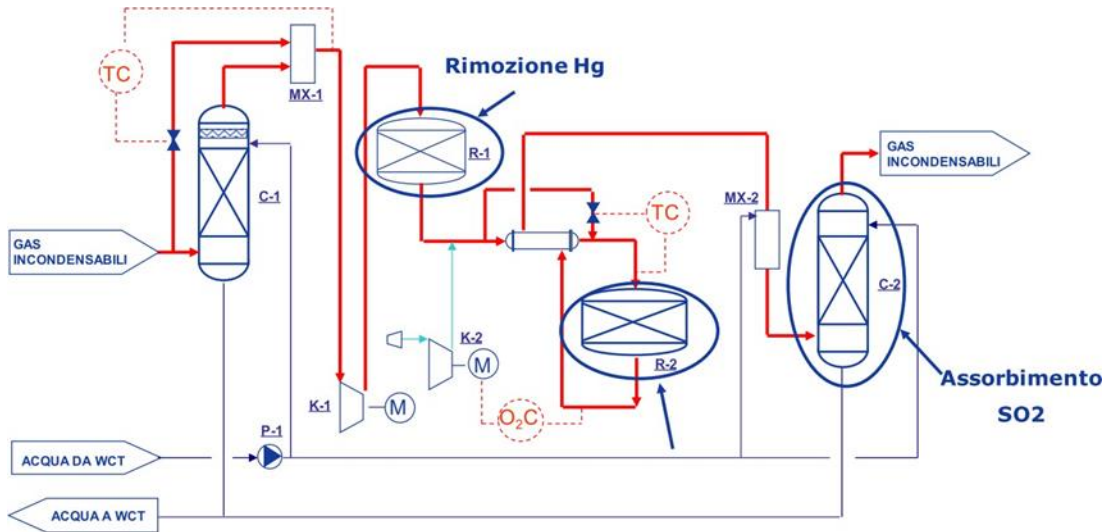


Figura 3.9 schema dell'impianto AMIS.

Dalla seconda uscita del condensatore, posta in basso, si estrae l'acqua di condensa attraverso l'utilizzo di una pompa PAE (Pompa Acqua di Estrazione), una macchina che ha lo scopo di elaborare la portata di acqua per spingerla nelle condutture, fino ad arrivare alle torri di raffreddamento.



Figura 3.10 pompa PAE.

Le torri di raffreddamento quindi, in numero variabile a seconda dell'impianto (comunemente 3 torri), permettono il raffreddamento dell'acqua e l'emissione in ambiente dei gas incondensabili trattati



dall'AMIS. L'acqua ricade, quindi, sotto forma di una vera e propria pioggia, all'interno di queste torri, alla temperatura di 19 °C (valori indicativi), e viene raccolta in una vasca.



**Figura 3.11 torri di raffreddamento.**

Infine, anche per garantire la rinnovabilità del processo, il fluido in uscita dall'impianto è a sua volta raccolto e riportato nel sottosuolo dove verrà nuovamente riscaldato, in maniera naturale, fino a tornare vapore. Questa reintroduzione di acqua nel sottosuolo avviene attraverso degli appositi pozzi, distinti da quelli di produzione, detti appunto "pozzi di reiniezione", che consentono quindi un riutilizzo, seppur non totale, di quello che è il fluido in uscita dagli impianti di produzione di energia.

## 3.2 - DCS di impianto: PI System

Il sistema di gestione dati utilizzato nei siti produttivi del parco geotermico di ENEL GREEN POWER, compreso quello di Rancia 2, impianto pilota del progetto SmartGEO, è il PI System di Osisoft. Si tratta di una collezione organizzata di applicazioni software, che ha la capacità di registrare efficacemente dati provenienti da una pluralità di sistemi di controllo (tra cui i DCS), all'interno di un database di serie temporali. In particolare, con il termine di 'PI Server', viene indicato il cuore stesso del sistema di gestione, dato che esso si occupa di raccogliere, organizzare, storicizzare i dati e di trasformarli in informazioni immediatamente usufruibili dall'utente per monitorare lo stato di funzionamento dell'impianto. Inoltre, se necessario, produce delle notifiche di allarme relative alla deviazione subita dai

parametri in gioco. Nello svolgere le sue funzioni, il PI Server utilizza due componenti: il 'Data Archive' e il 'PIAssetFramework'. Il primo è l'elemento in grado di registrare milioni di dati riguardanti i tag dell'impianto, mantenendone uno storico accurato. AF, invece, trasforma il flusso di dati raccolto in informazioni organizzate, grazie ad un raggruppamento gerarchico delle varie attività secondo una logica object oriented, di cui approfondiremo il significato nel prossimo paragrafo. L'architettura di riferimento di un PI System è illustrata in Figura 3.12.

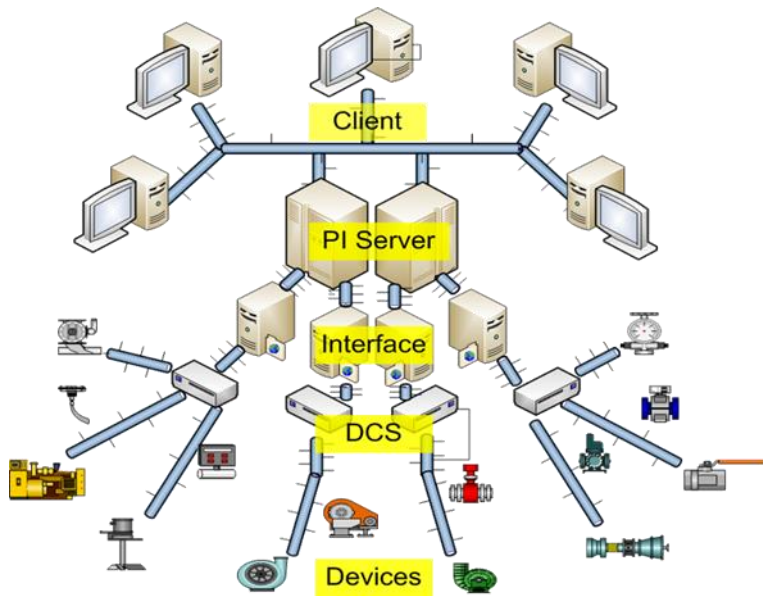


Figura 3.12 Architettura di un PI System su cui si basano anche gli impianti di produzione di energia elettrica di Enel.

Per mezzo degli strumenti forniti da 'Asset Framework', gli utenti possono definire dei modelli (detti template) per ogni dispositivo facente parte dell'impianto di produzione e utilizzare i modelli creati per svolgere analisi più o meno complesse. Infatti, il PI System possiede molte opzioni per lo studio e la visualizzazione dei dati, tra le quali quelle offerte all'interno della cosiddetta 'Asset Analytics'. Quest'ultima, consente di calcolare le principali grandezze statistiche (medie, deviazioni standard, massimi e minimi...), di utilizzare gli operatori logici per costruire espressioni di valutazione sui parametri e di visualizzare i risultati ottenuti sia in tempo reale che in un periodo assegnato nel passato. Questo modo di procedere, di fatto, incentiva la possibilità di confronti tra le condizioni attuali degli asset e quelle che avevano in istanti precedenti, fornendo, così, un importante mezzo di supporto alle decisioni e trasformando il singolo dato in *'insight'*, ovvero in conoscenza del dato stesso.

### 3.2.1 - La logica object-oriented

Dal punto di vista strutturale, un impianto geotermico può essere considerato un insieme di dispositivi interconnessi tra loro le cui differenze, anche in termini di sensori usati (valvole, misuratori di

pressione...), spesso sono solo marginali. Per questo motivo, una caratteristica fondamentale che deve possedere il sistema di diagnostica che “SmartGeo” desidera sviluppare, è quella della scalabilità. A tal fine, sfruttando AF, bisogna creare una rappresentazione accurata della struttura dell’impianto geotermico, che tenga conto dei processi che lo interessano e degli elementi utilizzati al suo interno (tra cui ad es. il compressore, la turbina, il condensatore...). L’idea generale è di possedere una istanza virtuale per ogni dispositivo, che ne contenga sia le informazioni tipicamente strutturali che le caratteristiche utili per la diagnostica.

È qui, allora, che nasce l’altro obiettivo del progetto, che va ad affiancare in modo parallelo quello della costruzione di una metodologia di diagnostica/prognostica: adottare una visione object-oriented per tutti i componenti del sito produttivo. Vediamo meglio, quindi, in cosa consiste questa logica.

Le strutture di dati e le procedure operanti su di esse vengono raccolte all’interno di un’unica entità astratta, la classe, che una volta istanziata permette di creare degli oggetti concreti (in AF detti ‘Elementi’), dotati di proprietà (gli attributi) e di metodi che operano sui dati dell’oggetto stesso. Il risultato è che gli elementi vengono suddivisi secondo classi di funzionalità, rispondendo perfettamente al fatto che gran parte di quelli presenti nei siti del parco geotermico, sono gli stessi. Inoltre, così facendo, diamo luogo a strutture di base omogenee, alle quali aggiungere quando necessario attributi specifici solo del singolo oggetto. Ogni cambiamento introdotto nella classe si propaga in automatico a tutti gli elementi associati, semplificando in modo notevole i procedimenti di aggiornamento dei dati. Per esempio, se supponiamo di avere 100 pompe tutte con le stesse quattro caratteristiche, sarà sufficiente creare un modello (‘template’ in AF) per una sola pompa e poi basare tutte le altre sul modello appena creato. Infatti, le proprietà della classe di base verranno automaticamente ereditate da ogni elemento che fa riferimento a tale classe. Inoltre, se fosse necessario introdurre una modifica nell’elemento ‘pompa’, basta farlo solo al livello del template: i cambiamenti si estenderanno in automatico a tutti gli elementi che esso ha associati. Questa strategia, oltre a dimostrarsi fortemente time-saving, assicura anche una certa consistenza nella definizione degli elementi.

Quindi, l’integrazione tra il sistema di diagnostica e la strutturazione secondo template dell’impianto, accompagna tutto il progetto “SmartGeo” ed è ben rappresentata nella Figura 3.13.

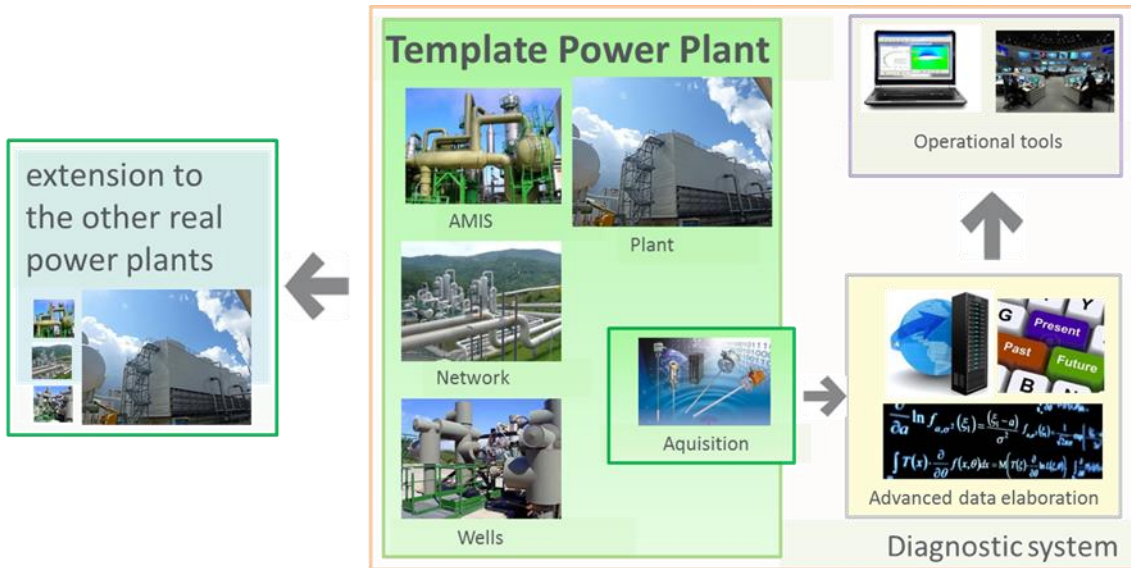


Figura 3.13 Nel progetto vi è l'idea di mantenere in stretto collegamento il sistema di diagnostica con una rappresentazione fedele, secondo classi, dell'impianto di Rancia2. Tale metodo, poi, deve essere esteso a tutti gli altri siti del parco geotermico.

### 3.2.2 - Architettura informatica per modelli soft real-time

Nella versione 2018 del PI System, OsiSoft, per potenziare i modelli analitici presenti sull'Asset Analytics, ha effettuato un accordo con Mathworks, integrando la piattaforma MATLAB Production Server al PI System. In questo modo è possibile sviluppare analitiche direttamente in linguaggio MATLAB, con conseguente incremento della complessità dei modelli sviluppabili, i quali interfacciandosi in lettura e in scrittura con l'AF di OsiSoft, permette ai modelli di accedere ai dati dei sensori di impianto in real-time, di scrivere gli output dei modelli direttamente su PI point e quindi visualizzabili e gestibili dall'AF e di essere eseguiti o meno direttamente da un'unica interfaccia. Come già descritto questo tipo di modelli vengono rilasciati a livello 3 del DCS di impianto, dove è presente la potenza di calcolo necessaria ad elaborare con modelli statistici ed ML i dati storicizzati (PI point) a database.

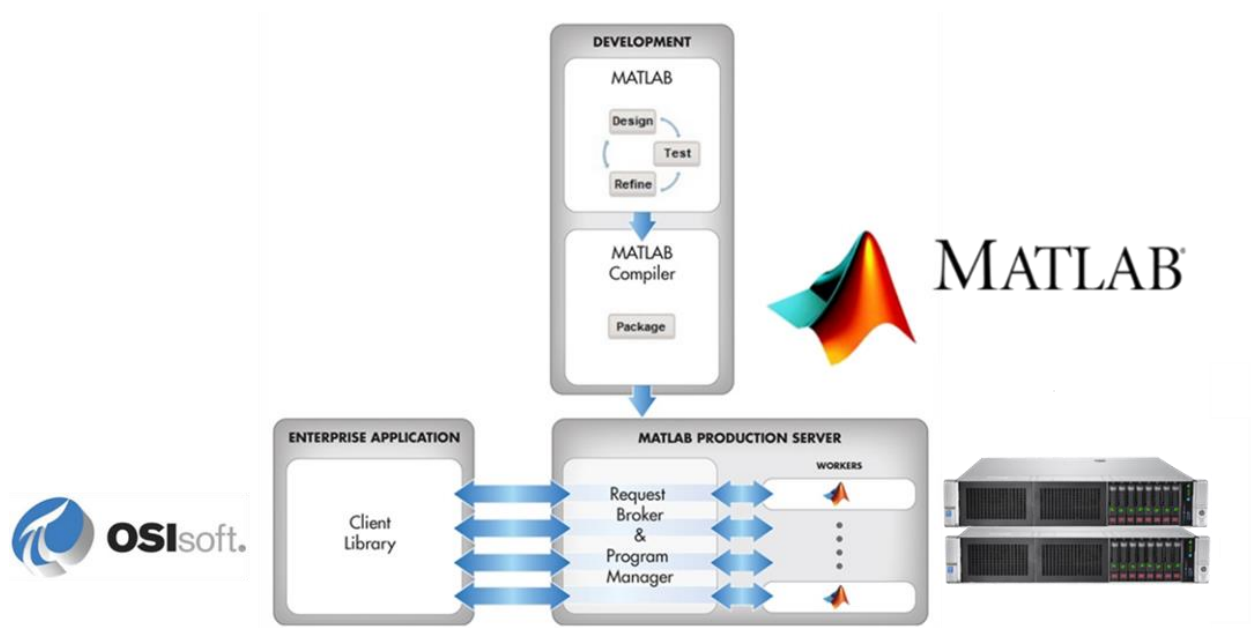


Figura 3.14 Integrazione Matlab Production Server (MPS) e Pi System 2018.

L'utilizzo di questo sistema necessita di un ingente investimento in licenze Mathworks principalmente per:

1. Licenza MATLAB&Simulink con pacchetti di analisi avanzate, per la parallelizzazione degli script e rilascio di applicazioni:
  - a. Intelligenza artificiale, data science e statistiche: Statistics and Machine Learning Toolbox, Deep Learning Toolbox, Reinforcement Learning Toolbox, Deep Learning HDL Toolbox, Text Analytics Toolbox, Predictive Maintenance Toolbox;
  - b. matematica e statistiche: Curve Fitting Toolbox, Optimization Toolbox, Global Optimization Toolbox, Symbolic Math Toolbox, Mapping Toolbox, Partial Differential Equation Toolbox
  - c. calcolo parallelo: Parallel Computing Toolbox, MATLAB Parallel Server
  - d. rilascio di applicazioni: MATLAB Compiler, MATLAB Compiler SDK, MATLAB Production Server, MATLAB Web App Server;

Pacchetti necessari sia all'implementazione dell'integrazione con il PI System che per lo sviluppo di modelli diagnostici/prognostici avanzati che hanno portato EGP a dover investire all'incirca 300k€.

### 3.2.3 - Architettura informatica per modelli hard real-time

In questo caso si è previsto di andare a inserire gli algoritmi sviluppati direttamente online sul controllo del processo d’impianto, quindi a livello 1 della struttura DCS. Questo consentirà sia di fare diagnostica in tempo reale, che di agire direttamente sulle regolazioni di impianto per la massimizzazione dell’efficienza produttiva e la prevenzione dei blocchi.

Questa fase di inferenza e calcolo hard real-time avverrà attraverso l’uso di piattaforme che hanno la capacità di eseguire gli algoritmi sviluppati su Matlab.

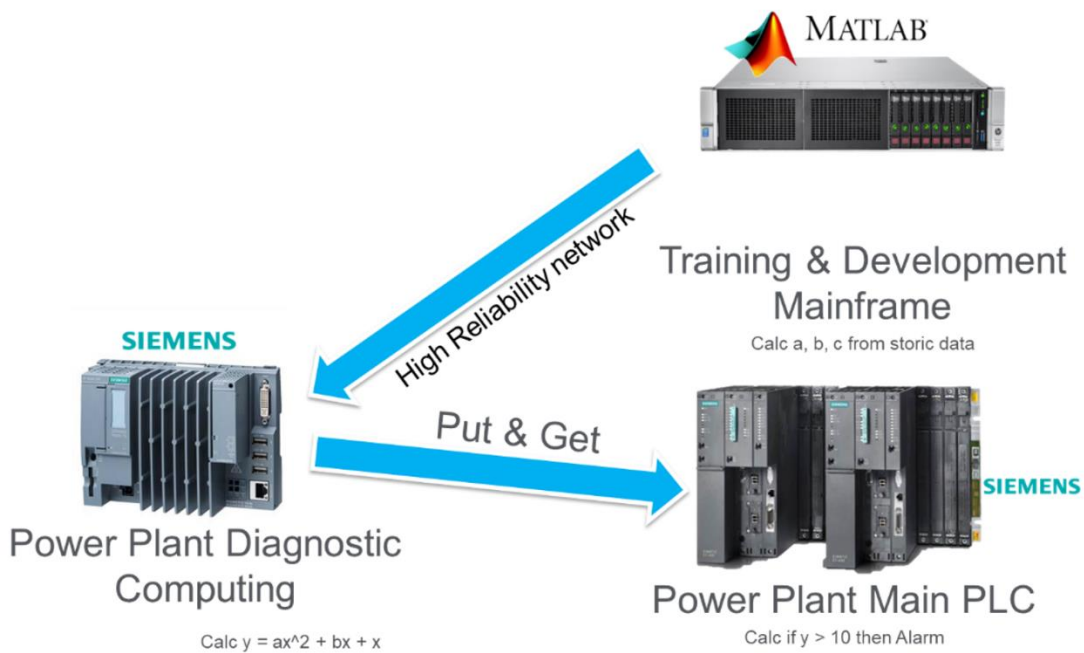


Figura 3.15 Struttura informatica Hard real-time.

Per lo scopo è stato affiancato al PLC principale di impianto (Power Plant Main PLC in figura) un PLC di ultima generazione (SIEMENS SIMATIC STEP 7, Power Plant Diagnostic Computing in figura) che supporta modelli Simulink rilasciati come pacchetti Simulink PLC coder e che permette quindi di sviluppare algoritmi avanzati di diagnostica e controllo. Si è comunque deciso di mantenere anche il PLC principale di impianto per incrementare l’affidabilità del sistema di controllo che funziona anche in presenza di malfunzionamenti del PLC diagnostico.

Tutte le fasi di addestramento, sviluppo e conversione in pacchetti Simulink PLC coder vengono effettuate nel mainframe, quindi a livello 3 del DCS di impianto. I modelli sviluppati e/o i parametri aggiornati sono poi rilasciati sul Power Plant Diagnostic PLC.

Concludiamo, sottolineando quali sono le caratteristiche chiave del PI System che ne giustificano l'utilizzo per l'implementazione di soluzioni di CBM (Condition Based Maintenance) efficaci. Esse si riassumono in:

- capacità di gestione di dati real-time che provengono da una consistente pluralità di fonti operative;
- immagazzinamento fedele dei dati, mantenendo uno storico del ciclo di vita del sensore da cui essi provengono;
- organizzazione dei dati e delle loro analisi secondo classi di funzionalità;
- conversione del dato in informazione sulla condizione di uno specifico asset

### 3.3 - Implementazione del sistema di validazione delle misure

In questo paragrafo viene riportato come è stato implementato in impianto il modello di validazione dei sensori ridondati e la validazione effettuata.

#### 3.3.1 - Diagnostica sui sensori ridondati del condensatore

Nel nostro caso specifico, abbiamo applicato la tecnica di monitoraggio sopra descritta, ai sensori ridondati che misurano il livello di pressione al condensatore dell'impianto di Rancia 2. Qui, vi sono, appunto, 3 dispositivi uguali che contemporaneamente rilevano il livello di pressione presente. I nominativi con cui questi tag compaiono nel database, per facilitare la loro ricerca, sono:


- RA2\_TU\_PT\_301A
- RA2\_TU\_PT\_301B
- RA2\_TU\_PT\_301C

Dove la prima sigla si riferisce al sito produttivo, TU indica il codice dell'area di appartenenza nell'impianto, che in questo caso sta per 'Area Turbina' e, infine, PT identifica la grandezza misurata, ovvero la pressione totale. Per capire che si tratta di sensori ridondati, il codice numerico è seguito alternativamente dalle lettere A, B e C.

Abbiamo caricato sul Server i dati relativi a tali sensori per un periodo che va dall' 1/01/2014 al 26/02/2014, periodo che useremo anche per le valutazioni. Però, a causa di una limitazione nella nostra licenza di utilizzo del database, non siamo legittimati a registrarvi dati e a condurre analisi che si estendano nel passato per un time-range superiore ai 90 giorni. Dunque, le date di registrazione sono state traslate alle seguenti: 1/07/2018 - 27/07/2018. Invece, la scelta di questo intervallo di tempo deriva

dal fatto che è l'unico caso in nostro possesso in cui un sensore ridonato (esattamente il 301B), era stato riconosciuto come difettoso e, perciò, messo fuori servizio in data 21/02/2014. Cercheremo di dimostrare che la nostra tecnica di riconoscimento, è in grado di dedurre con un certo anticipo il malfunzionamento al sensore rispetto all'individuazione fatta dagli operatori di esercizio.

### 3.3.2 - Analisi per i sensori ridonati

Dopo aver caricato i dati di interesse, ci siamo concentrati sul costruire in AF i singoli attributi necessari per l'applicazione della tecnica di previsione. Alcuni di questi hanno un valore costante, altri, invece, sono il risultato dei calcoli effettuati con un tipo di analisi detto 'Expression Analysis', a disposizione all'interno dell'Asset Analytics. In tal caso, essi avranno indicata nel campo di referenza dell'attributo ('Data Reference') la parola 'Analysis' e a sinistra del loro nome comparirà questo simbolo .

#### 3.3.2.1 Pairwise\_comparison

Per poter valutare in ogni istante la differenza a coppie tra i sensori, abbiamo creato una prima analisi che calcolasse queste differenze in maniera sincrona. Il nome assegnato all'analisi è "Pairwise\_comparison" e il risultato ottenuto viene memorizzato all'interno di tre nuovi attributi di tipo PI point:

- BIAS\_AB;
- BIAS\_AC;
- BIAS\_BC

I PI point, di fatto, costituiscono il corpo del PI System, poiché sono collegati direttamente al server e contengono lo storico dei dati che hanno associati.

Riportiamo in Figura 3.16 un esempio di questa analisi. Cliccando il tasto 'Evaluate', le grandezze che compaiono nell'espressione vengono valutate nell'istante attuale, in altre parole nel momento stesso in cui si conduce l'analisi. Per conoscerne il valore in istanti di tempo precedenti, è sufficiente visualizzare i loro risultati passati cliccando con il tasto destro sull'attributo e scegliendo alternativamente l'opzione 'Preview Results' o 'Trend'; infine andrà specificato un opportuno intervallo di tempo. Si comprende



come ciò renda efficacemente possibili i confronti tra la situazione attuale e le condizioni passate.

Name	Expression	Value at Evaluation	Value at Last Trial	Output Attribute
biasAB	('.\Pressure Sensor A Value'-. \Pressure Sensor B Va	-12,161	-12,161	BIAS AB
biasAC	('.\Pressure Sensor A Value'-. \Pressure Sensor C Va	-12,238	-12,238	BIAS AC
biasBC	('.\Pressure Sensor B Value'-. \Pressure Sensor C Va	-0,077103	-0,077103	BIAS BC

Figura 3.16 Analisi che confronta istante per istante la differenza a coppie tra i sensori e ne restituisce il valore al momento stesso della valutazione.

### 3.3.2.2 Analisi per l'addestramento al test diagnostico

Abbiamo, poi, definito con il nome di "Train\_DiagTest" una seconda analisi di tipo 'Expression'. Questa ha il compito di addestrare il sistema al test di diagnostica. Riportiamo tale analisi in Figura 3.17 e vediamo nel dettaglio come funziona.

Name	Expression	Value at Evaluation	Value at Last Trigger	Output Attribute
TobeTrainedNow	if 'TobeTrained TobeTrainedIn' then true else false	False	False	Map
if 'TobeTrained TobeTrainedIn' then true else false				
meanAB	if (Tobe_TrainedNow) then TagAvg('BIAS_AB', 'FromD	-	-	DIFF AVG AB
meanAC	if (Tobe_TrainedNow) then TagAvg('BIAS_AC', 'FromD	-	-	DIFF AVG AC
meanBC	if (Tobe_TrainedNow) then TagAvg('BIAS_BC', 'FromD	-	-	DIFF AVG BC
deviationAB	if (Tobe_TrainedNow) then StDev('BIAS_AB', 'FromD	-	-	DIFF DEV AB
deviationAC	if (Tobe_TrainedNow) then StDev('BIAS_AC', 'FromD	-	-	DIFF DEV AC
deviationBC	if (Tobe_TrainedNow) then StDev('BIAS_BC', 'FromD	-	-	DIFF DEV BC
diffAB3Sigma	if (Tobe_TrainedNow) then meanAB+ 'L'*deviationAB	-	-	UCL AB
diffAC3Sigma	if (Tobe_TrainedNow) then meanAC+ 'L'*deviationAC	-	-	UCL AC
diffBC3Sigma	if (Tobe_TrainedNow) then meanBC+ 'L'*deviationBC	-	-	UCL BC
diffABminus3	if (Tobe_TrainedNow) then meanAB- 'L'*deviationAB	-	-	LCL AB
diffACminus3	if (Tobe_TrainedNow) then meanAC- 'L'*deviationAC	-	-	LCL AC
diffBCminus3	if (Tobe_TrainedNow) then meanBC- 'L'*deviationBC	-	-	LCL BC
DateTrained	if (Tobe_TrainedNow) then '*' else NoOutput()	-	-	DateTrained
TrainedNow	if (Tobe_TrainedNow) then true else NoOutput()	-	-	Trained
TobeTrained	if (Tobe_TrainedNow) then false else NoOutput()	-	-	TobeTrained
NeedsTraining	if (Tobe_TrainedNow) then false else NoOutput()	-	-	NeedsTraining
NTraining_Rqst	if (Tobe_TrainedNow) then ('NTrainingRqst NTraini	-	-	NTrainingRqst

Evaluation Time: 07/10/2018 09:56:22 Last Trigger Time: 05/10/2018 16:04:01 Elapsed Evaluation Time: 12ms

Figura 3.17 Sono rappresentate tutte le espressioni dell’analisi che serve per addestrare il sistema al test di diagnostica.

La routine di addestramento è innescata da uno specifico attributo-PI, chiamato “TobeTrainedIn”. Quest’ultimo svolge appunto il ruolo di ‘trigger’ per l’analisi ed esiste come child-attribute di un’altra variabile, di nome “TobeTrained”. In particolare, esso è un attributo figlio con la proprietà aggiuntiva di apparire nascosto (‘hidden-attribute’) rispetto al genitore. Si può osservare tale legame nella Figura 3.18, dove il simbolo per riconoscere l’attributo nascosto è

Filter	
Name	Value
BIAS_AB	-12,161
TagName	UNIFI_RA2_TU_PT_301.BIAS_AB
BIAS_AC	-12,238
TagName	UNIFI_RA2_TU_PT_301.BIAS_AC
BIAS_BC	-0,077103
TagName	UNIFI_RA2_TU_PT_301.BIAS_BC
DateTrained	05/10/2018 16:04:01.759
FromDateTrain	10/07/2018 15:57:09
L	3
NeedsTraining	False
NeedsTrainingIn	False
NTrainingRqst	233
NTrainingRqstIn	233
TobeTrained	False
TobeTrainedIn	False
ToDateTrain	01/08/2018 15:56:41
Trained	True
TrainedIn	True

Figura 3.18 Nell'immagine si nota che l'attributo "TobeTrainedIn" è figlio dell'attributo "TobeTrained", con la proprietà aggiuntiva di essere non visibile ('hidden').

Se si assegna ad un attributo la proprietà 'hidden', questo non potrà più essere recuperato durante una ricerca, proprio perché non visibile. La nostra scelta di tale configurazione, deriva dall'esigenza di arginare il problema della circolarità, ovvero il rischio che una routine si richiami all'infinito. Infatti, per poter stabilire con 'Asset Analytics' un'analisi che abbia come punto di partenza della routine il cambiamento di una variabile che, poi, a sua volta si modifica (si noti la forte dinamicità dell'attributo "TobeTrained"), è necessario distinguere due modi diversi con cui chiamare l'attributo di trigger (ecco il motivo della creazione di "TobeTrainedIn"). Così facendo, il sistema non vede più il riferimento circolare e ci consente di utilizzare il cambiamento del child-attribute "TobeTrainedIn" come parametro di input di una funzione, anche se questo parametro varierà a sua volta. Il risultato è che l'attributo figlio non visibile punta allo stesso PI point dell'attributo padre, con la differenza, però, che solo sul genitore è possibile scrivere i nuovi dati, perché il figlio è impostato nella modalità di sola lettura dei valori (detta 'Read only'). Allora, quello che accade nell'analisi, è che nel momento in cui "TobeTrainedIn" cambia valore, esso chiama il suo attributo padre che alternativamente può essere vero o falso. Solo nel caso in cui questo risulti vero, viene valutata la prima espressione che compare in Figura 3.17 e inserito il risultato all'interno della variabile "TobeTrainedNow", a significare che l'addestramento deve essere fatto ora. Si passa, perciò, a

calcolare il limite superiore (UCL) da usare per il test sulle varie coppie di differenze. Come visto durante la spiegazione del metodo (paragrafo 2.5.1 -), il range di controllo deve riferirsi ad un periodo di funzionamento a regime dell'impianto. Perciò, si creano anche due attributi variabili, rispettivamente "FromDateTrain" e "ToDateTrain", che indicano l'uno la prima data in cui è stato svolto il test di diagnosi, l'altro l'ultima data dell'intervallo utilizzato per valutare l'espressione. E si salvano entrambi come PI point. L' UCL verrà, appunto, determinato entro il periodo tra le due date. Esso, a sua volta, si ottiene dalla somma tra la media delle differenze tra i sensori e la moltiplicazione della deviazione standard della differenza per un opportuno scalare. Quest'ultimo è un valore costante, che poniamo pari a 3 e inseriamo in un attributo chiamato "L", che indica appunto il leveraggio del sensore. Poi, in sequenza vengono calcolati e salvati in PI point:

- La media della differenza per ogni coppia di sensori →  $DIFF\_AVG\_AB / DIFF\_AVG\_AC / DIFF\_AVG\_BC$
- La deviazione standard della differenza →  $DIFF\_DEV\_AB / DIFF\_DEV\_AC / DIFF\_DEV\_BC$
- Il limite superiore per ogni coppia (che usiamo nella tecnica di monitoraggio) →  $UCL\_AB / UCL\_AC / UCL\_BC$
- Il limite inferiore per ogni coppia (che, invece, non viene usato nella tecnica) →  $LCL\_AB / LCL\_AC / LCL\_BC$

A questo punto, grazie alle ultime 5 espressioni dell'analisi, si riassumono in attributi-PI le informazioni relative all'addestramento effettuato. La prima memorizza la data attuale dell'addestramento all'interno del cosiddetto "DateTrained". La seconda espressione, invece, si occupa di segnalare che la routine è stata svolta. Quindi, pone uguale a 'True' un ulteriore attributo, chiamato "Trained". Poi, dato che il sistema è stato ormai addestrato, la routine può essere interrotta e, dunque, si assegna a "TobeTrained" il valore di 'False' (terza espressione). La quarta funzione, che valuta se il sistema ha bisogno di ricevere un training ("NeedsTraining" è il valore di output), ha a che fare con un aspetto che noi non abbiamo considerato, ma che in futuro dovrebbe essere aggiunto alla diagnostica. Infatti, quando le coppie di differenze osservate scendono al di sotto del limite di controllo inferiore (LCL), significa che il processo stocastico alla base ha subito un cambiamento. O meglio, si può dire che il processo sta lavorando su un altro livello, che non è uguale a quello che incrementa la differenza (e che, quindi, giustamente genera un allarme), ma comunque degno da segnalare. Ciò perché, trovandoci sotto il limite di controllo, se la differenza dovesse riaumentare, è probabile che il sistema non se ne accorga e, quindi, che valga la pena riaddestrarlo. Per questo motivo, il calcolo del limite di controllo è già stato previsto nell'analisi, anche se esso non viene direttamente utilizzato nel nostro test. Infine, l'ultima espressione serve per informare l'utente sulla quantità di volte in cui storicamente il sistema ha subito un addestramento. Ad ogni nuovo training, l'output "NTrainingRqst" viene incrementato di un'unità.

Si osservi dalla Figura 3.18 Nell'immagine si nota che l'attributo "TobeTrainedIn" è figlio dell'attributo "TobeTrained", con la proprietà aggiuntiva di essere non visibile ('hidden')., che anche gli attributi "NeedsTraining", "NTrainingRqst" e "Trained" sono genitori di child-attribute posti nella modalità 'hidden'. Ciò, infatti, ci lascia la possibilità di usare le variabili a sinistra e a destra della stessa espressione, senza far nascere un riferimento circolare.

### 3.3.2.3 Analisi per l'applicazione del test diagnostico

L'ultima 'Expression Analysis' che abbiamo definito, è quella che nella pratica serve a fornire il valore di pressione risultante dal modello di diagnostica. Perciò, si sono dovuti considerare tutti i casi ottenuti dalla tabella della verità (Tabella 2.2) per avere in uscita sia la misura che la qualità della misura. Queste sono state da noi indicate come "Value" ed "HealthStatus" ed entrambe memorizzate in uno specifico PI point. Inoltre, abbiamo predisposto un punteggio ("HealthScore") per la misura ottenuta, in relazione alla qualità che viene ritornata in uscita. In particolare, assegniamo ovviamente valore massimo (100) al caso in cui la qualità sia 'Healthy', 50 se viene prodotto uno 'Warning' e 0 in presenza di un errore. Anche il punteggio viene registrato in un PI point.

La funzione che restituisce il valore da preferire come pressione finale è riportata in Figura 3.19.

FinalValue	(Threshold1)or(Threshold2)or(Threshold3)	False	False	Final_Diag
ReliableValue	if(FinalValue=False)Then(RelValue1)else(if Threshc	97,537	97,537	Value

```

if(FinalValue=False)Then(RelValue1)else(if Threshold1 and Threshold2= True then(RelValue2_3)else
(if Threshold1 and Threshold3= True then(RelValue2_2)else(if Threshold2 and Threshold3=True then
(RelValue2_1)else(if Threshold1=True then (RelValue3_1)else(if Threshold2=True then
(RelValue3_2)else(if Threshold3=True then(RelValue3_3)else(false))))))
    
```

Evaluation Time: 03/10/2018 09:29:56 Last Trigger Time: 02/10/2018 16:50:21 Elapsed Evaluation Time: 0,9ms

Figura 3.19 Si vede in figura l'espressione che permette di ottenere il valore finale di output, una volta che vengono valutate tutte le possibilità presenti nella tabella della verità.

Le espressioni per il calcolo della qualità e del punteggio, invece, sono le seguenti (Figura 3.20).

Healthstatus	<code>if(FinalValue=False)Then("Healthy")else(if Threshc</code>	Healthy	Healthy	Health Status
<pre> if(FinalValue=False)Then("Healthy")else(if Threshold1 and Threshold2 =True then ("Error")else(if Threshold1 and Threshold3 = True then("Error")else(if Threshold2 and Threshold3=True then ("Error")else(if Threshold1=True then("Warning")else(if Threshold2=True then("Warning")else(if Threshold3=True then("Warning")else(false))))))         </pre>				
Evaluation Time: 03/10/2018 09:29:56 Last Trigger Time: 02/10/2018 16:50:21 Elapsed Evaluation Time: 0,9ms				

Healthscore	<code>if(Health_status="Healthy")then(100)else(if(Health_status="Error")then(0)else(</code>
<pre> if(Health_status="Healthy")then(100)else(if(Health_status="Error")then(0)else(if (Health_status="Warning")then(50)else(10)))         </pre>	

Figura 3.20 Le funzioni che compaiono in figura, servono a calcolare la qualità e il punteggio del valore di output. Il risultato tiene conto di tutti i casi ottenuti dalla tabella della verità.

Le variabili "Threshold 1/2/3" servono per capire se le varie coppie di differenze abbiano superato o meno la soglia. Il risultato di questa valutazione viene riportato in tre differenti attributi, che sono: THR\_RES\_AB; THR\_RES\_AC e THR\_RES\_BC, di cui non importa salvare lo storico. Si comprende meglio il funzionamento dell'analisi, osservandone un esempio completo in Figura 3.21.

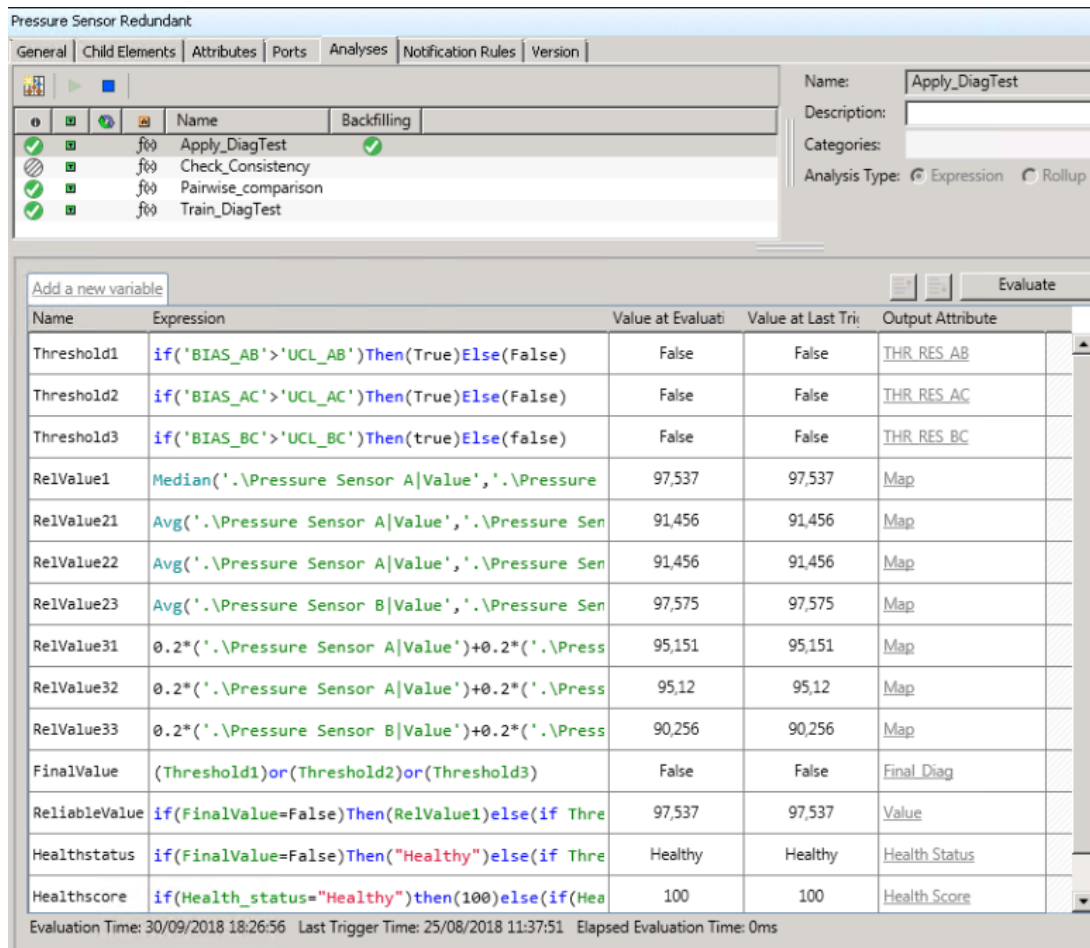


Figura 3.21 E' qui rappresentata l'analisi completa che porta al calcolo del valore finale, della sua qualità e del suo punteggio. Il risultato ottenuto deriva dall'aver prima valutato tutti i casi legati ai superamenti di soglia.

In questo caso, avendo ottenuto 'False' per la variabile "Valore finale", ci troviamo nella prima categoria: nessuna coppia di differenze supera la soglia e, perciò, la pressione totale è data dalla mediana tra i valori dei tre sensori.

L' esempio descritto dà una visione in tempo reale sulla condizione dei sensori di pressione del condensatore. Invece, per dimostrare del tutto l'efficacia della tecnica diagnostica, abbiamo dovuto studiare l'andamento delle coppie di differenze in un sottoinsieme nel passato dell'intervallo di tempo in cui sono stati caricati i dati sul Server. Riassumiamo i risultati ottenuti nel paragrafo successivo e vediamo quanto prima il metodo prevede il guasto al sensore.

### 3.3.3 - Efficacia del metodo di diagnostica predittiva

Come già anticipato, dato che conoscevamo la data (21/02/2014) in cui era stato rilevato dagli operatori un guasto al sensore 301B, abbiamo ritenuto opportuno visualizzare l'andamento dei dati in un periodo

immediatamente precedente, cosicché potessimo constatare l'effettiva capacità del metodo di prevedere le eventuali anomalie. Perciò, rappresentiamo in Figura 3.22, Figura 3.23 e Figura 3.24 i trend dei valori assunti dalle coppie di differenze ("bias AB/AC/BC") tra il 5/02/2014 e il 19/02/2014, ricordando che tali date compaiono sui grafici opportunamente traslate. Inoltre, negli stessi andamenti sono stati aggiunti i valori dei limiti di controllo superiori contenuti negli attributi-PI e indicati con: UCL\_AB; UCL\_AC e UCL\_BC.

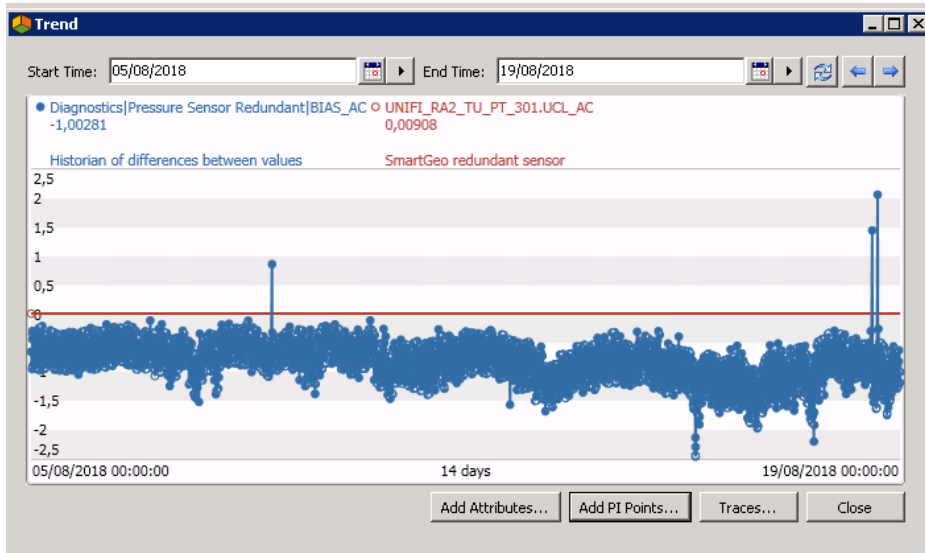


Figura 3.22 La figura rappresenta l'andamento della differenza tra A e C, nel periodo di tempo selezionato. La riga rossa orizzontale, invece, indica il valore di UCL\_AC che è stato usato per il test diagnostico.

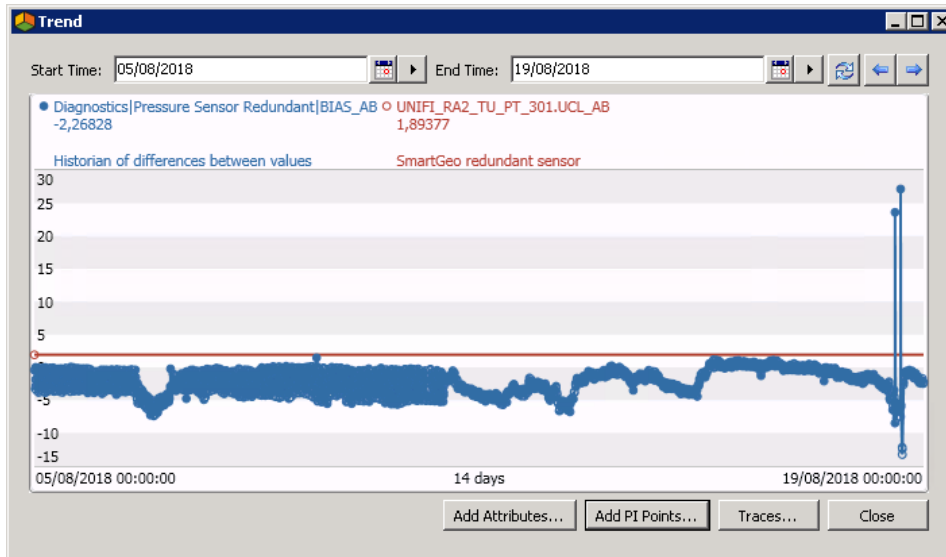


Figura 3.23 La figura rappresenta l'andamento della differenza tra A e B, nel periodo di tempo selezionato. La riga rossa orizzontale, invece, indica il valore di UCL\_AB che è stato usato per il test diagnostico.



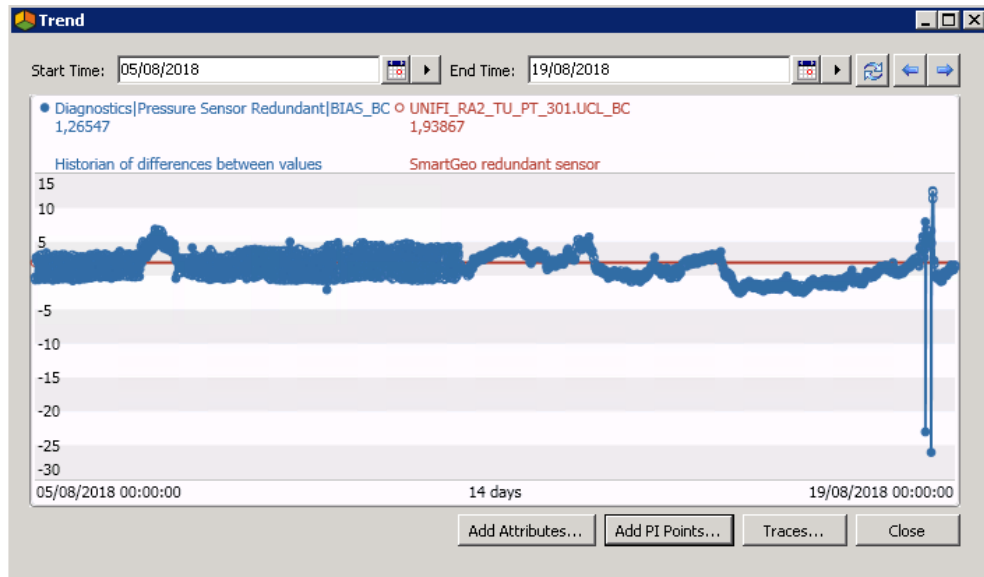


Figura 3.24 La figura rappresenta l'andamento della differenza tra B e C, nel periodo di tempo selezionato. La riga rossa orizzontale, invece, indica il valore di UCL<sub>BC</sub> che è stato usato per il test diagnostico.

I grafici ottenuti possono essere considerati esempi delle carte di controllo di Shewhart per misure singole. Quest'ultime, hanno uno stretto legame con il test d'ipotesi che monitora il funzionamento corretto dei sensori. Infatti, se il valore osservato per i vari "bias" cade al di fuori del limite di controllo superiore, saremo portati a rifiutare l'ipotesi nulla e, quindi, a pensare che in quell'istante i due sensori stiano fornendo valori significativamente diversi. Inoltre, in questo caso, si parla di carte di controllo a campione unitario, perché viene impiegata una modalità di misurazione automatica che controlla tutti i valori prodotti dai sensori nell'intervallo di tempo di valutazione (e non soltanto un loro sottogruppo).

Analizzando, adesso, i tre grafici si nota immediatamente una differenza tra loro. In particolare, quello associato alla coppia (B – C) ha un numero nettamente più consistente di allarmi. Infatti, mentre la differenza (A – B) non supera la soglia per tutto il periodo compreso tra il 5/02 e il 18/02, e così lo stesso per la coppia (A – C), se pur con un unico allarme in data 8 febbraio; (B – C) va sopra soglia già all'inizio del periodo di esame. Si può, quindi, intuire che il sensore B o il C stiano avendo un malfunzionamento. L'allarme associato a questa coppia è rientrato e riscattato alcune volte, con periodi anche di qualche giorno di misure sottosoglia. Per stabilire, però, in maniera univoca quale dei due sensori probabilmente si sia guastato, occorre che almeno 2 coppie su 3 restituiscano valori fuori controllo. Questo accade nel pomeriggio del 18 febbraio, dove sia AB che BC eccedono il range: il sensore B, dato dalla loro intersezione, è quello difettoso. In conclusione, se gli operatori di impianto avessero adottato la nostra tecnica di monitoraggio, avrebbero potuto intuire un comportamento anomalo dei sensori ridondati già a partire dal 5/02, dove si ha un periodo di circa sei giorni (5-11 febbraio) in cui la coppia BC supera la soglia. Il guasto sarebbe poi stato completamente rilevato il 18/02, ma in ogni caso prima della data di

messa fuori servizio. Il metodo, perciò, dimostra una capacità di prevedere il verificarsi delle anomalie con un anticipo di ben tre giorni rispetto alla situazione attuale.

La sensibilità del modello dipende ovviamente dalla scelta fatta per il moltiplicatore “L”, che contribuisce a definire il limite di controllo superiore per il test. Tuttavia, avendo noi preso per quest’ultimo un valore pari a 3, la possibilità che si abbiano falsi positivi si riduce e, in questo caso specifico, la tecnica avrebbe condotto ad effetti migliorativi.

### 3.4 - Analisi delle problematiche di impianto

L’obiettivo del progetto di dottorato e del progetto SmartGEO è quello di migliorare e progettare, ove mancanti, le metodologie di diagnostica per garantire un’affidabilità maggiore per tutti gli elementi definiti come “ausiliari di impianto”, ovvero tutta la componentistica non facente parte dell’asse generatore-turbina, sistemi che già adottano soluzioni avanzate e sviluppate ad hoc nel corso degli anni.

Per riuscire a capire quali sono le problematiche che si riscontrano e quali strategie diagnostiche sono attuate ci siamo avvalsi dell’esperienza degli operatori di impianto e di quelli del centro di controllo di Larderello.

Seguiamo idealmente quello che è il ciclo geotermico per elencare quelle difficoltà che sono state fin da subito riportate come critiche. Il primo riguarda le tecniche di lavaggio del vapore direttamente in pozzo, che evidenziano difficoltà nel corretto dosaggio della miscela acqua-soda con conseguente rischio di “affogamento” o in casi più gravi anche di danni strutturali al pozzo stesso. Questi problemi risultano essere abbastanza frequenti e necessitano anche di competenze di tipo geologico per poter essere meglio affrontati.

Passando alle reti di distribuzione del vapore il problema che sembrava più preoccupante era quello del controllo dello spessore delle tubazioni. Infatti, il vapore trasportato, per sua natura, può corrodere i tubi in cui scorre. Nel caso in cui la corrosione sia eccessiva si rischiano di avere perdite di vapore, con conseguenze economiche oltre che ovviamente ambientali dato che il vapore in questo punto non è in condizioni consone all’emissione in aria. Al momento l’unica diagnostica che è attuata sono controlli periodici puntuali su alcuni punti significativi della rete. È inoltre possibile il monitoraggio della portata delle tubazioni in real time anche se solo in alcuni punti della rete e con affidabilità non ottima.

Giungendo invece alle componenti principali oggetto del progetto SmartGeo, ovvero gli ausiliari di impianto, si evidenziano problemi alle valvole di controllo in ingresso alla turbina e a quelle successive alla

pompa PAE. A causa dell'aggressività del fluido sono soggette a corrosione e frequentemente si ricoprono di incrostazioni che ne ostacolano il corretto azionamento. L'unico aspetto positivo è che alcune di queste valvole sono ridondate, quindi il sistema risulta essere sufficientemente affidabile.

Un altro componente che risulta problematico è il compressore di aspirazione di incondensabili dal condensatore, in quanto in fase di esercizio può aspirare oltre agli incondensabili, anche piccole quantità di acqua che causano incrostazioni sulle pale che risultano dannose. È stato inoltre evidenziato un problema nell'identificazione del fenomeno di pompaggio, che risulta essere molto frequente a causa dell'inserimento dell'impianto di trattamento dei gas in uscita dal compressore (AMIS), non previsto in fase di progetto dell'impianto.

Quindi possiamo direttamente ad analizzare la pompa PAE. Questa macchina, secondo l'esperienza degli operatori, sembra essere robusta dal punto di vista affidabilistico, ma il fatto che non sia in alcun modo ridondata fa sì che risulti vitale ridurre al minimo le fermate, dato che ogni ora in cui la pompa non è in funzione comporta un'ora di fermo impianto e conseguentemente grandi perdite economiche di esercizio. Inoltre, la macchina ha anche un costo capitale rilevante e ad ogni danno grave può corrispondere un costo non trascurabile di riparazione o ricondizionamento. I problemi che sono emersi dal brainstorming condotto in più sedute per sviluppare l'analisi FMECA, riguardano principalmente i cuscinetti polimerici di strisciamento che si trovano al suo interno, e che tendono a danneggiarsi con una certa frequenza, oltre a problemi di esercizio quali la cavitazione, per la quale i tecnici ENEL GREEN POWER riterrebbero utile la progettazione di un metodo diagnostico apposito. Ultimo sistema da analizzare è quello delle torri di raffreddamento, in particolare ci interessano i riduttori che collegano i motori elettrici alle pale, che sembra siano soggetti a danni frequenti, localizzati sulle ruote dentate e i cuscinetti. Il fatto che le torri siano in numero maggiore di 2 e che non sia necessario che tutte funzionino insieme rende la gravità del guasto meno rilevante (affidabilità 2 out of 3). Attualmente sono mantenute in via straordinaria solo al momento della avvenuta rottura.

L'impianto AMIS non sembra avere particolari problemi affidabilistici, piuttosto sembra che non sia efficiente l'utilizzo della soda. Questo problema però non necessita di particolari accorgimenti diagnostici e quindi non lo analizzeremo in questa tesi.

Da tenere presente, anche ai fini del nostro studio, che sono definite per tutti gli impianti delle cadenze di fermata regolare nelle quali si tende a sostituire o mantenere tutto quello che non è ritenuto "in salute". Di solito la cadenza di fermata è imposta ogni 4-5 anni, anche per vincoli di tipo legislativo che impongono una certa frequenza di manutenzione, ma altre fermate di impianto più corte sono schedate con più frequenza per permettere alcune manutenzioni straordinarie.

### 3.4.1 - Analisi FMECA

Per cercare di standardizzare la procedura ed evidenziare tutti i possibili modi di guasto dell'impianto è stata condotta una analisi FMECA (Failure Mode, Effects, and Criticality Analysis) per tutti i componenti giudicati di rilevanza per il progetto. Perciò abbiamo creato una tabella che è stata successivamente riempita attraverso le indicazioni degli esperti di ogni componente.

Id	AGE (L0)	Area Funzionale (L1)	Sistema e device (L2)	Riferimento SAP PM	Sottosistema (L3)
1.1.1.1.1	Rancia 2	Ciclo acqua	Torre refrigerante elettroventilatori	ITG-RAN2-01-07-VT1, ITG-RAN2-01-07-V T2, ITG-RAN2-01-07-VT3	Gearbox
1.1.1.2.1	Rancia 2	Ciclo acqua	Torre refrigerante elettroventilatori	ITG-RAN2-01-07-VT1, ITG-RAN2-01-07-V T2, ITG-RAN2-01-07-VT3	Gearbox
1.1.1.3.1	Rancia 2	Ciclo acqua	Torre refrigerante elettroventilatori	ITG-RAN2-01-07-VT1, ITG-RAN2-01-07-V T2, ITG-RAN2-01-07-VT3	Gearbox

Sistema	Componente (L4)	Modo di guasto	Cause	Effetto locale (L4)	Effetto Sottosistema (L3)	Effetto Sistema
gearbox	Sistema chiavette	Rottura della chiavetta litorale	Lubrificazione, colpi di vento, cambiamenti di assetto bruschi	Interruzione accoppiamento	Interruzione trasmissione del moto	Interruzione
gearbox	Copie cilindriche asse veloce	Rottura di un dente	problemi lubrificazione, usura temporale	grippaggio	Bloccaggio trasmissione del moto	Interruzione
gearbox	Copie cilindriche asse lento	Rottura di un dente	problemi lubrificazione, usura temporale	vibrazione, degrado, grippaggio		

effetti			
ma (L3)	Effetto Sistema device (L2)	Effetto area funzionale (L1)	Effetto finale (L0)
Interruzione	Interruzione friggio/refrigerazione vapore		
Interruzione	Interruzione friggio/refrigerazione vapore		

Indice di Occorrenza	Indici di Severità		
	Frequenza	Indice di frequenza	Quantitativo Capex
molto rara	1	1	1
0.25/4 anni	2	1	1
0.15/4 anni	2	1	1

Quantitativo Capex [€]	Quantitativo Opex2 [€]	Downtime di manutenzione preventiva	generalizzabilità dell'analisi	Sensori di diagnostica già adottati	Indice di criticità
			buona	vibrazioni, temperatura	
			buona	vibrazioni asse veloce, temperatura	
			buona	vibrazioni asse lento, temperatura	

Indice di Detectability	Modalità di rilevamento guasto	Controlli di prevenzione	Azioni manutentive effettuate	Migliorie di impianto consigliate	Indice di criticità

Figura 3.25 tabella FMEA di prodotto.

Abbiamo optato per l'utilizzo di una forma della tabella di tipo lineare. Questo layout risulta efficace nel nostro caso in quanto il suo utilizzo risulta molto semplice e quindi adottabile anche da parte degli operatori di ENEL GREEN POWER, dato che questa tabella dovrà poi entrare nelle routine interne all'azienda applicate per censire i guasti futuri. La principale pecca di questa scelta è il fatto che si rischiano ripetizioni di guasti di ugual natura che si verificano su più elementi, che portano chiaramente alla duplicazione di alcune righe della tabella.

Un layout forse più semplice ed elegante sarebbe risultato quello gerarchico indentato, con il quale si sarebbe aggirato il problema delle duplicazioni di righe. Per poter però usufruire a pieno delle sue potenzialità sarebbe stata necessaria l'introduzione di un ulteriore software specifico, cosa che abbiamo ritenuto complicata e quindi in questa prima fase la abbiamo esclusa.

Di seguito si riporta una tabella in cui si esplicano il significato e le linee guida adottate per la corretta compilazione dell'FMECA di prodotto.

**Tabella 3.1** descrizione dei campi presenti nella tabella FMECA di prodotto.

<b>Campo FMECA</b>	<b>Descrizione</b>
Id	Codice numerico di 4 cifre separate dal segno “.” ognuna delle quali identifica in ordine: il <b>sistema e device</b> , il <b>sottosistema</b> , il <b>componente</b> ed il <b>modo di guasto</b> considerato.
Impianto (L0)	Impianto geotermico analizzato (Livello 0)
Area Funzionale (L1)	Area funzionale dell' <b>Impianto</b> analizzato (Livello 1)
Sistema e device (L2)	Nome del <b>sistema dell'area funzionale</b> analizzata (Livello 2)
Riferimento SAP PM	Codice alfanumerico SAP identificativo del <b>sistema</b> analizzato
Sottosistema (L3)	Nome del <b>sottosistema</b> del <b>device</b> analizzato (Livello 3)
Componente (L4)	Nome del <b>componente</b> del <b>Sottosistema</b> analizzato (Livello 4)
Modo di guasto	Descrizione del <b>modo di guasto</b> del <b>componente</b> analizzato. Ogni componente può avere più modi di guasto che vengono aggiunti inserendo una nuova riga nella tabella
Cause	Descrizione di tutte le possibili cause del particolare <b>modo di guasto</b>
Effetto Locale (L4)	Descrizione dell'effetto che il <b>modo di guasto</b> analizzato produce sul <b>componente</b> (livello 4)
Effetto Sottosistema (L3)	Descrizione dell'effetto che il <b>modo di guasto</b> analizzato produce sul sottosistema di cui il <b>componente</b> fa parte (Livello 3).  NB: Ad uno stesso <b>Effetto Locale</b> dovrà corrispondere uno stesso <b>Effetto Sottosistema</b> .
Effetto Sistema device (L2)	Descrizione dell'effetto che il <b>modo di guasto</b> analizzato produce sul <b>Sistema e Device</b> di cui il <b>componente</b> fa parte (Livello2).  NB: Ad uno stesso <b>Effetto Sottosistema</b> dovrà corrispondere uno stesso effetto sul <b>Sistema e Device</b> .
Effetto Area	Descrizione dell'effetto che il <b>modo di guasto</b> analizzato produce sull' <b>area</b>

Funzionale (L1)	<p><b>funzionale</b> di cui il <b>componente</b> fa parte (Livello 1).</p> <p>NB: Ad uno stesso <b>Effetto Sistema e Device</b> dovrà corrispondere uno stesso <b>Effetto Area Funzionale</b>.</p>
Effetto Finale (L0)	<p>Descrizione dell'effetto che il <b>modo di guasto</b> analizzato produce sull'<b>Impianto</b> di cui il <b>componente</b> fa parte (Livello 0).</p> <p>NB: Ad uno stesso <b>Effetto Sistema e Device</b> dovrà corrispondere uno stesso <b>Effetto Finale</b>.</p>
Altri elementi interessati	<p>Elenco degli ulteriori <b>componenti</b> che risentono indirettamente del modo di guasto considerato.</p> <p>ES: Il <b>modo di guasto</b>: "rottura di un dente" del componente <b>gearbox</b> potrebbe provocare la rottura del componente <b>pala</b></p>
Frequenza	Descrizione quantitativa della frequenza del verificarsi del <b>modo di guasto</b> in analisi
Indice di frequenza	Numero identificativo della classe di appartenenza della <b>frequenza</b> del verificarsi del <b>modo di guasto</b> analizzato. Si faccia riferimento alla tabella "INDICE DI FREQUENZA".
Impatto sulla sicurezza del personale	Numero identificativo della classe di appartenenza della gravità del danno atteso al personale e/o all'impianto dovuto al verificarsi del <b>modo di guasto</b> in analisi. Si faccia riferimento alla tabella "IMPATTO SULLA SICUREZZA DEL PERSONALE" inserendo l'indice che si trova nella casella di incrocio della probabilità di accadimento e del danno. In questo caso la probabilità è da intendersi come probabilità condizionata all'essersi verificato il modo di guasto corrispondente.
Impatto sull'ambiente	Numero identificativo della classe di appartenenza della gravità del danno atteso sull'ambiente dovuto al verificarsi del <b>modo di guasto</b> in analisi. Si faccia riferimento alla tabella "IMPATTO SULL'AMBIENTE".
Impatto sull'impianto (Capex)	Numero identificativo della classe di appartenenza della gravità del danno atteso sull'impianto dovuto al verificarsi del <b>modo di guasto</b> in analisi. Si faccia riferimento alla tabella "IMPATTO SULL'IMPIANTO (CAPEX)". Per le quattro classi è opportuno definire i limiti di impatto come danno relativo al singolo evento sugli asset coinvolti (ad esempio costi di sostituzione, riparazione o ricondizionamento)
Impatto sul ciclo di produzione geotermico (Opex)	Numero identificativo della classe di appartenenza della gravità del danno atteso sul ciclo di produzione geotermico dovuto al verificarsi del <b>modo di guasto</b> in analisi. Si faccia riferimento alla tabella "IMPATTO SUL CICLO DI PRODUZIONE GEOTERMICO (OPEX)". Per le quattro classi è opportuno definire i limiti di impatto come danno sulla produzione geotermica riferito all'interruzione attesa della funzione affetta dal modo di guasto. Per interruzione attesa si considera il tempo complessivo atteso di ripristino in correttiva, tenuto conto anche dei tempi gestionali e di approvvigionamento materiali.
Quantitativo Capex [€/evento]	Descrizione quantitativa economica della gravità del danno atteso sull'impianto dovuto al verificarsi del <b>modo di guasto</b> in analisi. Si intende il danno relativo al singolo evento sugli asset coinvolti (ad esempio costi di sostituzione, riparazione o ricondizionamento)
Quantitativo Opex2	Descrizione quantitativa economica della gravità del danno atteso sul ciclo di

[€/evento]	produzione dovuto al verificarsi del <b>modo di guasto</b> in analisi. Si intende il danno sulla produzione geotermica riferito all'interruzione attesa della funzione affetta dal modo di guasto
Down time di manutenzione preventiva	Numero identificativo della classe del tempo necessario per l'intervento manutentivo di prevenzione teso ad intercettare il <b>modo di guasto</b> in analisi. Si faccia riferimento alla tabella "DOWNTIME MANUTENZIONE PREVENTIVA".
Generalizzabilità dell'analisi	Indice qualitativo del grado di generalizzazione dell'analisi del particolare modo di guasto sugli altri impianti geotermici ENEL. Si faccia riferimento alla tabella "GENERALIZZABILITÀ DELL'ANALISI".
Sensori di diagnostica adottata	Elenco dei sensori di diagnostica ad oggi utilizzati, utili all'individuazione del <b>modo di guasto</b> in analisi
Indice di Detectability	Numero identificativo della classe di appartenenza della capacità di indentificare il <b>modo di guasto</b> in analisi. Si faccia riferimento alla tabella "INDICE DI DETECTABILITY"
Modalità di rilevamento guasto	Descrizione dell'attuale modo con cui viene individuato il <b>modo di guasto</b> in analisi.
Controlli di prevenzione	Descrizione degli eventuali controlli di prevenzione ad oggi impiegati, utili alla rilevazione del <b>modo di guasto</b> in analisi.
Azioni manutentive effettuate	Descrizione delle azioni manutentive effettuate per la risoluzione del <b>modo di guasto</b> in analisi.
Migliorie di impianto consigliate	Descrizione di eventuali migliorie consigliate per l'individuazione del <b>modo di guasto</b> in analisi.  NB: Campo compilabile al termine dell'analisi FMECA
Indice di criticità	Valore numerico calcolato identificativo del livello di criticità del <b>modo di guasto</b> in analisi.  Il calcolo dell'indice è effettuato tramite il prodotto tra l'indice di "Severity" (media pesata dei campi: Impatto sulla sicurezza del personale, Impatto sull'ambiente, Impatto sull'impianto (Capex), Impatto sul ciclo di produzione geotermico (Opex); con pesi definiti tramite metodologia AHP) e l'indice di frequenza.

Possiamo notare, confrontando con altri esempi di FMECA di prodotto riscontrabili in letteratura (Bromley and Bottomley, 1994; Carlson, 2014; de Aguiar et al., 2015), come siano stati aggiunti dei campi che di norma non sono presenti. Abbiamo infatti deciso, per far fronte anche ad alcune necessità del progetto o per richieste esplicite direttamente dei responsabili di ENEL GREEN POWER, di apportare delle modifiche a quelli che sono gli standard previsti creando una tabella ad hoc.

Parallelamente a questa analisi, che è una FMECA di prodotto, abbiamo anche creato una seconda tabella per analizzare la problematica dei "guasti" dei pozzi di produzione di vapore che non sarebbero stati facilmente trattabili con la tecnica appena vista. Abbiamo quindi creato una seconda tabella FMECA, questa volta però di processo. La differenza tra le due è che la prima si concentra su come il modo di guasto si propaga risalendo verso livelli più alti, la seconda invece si preoccupa di come il modo di guasto

possa influire su tutta la linea di processo, e quindi in sostanza come va a disturbare il processo nella sua interezza e non più solo il sistema nel quale avviene.

Id	AGE (L0)	Area funzionale (L1)	Componente (L2)	Riferimento SAP PM	Funzione di processo
1.1.1.1	radicondoli 2	sistemi di lavaggio in pozzo	tubo apporto acqua di lavaggio		apporto della miscela di lavaggio al pozzo
1.2.1.1	radicondoli 2	alimentazione vapore di processo	pozzo		riformimento vapore di processo

Funzione di processo	Modo di guasto	Cause	Effetti su processo	Effetti su rete di distribuzione e impianto di produzione	Frequenza
apporto della miscela di lavaggio al pozzo	intasamento	incrostazioni	mancato lavaggio di vapore con conseguente aspirazione di vapore acido		
riformimento vapore di processo	affogamento	sovradosaggio acqua di lavaggio	aspirazione eccessiva di acqua da pozzo, danni strutturali o geotermici a pozzo che ne fanno interrompere la generazione di vapore		

Figura 3.26 tabella FMEA di processo. Sono state riportate solo le prime colonne in quanto la differenza rispetto alla tabella vista in precedenza consiste solo in queste.

Analogamente all’FMECA di prodotto si riporta la descrizione per ogni campo presente nella tabella della FMECA di processo, che come vediamo risulta in sostanza semplificata dal punto di vista dell’identificazione degli effetti del modo di guasto.

Tabella 3.2 descrizione dei campi presenti nella tabella FMECA di processo.

Campo FMECA	Descrizione
Id	Codice numerico di 4 cifre separate dal segno “.” ognuna delle quali identifica in ordine: l’ <b>Impianto</b> , l’ <b>Area funzionale</b> , il <b>componente</b> ed il <b>modo di guasto</b> considerato.
Impianto (L0)	Impianto geotermico analizzato (Livello 0)
Area Funzionale (L1)	Area funzionale dell’ <b>Impianto</b> analizzato (Livello 1)
Componente se presente (L2)	Nome dell’eventuale componente soggetto al modo di guasto analizzato (Livello 2)
Riferimento SAP PM	Codice alfanumerico SAP identificativo del <b>Area funzionale</b> analizzato
Funzione di processo	Funzione di processo interessata dal modo di guasto analizzato
Modo di guasto	Descrizione del <b>modo di guasto</b> del <b>componente</b> o della <b>Funzione di processo</b> analizzata. Ogni componente o funzione di processo può avere più modi di guasto che vengono aggiunti inserendo una nuova riga nella tabella
Cause	Descrizione di tutte le possibili cause del particolare <b>modo di guasto</b> .
Effetti su processo	Descrizione degli effetti che il <b>modo di guasto</b> analizzato produce sulla <b>funzione di processo</b>
Effetti su rete di distribuzione e impianto di produzione	Descrizione degli effetti che il <b>modo di guasto</b> analizzato produce sulla rete di distribuzione del vapore e sugli impianti di produzione elettrica connessi.



Frequenza	Descrizione quantitativa della frequenza del verificarsi del <b>modo di guasto</b> in analisi
Indice di frequenza	Numero identificativo della classe di appartenenza della <b>frequenza</b> del verificarsi del <b>modo di guasto</b> analizzato. Si faccia riferimento alla tabella "INDICE DI FREQUENZA".
Impatto sulla sicurezza del personale	Numero identificativo della classe di appartenenza della gravità del danno atteso al personale e/o all'impianto dovuto al verificarsi del <b>modo di guasto</b> in analisi. Si faccia riferimento alla tabella "IMPATTO SULLA SICUREZZA DEL PERSONALE" inserendo l'indice che si trova nella casella di incrocio della probabilità di accadimento e del danno. In questo caso la probabilità è da intendersi come probabilità condizionata all'essersi verificato il modo di guasto corrispondente.
Impatto sull'ambiente	Numero identificativo della classe di appartenenza della gravità del danno atteso sull'ambiente dovuto al verificarsi del <b>modo di guasto</b> in analisi. Si faccia riferimento alla tabella "IMPATTO SULL'AMBIENTE".
Impatto sull'impianto (Capex)	Numero identificativo della classe di appartenenza della gravità del danno atteso sull'impianto dovuto al verificarsi del <b>modo di guasto</b> in analisi. Si faccia riferimento alla tabella "IMPATTO SULL'IMPIANTO (CAPEX)". Per le quattro classi è opportuno definire i limiti di impatto come danno relativo al singolo evento sugli asset coinvolti (ad esempio costi di sostituzione, riparazione o ricondizionamento)
Impatto sul ciclo di produzione geotermico (Opex)	Numero identificativo della classe di appartenenza della gravità del danno atteso sul ciclo di produzione geotermico dovuto al verificarsi del <b>modo di guasto</b> in analisi. Si faccia riferimento alla tabella "IMPATTO SUL CICLO DI PRODUZIONE GEOTERMICO (OPEX)". Per le quattro classi è opportuno definire i limiti di impatto come danno sulla produzione geotermica riferito all'interruzione attesa della funzione affetta dal modo di guasto. Per interruzione attesa si considera il tempo complessivo atteso di ripristino in correttiva, tenuto conto anche dei tempi gestionali e di approvvigionamento materiali.
Quantitativo Capex [€/evento]	Descrizione quantitativa economica della gravità del danno atteso sull'impianto dovuto al verificarsi del <b>modo di guasto</b> in analisi. Si intende il danno relativo al singolo evento sugli asset coinvolti (ad esempio costi di sostituzione, riparazione o ricondizionamento)
Quantitativo Opex2 [€/evento]	Descrizione quantitativa economica della gravità del danno atteso sul ciclo di produzione dovuto al verificarsi del <b>modo di guasto</b> in analisi. Si intende il danno sulla produzione geotermica riferito all'interruzione attesa della funzione affetta dal modo di guasto
Down time di manutenzione preventiva	Numero identificativo della classe del tempo necessario per l'intervento manutentivo di prevenzione teso ad intercettare il <b>modo di guasto</b> in analisi. Si faccia riferimento alla tabella "DOWNTIME MANUTENZIONE PREVENTIVA".
Generalizzabilità dell'analisi	Indice qualitativo del grado di generalizzazione dell'analisi del particolare modo di guasto sugli altri impianti geotermici ENEL. Si faccia riferimento alla tabella "GENERALIZZABILITÀ DELL'ANALISI".

Sensori di diagnostica adottata	Elenco dei sensori di diagnostica ad oggi utilizzati, utili all'individuazione del <b>modo di guasto</b> in analisi
Indice di Detectability	Numero identificativo della classe di appartenenza della capacità di indentificare il <b>modo di guasto</b> in analisi. Si faccia riferimento alla tabella "INDICE DI DETECTABILITY"
Modalità di rilevamento guasto	Descrizione dell'attuale modo con cui viene individuato il <b>modo di guasto</b> in analisi.
Controlli di prevenzione	Descrizione degli eventuali controlli di prevenzione ad oggi impiegati, utili alla rilevazione del <b>modo di guasto</b> in analisi.
Azioni manutentive effettuate	Descrizione delle azioni manutentive effettuate per la risoluzione del <b>modo di guasto</b> in analisi.
Migliorie di impianto consigliate	Descrizione di eventuali migliorie consigliate per l'individuazione del <b>modo di guasto</b> in analisi.  NB: Campo compilabile al termine dell'analisi FMEA
Indice di criticità	Valore numerico calcolato identificativo del livello di criticità del <b>modo di guasto</b> in analisi.  Il calcolo dell'indice è effettuato tramite il prodotto tra l'indice di "Severity" (media pesata dei campi: Impatto sulla sicurezza del personale, Impatto sull'ambiente, Impatto sull'impianto (Capex), Impatto sul ciclo di produzione geotermico (Opex); con pesi definiti tramite metodologia AHP) e l'indice di frequenza.

L'obiettivo principale della metodologia FMECA come già anticipato è quello di identificare tra tutti i guasti possibili quelli che sono considerati più critici, ovvero quelli che per motivi legati alla frequenza o alla severità delle conseguenze è meglio evitare che si verifichino. Per costruire l'indice di criticità abbiamo bisogno quindi del contributo legato alla gravità dell'accadimento del guasto. Le tabelle (Tabella 3.3) utilizzate per dare in modo univoco un valore ad ogni indice presente nella tabella FMECA di prodotto sono state stilate a partire da un confronto con gli addetti di ENEL GREEN POWER.

**Tabella 3.3** serie di tabelle nelle quali sono riportate le scale utilizzate per la definizione degli indici. Menzione particolare per la scala relativa all'impatto sulla sicurezza del personale per la quale abbiamo tenuto conto non solo dei danni a cui gli operatori possono incorrere, ma anche della probabilità che il danno possa essere di quella entità, in modo da soppesare ad esempio un danno grave o irreversibile ma con probabilità di accadimento raro allo stesso modo di un danno lieve con alta probabilità di accadimento.

INDICE DI FREQUENZA	DESCRIZIONE
1	meno di un guasto per componente ogni 20 anni
2	almeno un guasto per componente ogni 4-20 anni
3	almeno un guasto per componente ogni 1-4 anni
4	più di un guasto per componente ogni anno

Impatto sulla sicurezza del personale

Danno		Probabilità			
		Raro	Possibile	Probabile	Certo
		1	2	3	4
Non si hanno effetti di rilievo	1	1	1	2	2
Danni temporanei alle persone	2	1	2	3	3
Danni gravi alle persone talvolta irreversibili	3	2	3	3	4
Decessi	4	2	3	4	4

IMPATTO SUL CICLO DI PRODUZIONE GEOTERMICO (OPEX)	DESCRIZIONE
1	danno inferiore ai 10.000 €
2	danno tra i 10.000 € e i 100.000 €
3	danno tra i 100.000 € e i 1.000.000 €
4	danno superiore a 1.000.000 €

DOWNTIME DI MANUTENZIONE PREVENTIVA	DESCRIZIONE
1	Non si hanno effetti di rilievo
2	inferiore ad una giornata
3	inferiore ad una settimana
4	superiore ad una settimana

IMPATTO SULL'AMBIENTE	DESCRIZIONE
1	Non si hanno effetti di rilievo
2	Danni temporanei all'ambiente
3	Danni gravi all'ambiente
4	Danni irreversibili

IMPATTO SULL'IMPIANTO (CAPEX)	DESCRIZIONE
1	danno inferiore ai 10.000 €
2	danno tra i 10.000 € e i 100.000 €
3	danno tra i 100.000 € e i 1.000.000 €
4	danno superiore a 1.000.000 €

INDICE DI DETECTABILITY	DESCRIZIONE
1	Facilmente individuabile, monitoraggio continuo
2	Individuabile da remoto

3	Ispezione visiva in loco
4	Necessaria ispezione approfondita sul componente

Un guasto può essere grave per vari motivi, come si deduce anche dalla presenza di più campi nell’FMECA. Quindi per generare un indice di Severity consistente è necessario trovare il modo di pesare ogni impatto nel modo più corretto possibile. Per ovviare a questo problema abbiamo adottato una versione personalizzata del metodo AHP (Analytic Hierarchy Process) che ci ha permesso di conciliare le diverse scale e unità di misura adottate per ogni impatto. Abbiamo in sostanza creato 2 questionari distinti (Tabella 3.4 e Tabella 3.5) per capire la percezione che hanno gli addetti ENEL GREEN POWER della gravità di ogni indice. Nel primo questionario abbiamo cercato di capire, confrontando i vari livelli 1 di ogni indice, quale fosse ritenuto il più grave; nel secondo questionario abbiamo fatto una serie di domande analoghe alle precedenti ma per il livello 4. A partire da questi risultati abbiamo potuto ricavare il peso per ogni indice che andrà a contribuire al calcolo della Severity di guasto.

Tabella 3.4 tabella dei questionari AHP per il livello 1.

livello 1			
criterio A	criterio B	Quale tra i due criteri ritieni più dannoso?	Quantifica con un valore tra 1 e 9 la differenza
Effetto raro con al massimo danni temporanei alle persone o possibile ma senza danni di rilievo	Nessun effetto di rilievo sull'ambiente		
Effetto raro con al massimo danni temporanei alle persone o possibile ma senza danni di rilievo	Impatto su ciclo di produzione geotermico (Opex) inferiore ai 10.000		
Effetto raro con al massimo danni temporanei alle persone o possibile ma senza danni di rilievo	Impatto sull'impianto (Capex) inferiore ai 10.000		
Nessun effetto di rilievo sull'ambiente	Impatto su ciclo di produzione geotermico (Opex) inferiore ai 10.000		
Nessun effetto di rilievo sull'ambiente	Impatto sull'impianto (Capex) inferiore ai 10.000		
Impatto su ciclo di produzione geotermico (Opex) inferiore ai 10.000	Impatto sull'impianto (Capex) inferiore ai 10.000		

Tabella 3.5 tabella dei questionari AHP per il livello 4.

<b>livello 4</b>			
<b>Criterio A</b>	<b>Criterio B</b>	<b>Quale tra i due criteri ritieni più importante rispettare?</b>	<b>Quantifica con un valore tra 1 e 9 la differenza di importanza</b>
Probabilità di decessi o certezza che si verificheranno almeno danni gravi o irreversibili	Impatto irreversibile sull'ambiente		
Probabilità di decessi o certezza che si verificheranno almeno danni gravi o irreversibili	Impatto su ciclo di produzione geotermico (Opex) superiore ai 1.000.000		
Probabilità di decessi o certezza che si verificheranno almeno danni gravi o irreversibili	Impatto sull'impianto (Capex) superiore ai 1.000.000		
Impatto irreversibile sull'ambiente	Impatto su ciclo di produzione geotermico (Opex) superiore ai 1.000.000		
Impatto irreversibile sull'ambiente	Impatto sull'impianto (Capex) superiore ai 1.000.000		
Impatto su ciclo di produzione geotermico (Opex) superiore ai 1.000.000	Impatto sull'impianto (Capex) superiore ai 1.000.000		

In sostanza attraverso il metodo FMECA siamo riusciti a far fronte al problema della mancanza di un database strutturato per quanto riguarda i guasti, problema che costringeva il personale a sforzi inutili di memoria o ingegno per capire come rispondere alle esigenze di intervento di impianto. A completamento della tabella strutturata gli addetti saranno in grado di capire fin da subito la procedura di risoluzione del guasto semplicemente controllando quali interventi erano storicamente stati fatti, e di compilare adeguatamente i campi di diagnostica dei guasti stessi presenti nel CMMS (Computerized Maintenance Management System, basato su SAP PM).

Ricordiamo però che l'ulteriore obiettivo, in realtà quello primario del progetto, era di selezionare i modi di guasto maggiormente meritevoli di sviluppo di eventuali nuove sensorizzazioni e di tecniche predittive.

La compilazione della struttura FMECA presentata e costruita appositamente per il caso studio non è ancora stata ultimata e ci siamo quindi concentrati sull'analisi preliminare dei componenti e dei modi di guasto ritenuti critici dal personale di impianto per lo sviluppo dei modelli prognostici.

### **3.4.2 - Analisi di letteratura delle tecniche prognostiche per i componenti critici**

Una volta inquadrato il contesto impiantistico e individuati i componenti critici del sistema ed i loro relativi modi di guasti, si possono applicare opportune metodologie di diagnostica e di prognostica per la previsione e la prevenzione di guasti che potrebbero compromettere le funzionalità del processo o della struttura.

A tal proposito, nel presente paragrafo vengono presentate le tecniche di diagnostica e prognostica presenti in letteratura, che si possono applicare su macchine rotative e motori presenti nell'impianto geotermico pilota industriali, in particolare su pompe, compressori, valvole, riduttori e motori elettrici.

#### **3.4.2.1 Diagnostica e prognostica sulle valvole**

Per quanto riguarda l'insieme delle valvole occorre ricercare soluzioni che possano essere applicate con legittimità su tipologie di impianti e sistemi aventi qualche analogia con l'impianto geotermico in esame (Gonçalves et al., 2011; Kalsi et al., 2004; McGhee et al., 1997, 2014).

Dato che le valvole sono componenti molto diffusi in quasi tutti gli ambiti, per prima cosa è necessario elencare tutti i possibili modi di guasto che queste possono avere. Supponendo di aver effettuato un'analisi FMEA/FMECA dell'impianto, tale attività può facilmente essere svolta facendo riferimento alla modulistica già compilata. In generale, i guasti alle valvole possono essere:

- di natura meccanica/strutturale
- legati al motore di movimentazione
- di origine elettrica/elettronica

I guasti più frequenti sono quelli di tipo meccanico, dovuti a condizioni di esercizio fuori progetto (ad esempio indotti dalla presenza di carichi eccessivi o urti), o a problemi di aggressività ambientale o a condizioni di lubrificazione imperfetta.

Un problema frequente quando si ha a che fare con valvole è la presenza di incrostazioni all'interno di esse. I depositi di materiale si accumulano sulla valvola limitandone lo spostamento e, dunque, impendendone l'azionamento. La valvola incrostata potrebbe non più garantire la completa apertura o

chiusura del condotto a cui fa riferimento, con conseguenze variabili a seconda del sistema in cui è inserita.

Oltre alle incrostazioni, quando si ha a che fare con valvole, si possono riscontrare problemi relativi al sistema motore-attuatore. Dato che, in alcuni casi, gli azionamenti della valvola sono rapidi e bruschi, può capitare che il sistema di azionamento venga sottoposto a eccessivi stress meccanici e, per questo, si guasti.

Non si possono escludere, inoltre, problemi di tipo elettrico. Questi ultimi sono citati in letteratura ma si verificano raramente negli impianti reali.

A causa della forte non linearità intrinseca del comportamento delle valvole, molti autori ritengono più efficace e conveniente l'utilizzo di metodi di *machine learning* (tra i quali il più largamente diffuso è l'impiego di *Artificial Neural Network* - ANN) come tecniche di diagnostica e prognostica dei guasti. Il vantaggio di questi strumenti è quello di saper riconoscere dei comportamenti non facili da identificare con i classici modelli analitici. Se si dovessero applicare metodi analitici per prevedere comportamenti complessi, sarebbe necessario disporre di approfondite conoscenze delle dinamiche di funzionamento della valvola.

Se da un lato il vantaggio delle ANN è imparare a prevedere il comportamento di un dispositivo memorizzando un set di dati storici e identificando all'interno di essi *pattern* non banali, dall'altro tale aspetto positivo rischia, se non si allena correttamente la rete, di trasformarsi in un grande svantaggio. Per esempio, se il *training set* che si usa per addestrare la rete neurale non è esauriente perché è poco numeroso e i dati che contiene non sono significativi di tutte le possibili casistiche che si possono verificare, di conseguenza la rete "imparerà" cose sbagliate e leggerà delle connessioni anche tra comportamenti puramente casuali. Viceversa, se il training set di partenza contiene troppi valori e troppo dettagliati, può capitare che la rete "impari a pappagallo" i dati e, di conseguenza, percepisca come anomalo qualsiasi altro valore differente da essi. In quest'ultimo caso si dice che la rete commette un errore di "*overfitting*" dei dati. Al fine di evitare il verificarsi di entrambe le situazioni indesiderate è buona regola effettuare sempre una pulizia dei dati prima di utilizzarli.

Secondo autori di letteratura le misure che possono essere utilizzate per fare diagnostica sulle valvole sono molte. Tipicamente si eseguono misurazioni di vibrazioni e temperatura sulle valvole e sul motore dell'attuatore; anomalie relative a tali misure percepite dai sensori, infatti, sono spesso sintomatiche di malfunzionamenti nelle movimentazioni della valvola dovuti a sovraccarichi o di altri difetti meccanici.

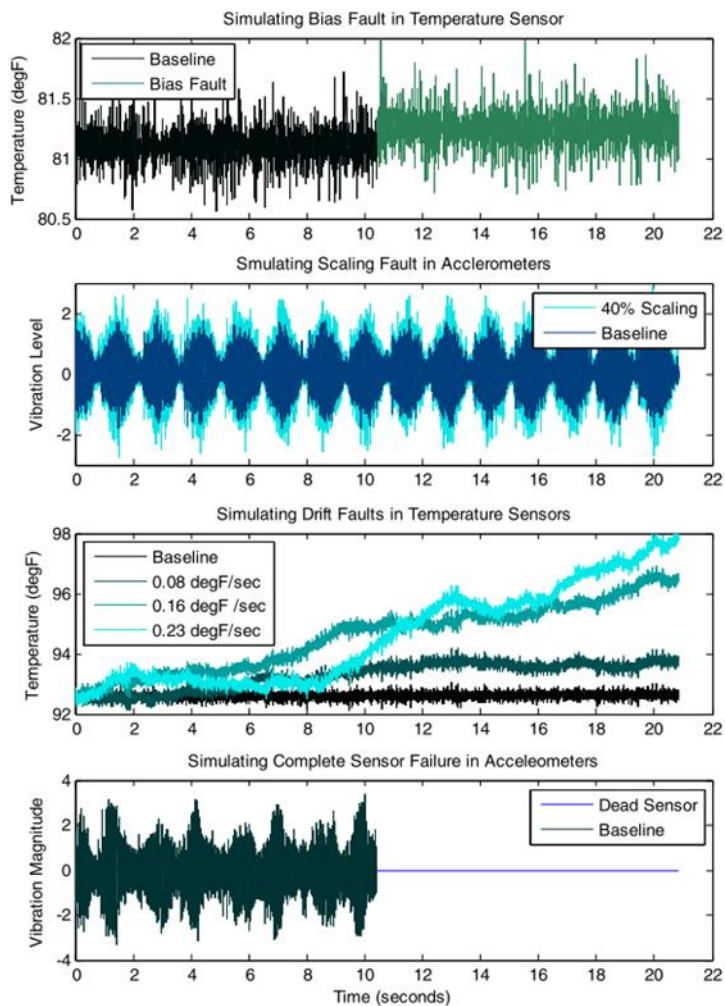


Figura 3.27 grafici riportanti esempi di derive dei segnali di temperatura o vibrazione in caso di guasto di una valvola [25].

Un'altra misura che si può rilevare per fare diagnostica dei guasti di valvole è la coppia torcente necessaria per aprire/chudere la valvola. Il monitoraggio di tale parametro (o, equivalentemente, dell'assorbimento elettrico del motore o della pressione dell'olio in caso di attuatori oleodinamici) consente di prevedere l'arrivo dei guasti; ad esempio, il metodo di McGhee et al. è basato sull'utilizzo di una ANN applicata alla misura di coppia torcente di azionamento: l'autore ritiene che analizzando queste grandezze si sia in grado di individuare correttamente fino al 90% dei guasti in cui le valvole possono incorrere durante l'esercizio.

In modo simile, il modello ideato da Gonçalves et al., si basa sull'estrazione di dati da un sensore di coppia e da un sensore di posizione applicati sulla valvola, sulla loro elaborazione mediante una Wavelet Packet Transform e sul loro apprendimento attraverso l'utilizzo di una *Self Organizing Map* (particolare rete neurale). Questo metodo è in grado non solo di identificare e prevedere l'arrivo dell'evento anomalo, ma anche di fare una prognosi sul tipo di guasto che si sta per verificare.



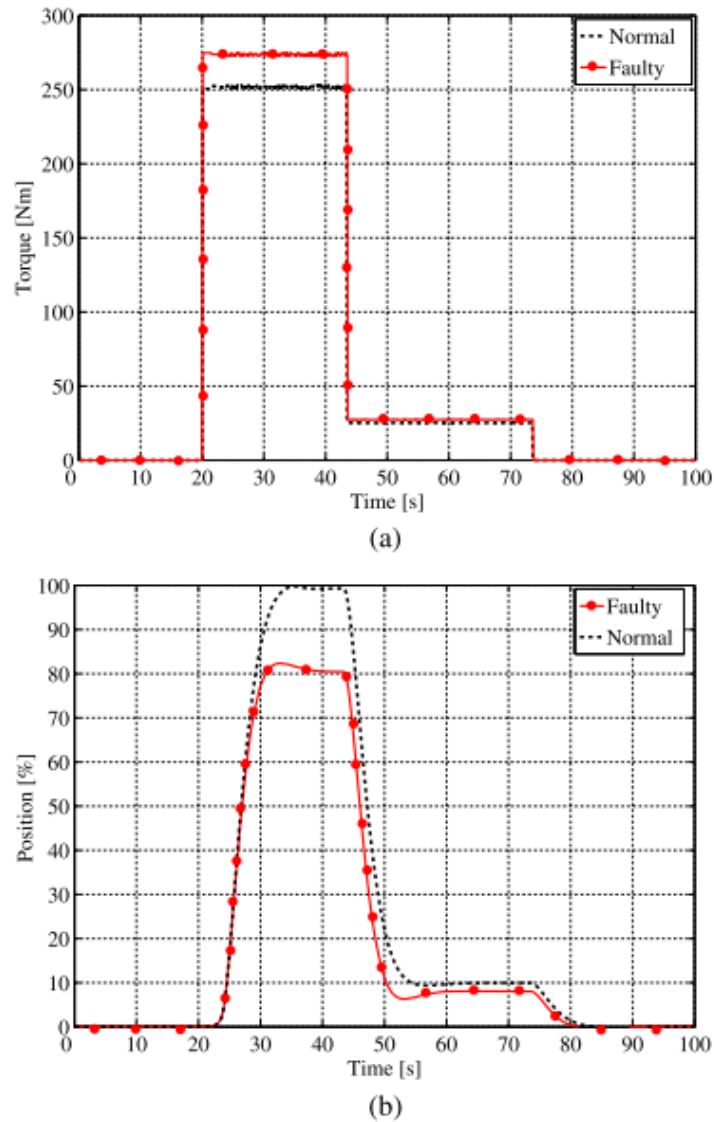


Figura 3.28 deriva dei misuratori di coppia e posizione in casi di guasti alle valvole

Esistono, inoltre, modelli di tipo matematico per la diagnostica su valvole: un esempio è quello elaborato da McGhee et al., che ha rappresentato i malfunzionamenti della valvola come riduzioni della sezione di passaggio di essa e, quindi, come un calo della pressione in uscita, mostrando come il degrado sia modellabile attraverso una relazione lineare. La valutazione della *Remaining Useful Life* (RUL) della valvola, in tal caso, risulta semplice: basta estrapolare la retta che descrive il comportamento appena citato e poi individuare il punto in cui la pressione intercetta lo zero. Il modello elaborato da M. J. McGhee semplifica il comportamento della valvola, infatti si basa sulle equazioni di conservazione della portata e di Bernoulli in un tubo cilindrico, che sono valide solo sotto ipotesi molto stringenti.

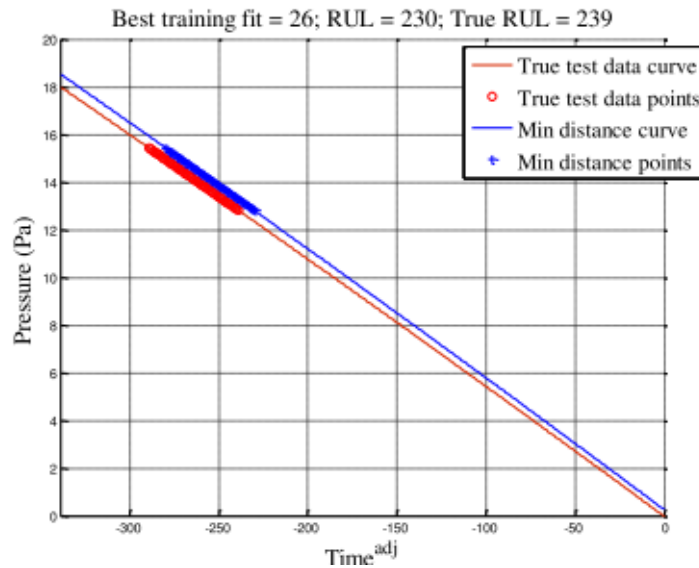


Figura 3.29 esempio di test di calcolo della RUL di una valvola

$$P_2 = P_1 + \frac{1}{2} \rho (V_1^2 - V_2^2)$$

$$A_1 V_1 = A_2 V_2$$

Infine, esistono in commercio alcune soluzioni software che consentono un'analisi anche approfondita sulla diagnostica delle valvole, uno di questo è KVAP (Kalsi et al., 2004).

### 3.4.2.2 Diagnostica e prognostica sui gearbox

Come tutte le macchine rotanti, i *gearbox* generano, durante il loro esercizio, delle vibrazioni e dei rumori che sono caratteristici degli elementi da cui sono composti. In particolare, a noi interessa capire qual è la "firma" vibrazionale che si ha quando non ci sono difetti e capire, al momento in cui si registrano modifiche del tipo di segnale vibrazionale, se è possibile associare tali anomalie ad un particolare modo di guasto (Bechhoefer and Kingsley, 2009; Li and Liang, 2012; Lin and Zuo, 2003; Omar and Gaouda, 2009; Sait and Sharaf-Eldeen, 2011).

Strumento fondamentale per fare diagnosi, in tal caso, è sicuramente almeno un sensore, comunemente un accelerometro, in grado di registrare le vibrazioni con una frequenza di campionamento adeguata (di solito dell'ordine delle decine di kHz, anche se ciò può dipendere dal tipo di *gearbox* studiato, dalla sua velocità di rotazione e dal numero di denti dei vari accoppiamenti).

Una volta in possesso del segnale vibrazionale, la più semplice e immediata analisi che si può fare è un'analisi visiva del segnale nel dominio del tempo (utile per capire se ci sono guasti che causano incongruenze di tipo impulsivo). Questa tecnica può risultare efficace soprattutto in casi in cui si vogliono identificare danni a ruote dentate o a cuscinetti, ma la sua potenzialità è limitata perché essa non ci fa capire dove è localizzato il danno (Sait and Sharaf-Eldeen, 2011).

Oltre ad un'analisi visiva, altri metodi rapidi (ma non sempre efficaci) per fare analisi diagnostica/prognostica su riduttori consistono nell'uso di indici quali *Root Mean Square*, *Peak factor* o *Crest factor* o altri strumenti statistici come la Curtosi (Sait and Sharaf-Eldeen, 2011).

Se si vogliono strumenti più potenti di diagnostica, dobbiamo spostarci dal dominio del tempo a quello della frequenza. Infatti, già controllando lo spettro del segnale vibrazionale si può essere in grado di riconoscere picchi a determinate frequenze che sono sintomo di danni specifici elementi del riduttore: ad esempio picchi a frequenze proporzionali al prodotto del numero di denti di una ruota per la sua velocità di rotazione indicano il danno a quella ruota (esistono formule analoghe per identificare le frequenze legate a difetti nei cuscinetti, ecc.). Con questi metodi si aumenta l'efficacia della diagnostica perché siamo in grado non solo di riconoscere in anticipo il funzionamento anomalo del sistema, ma anche di identificare quale elemento all'interno di esso si è danneggiato (Sait and Sharaf-Eldeen, 2011).

Altra tecnica di diagnostica è quella della *Time Synchronous Analysis* (TSA). Essa consiste nel campionare il segnale di vibrazione attraverso un trigger sincronizzato con il segnale stesso; in questo modo si creano una serie di campioni che, mediati tra loro, danno origine ad un campione finale in cui sono evidenziate tutte le tracce coerenti con il trigger scelto. Da quest'ultimo viene eliminata la parte di rumore (che sarà affievolita dal fatto che abbiamo fatto una media su più campioni) e ciò che rimane è, in sostanza, un segnale vibrazionale "pulito", privato di tutte le sue componenti non interessanti. Un segnale pulito è più facile da analizzare e manifesta in maniera più evidente eventuali comportamenti anomali del sistema (quali vibrazioni non sincrone). Alcune applicazioni di questa metodologia sono mostrate nei lavori di Li and Liang e di Bechhoefer and Kingsley.

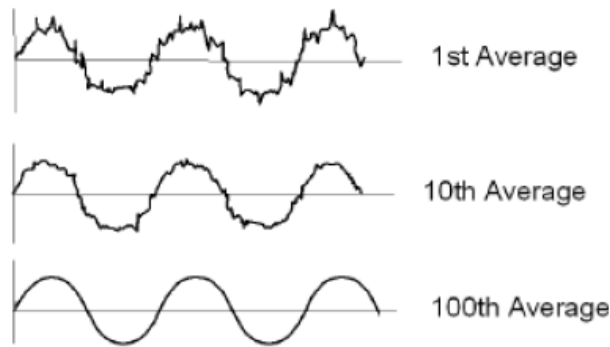


Figura 3.30 rappresentazione degli effetti del "Time Synchronous Averaging" di un segnale vibrazionale. Più campioni di segnale si estraggono mediante il trigger e più il campione finale, ottenuto come media dei campioni di partenza e privato del rumore, sarà pulito.

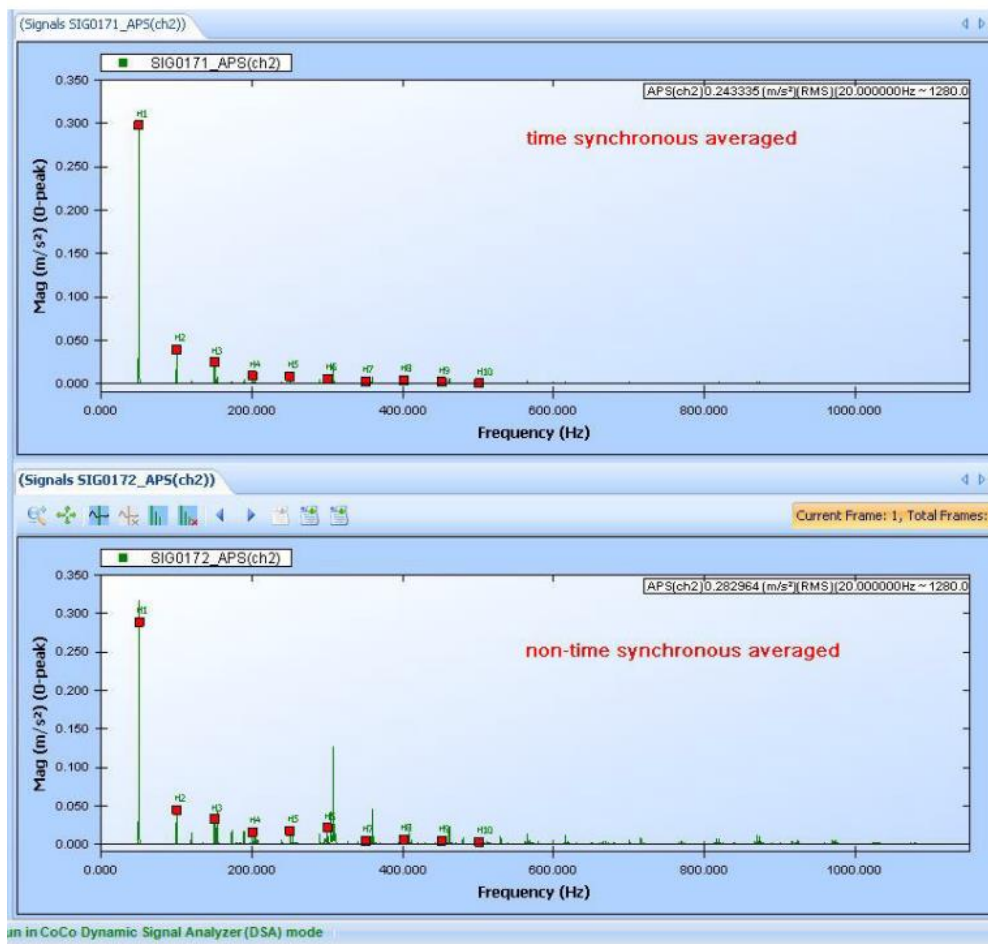


Figura 3.31 illustrazione degli effetti sul rumore di fondo della tecnica TSA tratti. Dai grafici in figura (estratti da lavori della Crystal Instruments) è possibile notare come un segnale non sincrono anomalo possa essere immediatamente riconosciuto mediante tecnica TSA.

Anche per questo tipo di diagnosi è possibile sfruttare le ANN. Le reti neurali vanno, in tal caso, applicate al segnale di vibrazione e impareranno a distinguere frequenze di vibrazione anomale che non sono

tipiche del sistema. Quando si usano ANN occorre sempre ricordare di usare un buon *training set*, che istruisca la rete senza provocare errori di apprendimento (Sait and Sharaf-Eldeen, 2011).

Un'ultima tecnica utilizzabile per riconoscere in anticipo i guasti caratteristici dei *gearbox* è quella descritta nei metodi presentati da Omar and Gaouda e da Lin and Zuo e basata sull'utilizzo della *Wavelet Transform*. Essa è efficiente ed efficace soprattutto nell'identificazione di componenti impulsive nel segnale vibrazionale che nascono al momento del danneggiamento di denti, sfere di cuscinetti o di altri elementi. Il vantaggio di questa tecnica è che essa, a differenza della analisi in frequenza dello spettro e della TSA, è applicabile anche in caso di funzionamento non a regime, cioè con segnali transitori. In figura sottostante si mostra come segnali impulsivi anomali vengano messi in evidenza dalla tecnica appena descritta:

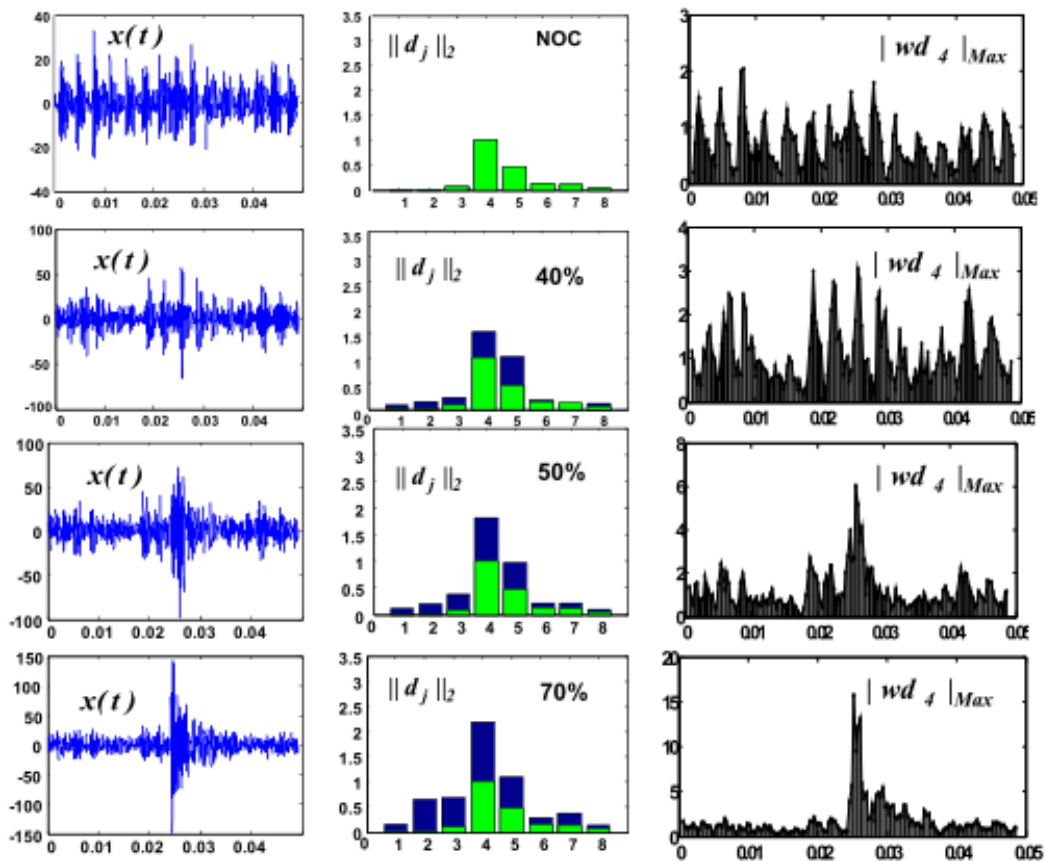


Figura 3.32 illustrazioni di come la tecnica basata sulle Wavelet Transform è in grado di rilevare i guasti. In questo caso gli autori hanno utilizzato una Discrete Wavelet Transform con Kaiser windows

### 3.4.2.3 Diagnostica e prognostica sulle pompe

Essendo anche le pompe PAE macchine rotanti, è intuitivo immaginare che buona parte delle tecniche di diagnostica/prognostica attuabili in tal caso siano analoghe a quelle presentate per i *gearbox* e che esse, in particolare, si basino sull'analisi di segnali vibrazionali (Gao et al., 2003; Hancock and Zhang, 2006; Parrondo et al., 1998; Sakthivel et al., 2010). Data la similitudine, si possono trarre per i metodi di diagnostica della pompa considerazioni simili a quelle riportate nel paragrafo precedente: ad esempio anche in queste applicazioni risulta fondamentale la scelta di sensori con adeguata frequenza di campionamento (ordine delle decine di kHz) che consentano di captare nel miglior modo possibile tutti i fenomeni di urto/sbilanciamento.

Un primo metodo utile a prevedere guasti di pompe è quello riportato da Sakthivel et al., nel quale, partendo dal segnale vibrazionale acquisito da un accelerometro montato sulla cassa della pompa, si calcolano alcune grandezze, quali la deviazione standard (SE) e il valore minimo del segnale (minimum) e, seguendo un algoritmo (C4.5), si crea un albero decisionale. Percorrendo con dei semplici passaggi i rami dell'albero si può concludere se la pompa sta lavorando in condizioni di salute oppure no e quale guasto è in atto.

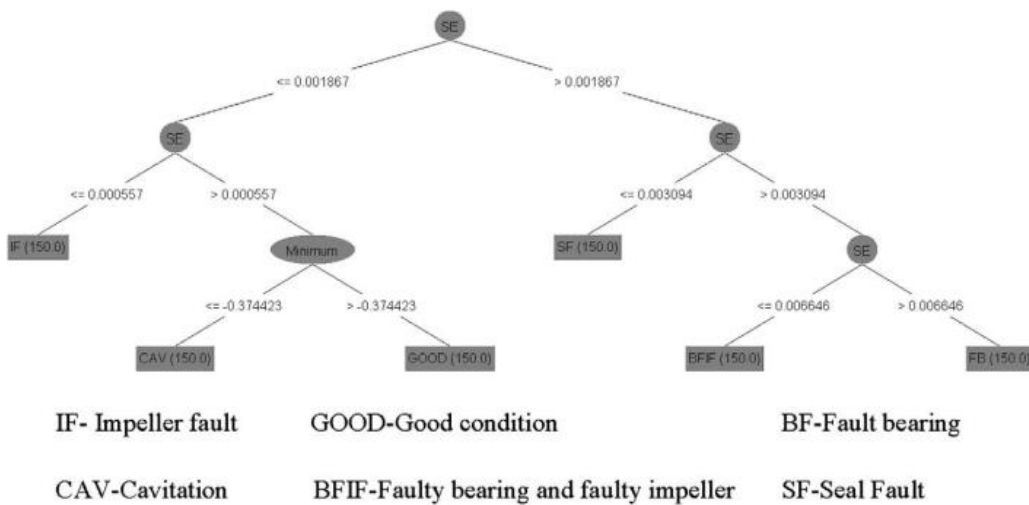


Figura 3.33 Decision Tree elaborato da Sakthivel

In alternativa, così come nel caso delle *gearbox*, anche in questo caso si possono sfruttare le *Wavelet Transform*, come illustrato da Hancock and Zhang che le hanno combinate all'uso di sistemi *Adaptive Neuro-Fuzzy Inference System* (ANFIS) per dare origine ad una particolare rete neurale basata su logiche *fuzzy*. Nelle figure di seguito si mettono a confronto studi fatti con *Wavelet Transform* su una pompa sana (a sinistra) e simulazioni fatte, invece, su una pompa difettosa (a destra).

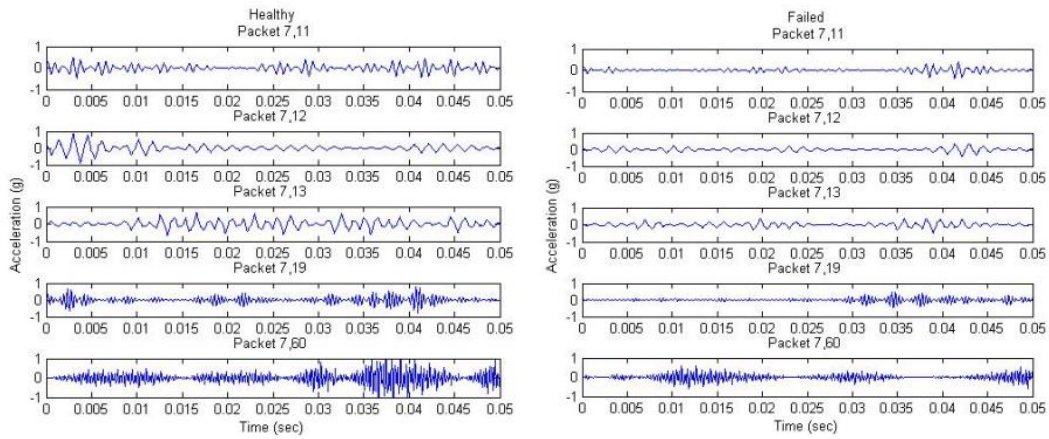


Figura 3.34 Confronto tra wavelet applicate a segnali di una pompa sana ed una difettosa

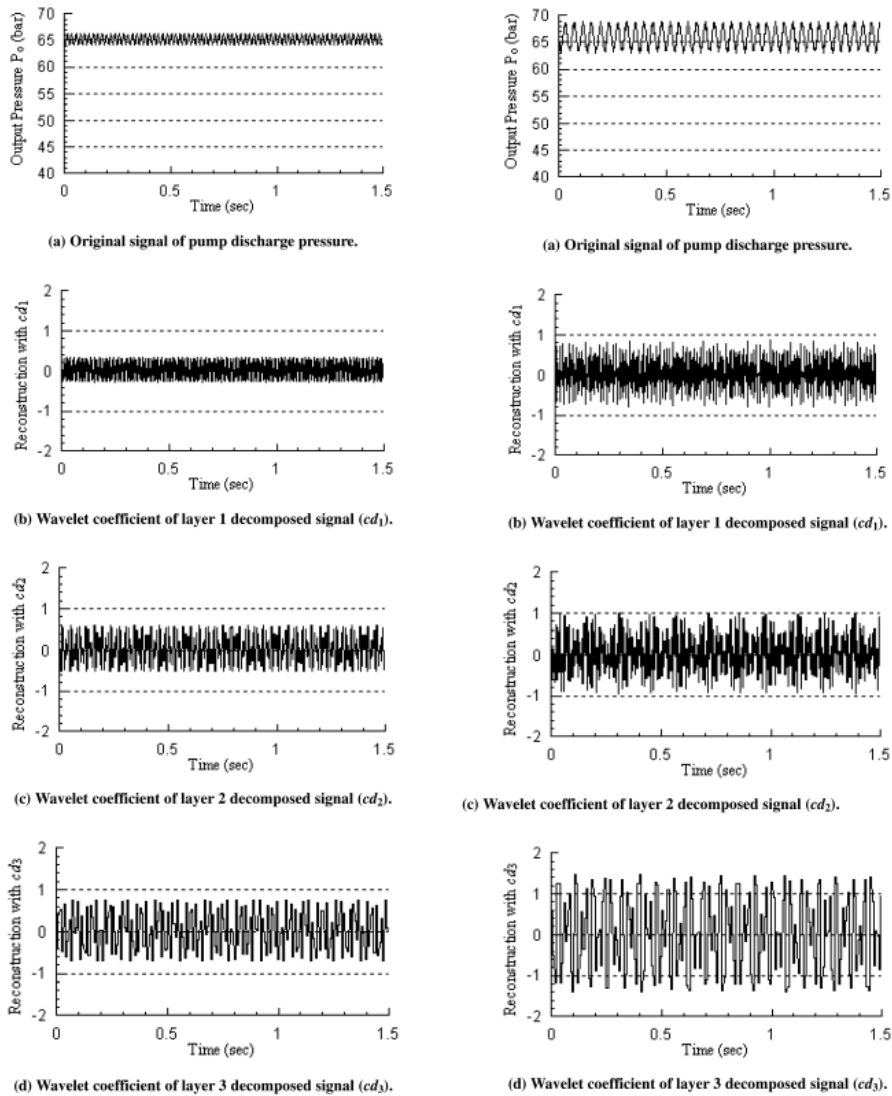


Figura 3.35 confronto tra pompa sana (sx) e pompa guasta (dx) in una simulazione con utilizzo delle wavelet

L'unico limite nell'utilizzo di modelli che necessitano sensori di vibrazione non è legato ai concetti teorici su cui essi si basano, ma piuttosto al fatto che, in caso di pompe, non è spesso possibile installare sensori di vibrazione nelle posizioni migliori per acquisire i segnali. Infatti, come riportato da Parrondo et al., posizionando gli accelerometri sulla cassa della pompa, molte volte non si ottengono vantaggi diagnostici, ma anzi si rischia che i sensori siano maggiormente sollecitati da vibrazioni di elementi esterni all'impianto stesso (ad esempio lievi scosse di terremoto) piuttosto che dalla pompa stessa.

Nel suo articolo Parrondo ci dimostra che è più efficace lo studio del segnale ottenuto da un misuratore di pressione posto all'uscita della pompa (che monitori le oscillazioni del valore di pressione di uscita), rispetto al segnale captato da sensori sulla cassa della girante. Questo nuovo metodo è conveniente perché consente, in aggiunta, di riconoscere eventuali fenomeni di cavitazione della pompa, cosa che non si poteva fare con soli accelerometri).

La tesi di Parrondo è stata sostenuta anche da Gao et al., che nel suo articolo ha applicato le *Wavelet* al segnale di pressione preso all'uscita della pompa e ha dimostrato che con esso, per alcune pompe particolari, si possono addirittura identificare le tipologie di guasto che stanno impedendo il corretto funzionamento della pompa. Poiché Gao ha svolto analisi su una turbomacchina ben specifica (pompa alternativa a pistoni) e non su vari tipi di pompa (ad esempio una pompa verticale-centrifuga o una macchina di altro genere), non si può, in realtà, generalizzare la sua teoria e concludere che uno studio dell'andamento delle pressioni in uscita alla macchina possa consentire di identificare l'esatto modo di guasto in arrivo. Inoltre, Gao non ha studiato i modi di guasto di tutti i componenti della pompa, ma si è limitato ad osservare difetti relativi a tenute o a pistoni (non a cuscinetti, né ad altri elementi), pertanto non è possibile nemmeno avere la garanzia che, con tale metodo, si possa prevedere l'arrivo di guasti relativi a tutti i componenti della macchina.

#### **3.4.2.4 Diagnostica e prognostica sui motori elettrici**

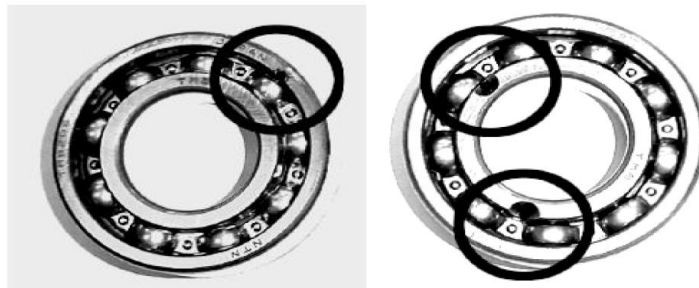
La diffusione in vari ambiti di motori elettrici ha fatto sì che fossero messe a punto sul mercato numerose soluzioni "da catalogo" anche per la diagnostica dei guasti; ad esempio sia Siemens che ABB hanno sviluppato un loro sistema personalizzato per il monitoraggio online del funzionamento dei motori. A tal proposito, in letteratura sono presenti *review* sui più diffusi metodi di diagnostica, quali, ad esempio, il metodo di Nandi and Toliyat, e quello di Siddiqui et al.,. I guasti in cui incorrono i motori elettrici sono di varia natura; i più comuni sono sicuramente i guasti ai cuscinetti (dovuti a usura, a lubrificazione assente o impropria, a difetti di montaggio o a errori di progettazione), che rappresentano circa il 50% di tutte le rotture di motori rilevate. Per prevedere questo tipo di guasto ci possiamo avvalere nuovamente di analisi vibrazionali con le quali, anche semplicemente andando a studiare i picchi nello spettro di frequenza,



siamo in grado di capire se si è danneggiata una sfera, una pista o la gabbia. Anche nel caso di guasti ai cuscinetti del motore elettrico, così come nel caso di pompe e *gearbox*, si può scegliere se:

- o eseguire un'analisi del segnale vibrazionale in frequenza
- oppure affidare la valutazione del segnale rilevato ad apposite reti neurali addestrate a distinguere il comportamento vibrazionale tipico del sistema da qualsiasi altro comportamento anomalo

Solitamente se si ha a che fare con un motore che funziona a regime si predilige l'uso di ANN, mentre se si ha un motore in transitorio si preferisce l'analisi in frequenza; le reti neurali, infatti, sono di più rapido utilizzo, ma non riescono ad apprendere bene il comportamento del sistema se esso cambia in continuo perché non è in stato stazionario.



**Figura 3.36** immagini di cuscinetti interni al motore elettrico che riportano danni alla pista esterna (a sinistra) e a quella interna (a destra)

Altri difetti tipici dei motori elettrici si riscontrano sullo statore e sulle armature; si tratta principalmente di difetti legati al cattivo isolamento della macchina (circa il 40% dei guasti al motore è imputabile a queste cause).

Quando si analizzano motori ad alto voltaggio (sopra 4000 V), i difetti legati al non perfetto isolamento si diagnosticano con metodi detti "a scarica parziale": essi rilevano eventuali impulsi di corrente ad alta frequenza che compaiono nello spettro del segnale in modo intermittente e sono sintomatici di malfunzionamento. In commercio esistono appositi strumenti, chiamati TGA-B, per eseguire analisi di questa natura. Quando, invece, si ha a che fare con motori a basso voltaggio, purtroppo non esistono metodi standardizzati per la previsione dei guasti. In tal caso è possibile diagnosticare i malfunzionamenti analizzando, ad esempio, parametri quali la componente assiale del flusso magnetico o l'impedenza del circuito. In generale, tutte le tecniche appena nominate garantiscono ottimi risultati in termini di capacità di individuazione dei guasti.

Altri difetti che affliggono i motori riguardano i rotor; tali guasti però si manifestano raramente perché gli studi di progettazione condotti negli anni hanno permesso di raggiungere risultati notevolmente migliori in termini di affidabilità.

I rotor possono essere ottenuti per fusione (di solito per motori non riparabili e fino a 3000 kW) o per fabbricazione vera e propria (per motori riparabili e a più alto voltaggio) ma, a prescindere da come essi vengano prodotti, le cause di guasto sono le stesse e corrispondono a stress termici, stress magnetici, stress derivati da difetti di fabbricazione, stress meccanici o stress legati alle condizioni ambientali. Detto ciò, difetti ai rotor si possono rilevare con un'analisi dello spettro della *Machine Line Current* definita "*Motor Current Signature Analysis (MCSA)*", oppure analizzando lo spettro in frequenza dei voltaggi indotti in avvolgimenti posizionati sul rotore.

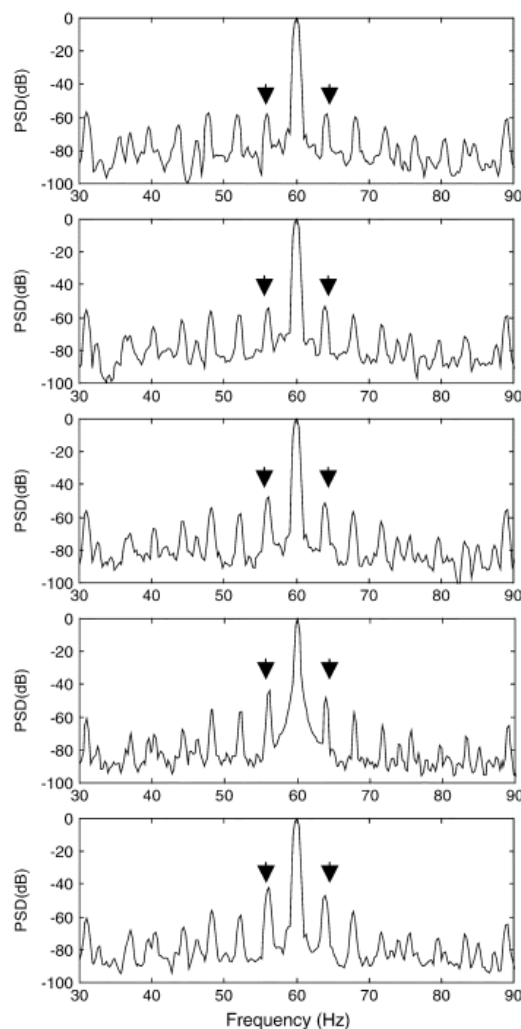


Figura 3.37 spettro di frequenza normalizzata della line current (analisi MCSA) in un motore sano (in alto) e in motori con criticità di danno al rotore crescente (immagini successive procedendo dall'alto verso il basso)

Ancora più rari, ma pur sempre interessanti sono i guasti legati a problemi di eccentricità eccessiva che, in casi estremi, portano il rotore a strisciare sullo statore. Questa tipologia di guasti può essere provocata da problemi progettuali o da disallineamenti in esercizio. Anche in questi casi la *Motor Current Signature Analysis* (MCSA) è in grado di prevedere la presenza di un eventuale danno, infatti esistono delle frequenze caratteristiche che variano in base alla tipologia di motore e che, se riscontrate, mettono in luce difetti di eccentricità.

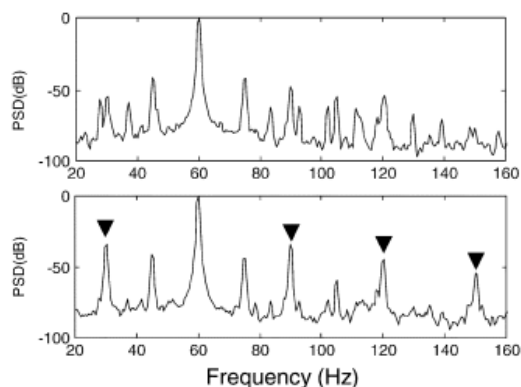


Figura 3.38 analisi dello spettro della line current normalizzata su un motore sano (in alto) e su un motore con disallineamenti (in basso) intorno alla frequenza fondamentale di funzionamento.

Anche per i motori il monitoraggio delle vibrazioni sarebbe valido, solo che a volte risulta economicamente troppo oneroso. Non è escluso nemmeno l'impiego di ANN, comunemente applicate usando come feature lo spettro della corrente dello statore. Le reti neurali nel caso di motori elettrici vengono adoperate soprattutto per motori a corrente alternata, data la non linearità tra corrente e velocità di rotazione.

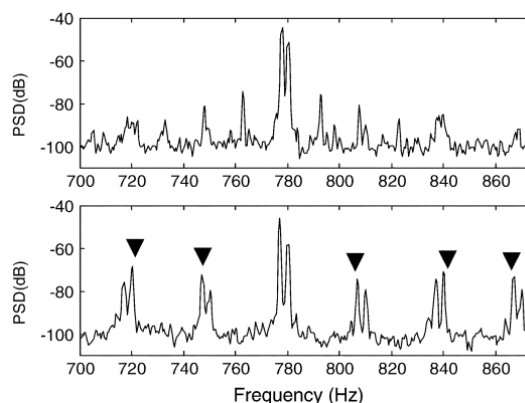


Figura 3.39 segnali di esperimenti con tecnica MCSA che dimostrano, in questo caso analizzando la corrente statorica, delle differenze che si notano tra motore sano (in alto) e motore danneggiato (in basso)

Per concludere, sembra che la tecnica MCSA sia la migliore per fare diagnostica su motori elettrici, a maggior ragione se affiancata da Fast Fourier Transform (FFT) in casi stazionari o da Wavelet Transform se ci interessano anche i transitori. Per quanto concerne la Wavelet Transform, tuttavia, occorre sottolineare che mancano in letteratura esempi e “case study” afferenti al suo utilizzo su motori elettrici. Pertanto, essendo il motore elettrico una macchina in rotazione analogamente a pompe e gearbox, si può supporre che le Wavelet possano essere convenientemente usate per fare diagnostica dei guasti, ma occorre ricordare che mancano risultati scientifici che confermino a livello applicativo la sua validità e la sua efficacia effettiva.

### 3.4.2.5 Diagnostica e prognostica sui compressori

Anche in questo caso si parla di macchine con componenti rotanti, di conseguenza le principali tecniche diagnostiche e prognostiche sono simili ai gearbox e pompe trattate fino ad ora. In particolare, le principali tecniche diagnostiche e prognostiche utilizzate sono le seguenti (Pusey, 2007):

Analisi del segnale di vibrazione. Queste tipologie di tecniche, che fanno parte delle tecniche diagnostiche non distruttive, sono utilizzate da diversi decenni e sono tra le tecniche più utilizzate per la diagnostica di componenti rotanti. In particolare, per ottenere un sistema diagnostico e prognostico efficace utilizzando questo tipo di analisi, è necessario concentrarsi su 3 differenti aree (Eshleman, 1990); queste sono: la conoscenza dei meccanismi di guasto e operativi, la modifica del segnale di vibrazione acquisito e l'analisi dei segnali sia in frequenza (FRF Wavelet) che nel tempo (Crest Factor, Curtosi ecc.).

Analisi degli oli. L'analisi dell'olio del compressore è una tecnica che si è dimostrata molto efficace nella diagnostica e prevenzione di guasti meccanici. Una delle più utilizzate è l'analisi spettrografica dell'olio per quantificare la presenza di metalli all'interno del lubrificante; ad una maggiore presenza di metalli corrisponderà una maggior usura dei componenti rotanti (Pusey, 2007).

Motor Current Signature Analysis. Anche questa è una delle tecniche non invasive di diagnostica e prognostica di eventi di guasto su turbomacchinari. Tale tecnica permette di individuare sia problemi di natura meccanica che elettrica per i macchinari movimentati da motori elettrici. Il funzionamento di base della MCSA consiste nell'assimilare che il motore elettrico accoppiato ad un carico meccanico agisce come un efficiente trasduttore in continuo (il motore può essere sia AC che DC). Il motore di conseguenza sente le variazioni di carico della parte meccanica e le converte in variazioni di corrente che sono trasmesse nei cablaggi del motore. Analizzando le variazioni di corrente, con le tipiche tecniche di analisi dei segnali, è quindi possibile monitorare le condizioni della macchina e prognosticare eventuali guasti e/o malfunzionamenti (Castleberry and Nowlin, 1990).

A queste tecniche generali si uniscono anche le analisi sui cuscinetti delle turbomacchine (temperature, emissioni acustiche, ultrasuoni ecc.), monitoraggio delle performance (rendimento, rapporto di compressione ecc.), termografia infrarossa.

Inoltre, per questo tipo di macchinari, risulta di fondamentale importanza la diagnostica e prognostica di eventi di pompaggio, che, una volta insorto, induce forti vibrazioni alla macchina, causando danni meccanici anche catastrofici.

Il funzionamento dei compressori è espresso per via grafica dalla cosiddetta curva caratteristica, che mostra l'andamento del rapporto di compressione (oppure il salto di pressione) realizzato dal compressore, in funzione della portata in massa. In generale, in un compressore, il rapporto di compressione aumenta al diminuire della portata in massa, finché si raggiunge un punto di funzionamento in cui il rapporto di compressione è massimo ed una ulteriore riduzione di portata comporta una più o meno brusca variazione delle prestazioni e della struttura del flusso: il compressore entra in stallo o in pompaggio. Tale punto di instabilità è noto come surge point (che ne consegue stallo o pompaggio), ed il luogo di tali punti, per diversi numeri di giri, come surge line. Si distinguono (specialmente per i compressori assiali) tre principali modalità attraverso le quali l'instabilità di funzionamento si manifesta: (Brown, 1997; Day, 2016; De Jager, 1995; Gravidahl and Egeland, 1999)

1. Stallo graduale,
2. Stallo improvviso,
3. Pompaggio.

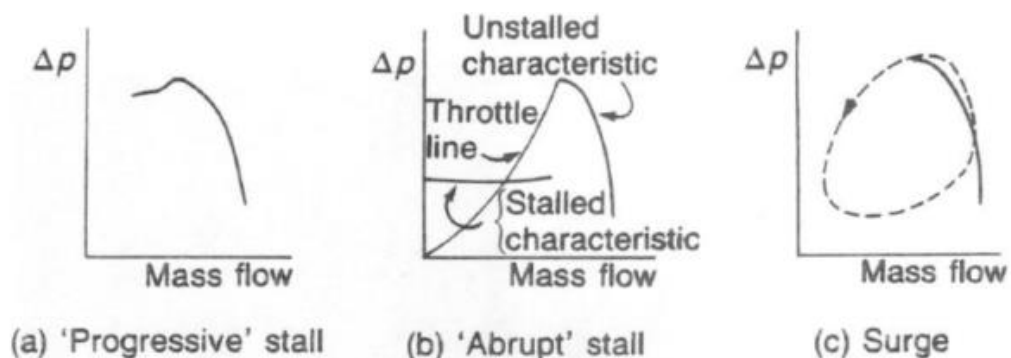


Figura 3.40 caratteristica di funzionamento del compressore in stallo graduale, stallo improvviso e pompaggio

In caso di stallo graduale, si verifica un calo moderato delle prestazioni, e la presenza dello stallo è spesso indicata solo da una variazione della rumorosità o è rilevabile tramite strumentazione ad alta frequenza all'interno della macchina.

In caso di stallo improvviso, si verifica un brusco crollo del rapporto di compressione  $\beta$  e della portata: il punto di funzionamento si sposta rapidamente seguendo la throttle line, posizionandosi intorno ad un  $\beta$  di molto inferiore all'originale.

In entrambi i casi, il flusso non rimane assialsimmetrico e presenta una struttura circonferenzialmente disuniforme, che ruota lungo l'annulus: per tale motivo si parla di stallo rotante. L'annulus contiene regioni di flusso in stallo (celle di stallo) e regioni di flusso regolare. Lo stallo rotante, pertanto, è un meccanismo che consente al compressore di adattarsi a smaltire una portata più bassa della minima associata al funzionamento stabile, con il flusso che si distribuisce in modo disuniforme. Dunque, in condizioni di stallo rotante, la portata in massa complessiva rimane costante nel tempo. Una spiegazione fisica del fenomeno è data dal modello di Emmons, secondo cui lo stallo si manifesta interessando inizialmente un solo vano, dando poi luogo alla formazione di un'ampia cella di stallo che copre più vani; l'effetto di bloccaggio associato allo stallo del primo vano comporta una variazione di incidenza sulle pale contigue (riduzione su un lato e aumento sull'altro). Le pale che subiscono un aumento di incidenza tendono a stallare.

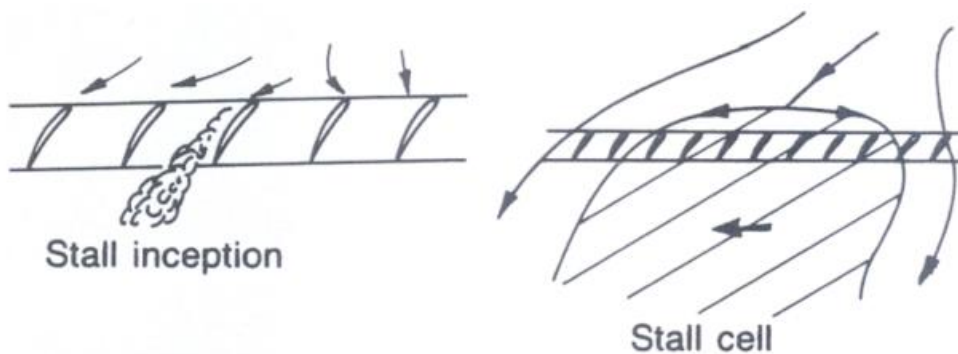


Figura 3.41 nascita e sviluppo dello stallo rotante

Il pompaggio si verifica nel momento in cui il compressore perde la capacità di produrre una sufficiente pressione, tale da bilanciare la pressione presente allo scarico del sistema di compressione. Questa instabilità è caratterizzata prevalentemente da oscillazioni di pressione e di portata. La portata in massa complessiva è variabile nel tempo, a differenza di quanto succede in caso di stallo, e l'intero compressore oscilla periodicamente fra la condizione di funzionamento regolare e la condizione di stallo.

Dal punto di vista fisico, il pompaggio si manifesta quando il compressore sta lavorando in un punto della curva caratteristica a pendenza positiva: in queste condizioni una perturbazione di portata provoca una perturbazione di pressione con medesimo "segno", che, a sua volta, alimenta la perturbazione di portata

stessa, portando quindi all'instabilità. Per una trattazione più rigorosa, si può considerare la questione in termini energetici, distinguendo due casi:

1. Il punto di funzionamento del compressore è in una zona a pendenza positiva della curva caratteristica, quindi si hanno perturbazioni di pressione e portata in fase. In tali condizioni si è in una regione instabile, in quanto l'energia di pompaggio  $\delta m \times \delta (\Delta p_0)$ , che alimenta il moto ondulatorio, è positiva su un ciclo e l'energia meccanica è superiore a quella del caso stazionario, per cui le perturbazioni si amplificano e portano al pompaggio.
2. Il punto di funzionamento del compressore è in una zona a pendenza negativa della curva caratteristica, quindi si hanno perturbazioni di pressione e portata in controfase. In tali condizioni si è in una regione stabile, in quanto l'energia di pompaggio  $\delta m \times \delta (\Delta p_0)$  è negativa su un ciclo e l'energia meccanica è inferiore a quella del caso stazionario, per cui le perturbazioni si smorzano e si riottiene il punto di partenza.

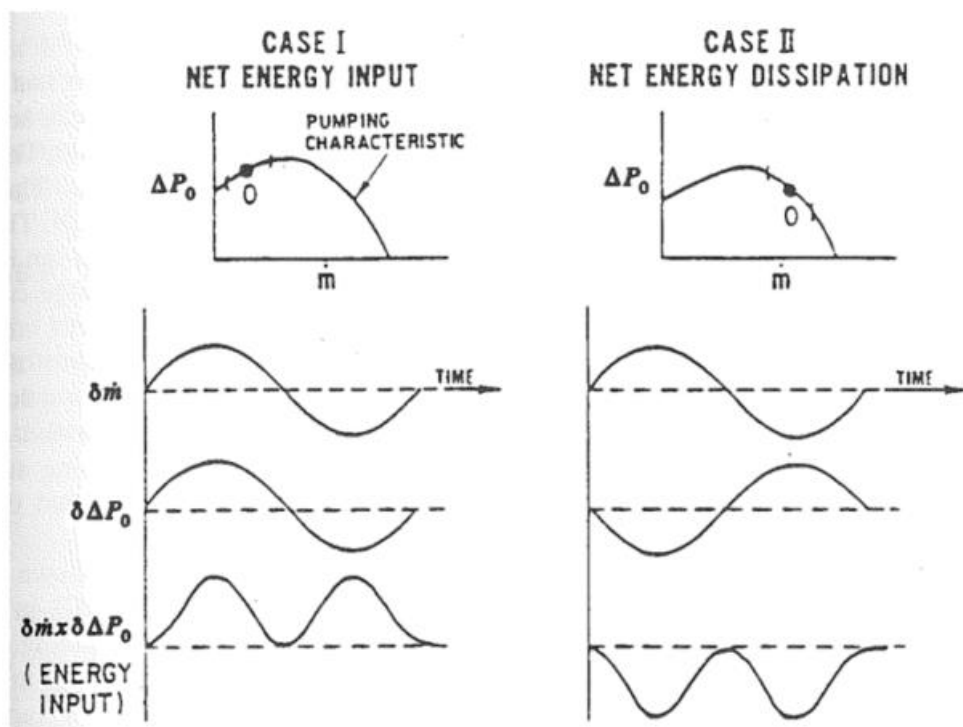


Figura 3.42 funzionamento stabile e instabile in termini energetici

Tale processo può manifestarsi in modo talmente violento da comportare l'inversione completa del flusso durante l'oscillazione (deep surge), oppure in modo lieve (mild surge), con il punto di funzionamento che oscilla in un intorno della condizione di massimo  $\beta$  con una variazione della rumorosità. Se si è in condizioni di mild surge, la portata da monte a valle del compressore è drasticamente ridotta se non addirittura annullata. In caso di deep surge, le elevate oscillazioni di pressione inducono delle vibrazioni

che possono danneggiare, ed addirittura portare alla rottura a fatica, i componenti meccanici del compressore, come la girante e i cuscinetti. Un altro effetto tangibile è l'incremento di temperatura all'ingresso, a causa dell'oscillazione di portata, che concorre insieme agli altri sintomi del pompaggio al calo dell'efficienza del compressore e, in definitiva, all'instabilità di funzionamento dell'intero sistema di compressione. Queste considerazioni evidenziano quanto sia importante evitare l'occorrenza di tale fenomeno.

Dal momento che la curva caratteristica del compressore ad un determinato numero di giri è concava, il limite di stabilità (surge point) corrisponde al punto di massimo della curva. Tuttavia, la determinazione del surge point e, di conseguenza, della surge line, è affetta da incertezza, in quanto questa dipende da vari fattori, relativi prevalentemente alle condizioni ambientali e al danneggiamento meccanico dei componenti. Spesso i produttori di compressori forniscono le curve caratteristiche, con annessi i limiti di stabilità sotto determinate condizioni standard, ma data la variabilità delle condizioni ambientali e del livello di stress della macchina, queste sono poco affidabili. Dal punto di vista operativo, è invece auspicabile ottenere una mappa del compressore che si adatti alle reali condizioni di funzionamento. In questo senso è possibile aggiungere dimensioni al problema della determinazione della curva caratteristica di funzionamento utilizzando un modello Machine Learning di tipo SVM (Support Vector Machine), con il compito di separare il funzionamento a regime da quello di pompaggio in funzione di tutte le variabili che si ritengono responsabili dell'insorgere del fenomeno di instabilità del compressore. In questo modo, monitorando in real-time la distanza del punto di funzionamento del compressore indicata da tutte le feature selezionate, dall'ipersuperficie di separazione tra le due classi di funzionamento del compressore, creata dal modello SVM, è possibile controllare, diagnosticare e prognosticare eventuali funzionamenti anomali (Wu and Li, 2012).

### 3.5 - Riflessione sui Big data

L'inconsistenza dei dati di guasto riscontrata durante lo svolgimento del progetto ha ridotto drasticamente le possibilità di diagnostica e prognostica che erano stati prefissati in fase di analisi dei componenti critici di impianto. Questa problematica porta alla luce una situazione che ad oggi è stata rilevata anche in altre realtà: la tanto decantata presenza di database ricchi di dati, più comunemente noti come Big Data, sono spesso una sola raccolta di misure fini a sé stesse, ovvero che non hanno particolare impiego se non quello per cui erano stati richiesti al momento della progettazione.

Nel nostro caso la grande mole di dati raccolta al centro di controllo, ritenuta fonte nascosta di informazioni in attesa solo di essere svelate, è risultata in effetti utile solo ai fini del controllo già in atto



all'interno dell'impianto. Ad esempio, tutti i sensori di vibrazione, che avrebbero avuto enormi potenzialità ai fini diagnostici ed eventualmente prognostici, si sono rivelati adatti solo al controllo su superamento di una soglia attuato anche precedentemente all'inizio del progetto SmartGeo.

Ulteriore pecca dei dati di cui disponevamo è quella della quasi totale assenza di eventi di guasto che possano validare un qualsiasi modello di diagnostica o prognostica. Infatti, per sviluppare questo tipo di modelli è innanzitutto necessario conoscere tutti i regimi di funzionamento dei macchinari, ovvero capire come un sistema si comporta sia in stato di salute ottimo, che con produttività ridotta o con guasto incipiente. Se non abbiamo abbastanza informazioni per capire quello che succede quando si ha un guasto non è possibile creare un modello che sia totalmente validabile ai fini diagnostici e prognostici.

Prendiamo ad esempio tutte le tecniche basate sul machine learning: questi metodi si fondano sul fatto di avere un numero di dati sufficienti su cui poter fare training. Chiaramente se parliamo di big data il quantitativo di dati dovrebbe essere più che sufficiente. Invece quello che spesso accade nella nostra applicazione è che la distribuzione dei dati è sbilanciata verso quelli relativi al funzionamento "sano", e i dati di comportamento riguardanti la presenza di guasti è pressoché assente. In altre parole, non è adeguatamente coperto l'intero dominio degli stati di funzionamento per poter addestrare un algoritmo, ad esempio, di tipo classificatorio. Di conseguenza saremmo perfettamente in grado con una ANN di identificare le zone di funzionamento a regime ma quando ci troviamo di fronte ad anomalie (quando accade) siamo solo in grado di dire che c'è qualcosa che non va nelle misure, in molti casi derivante da guasti di componenti esterni a quello in analisi, senza permetterci di rispondere ad una domanda fondamentale nella diagnostica: "cosa non funziona e perché?". Chiaramente non sapendo identificare cosa si guasta ipotizzare una soluzione che sia in grado anche di fare prognostica è pura utopia.

Un teorema ritenuto fondamentale per la definizione dei Big Data è l'HACE Theorem:

"Big Data starts with large-volume, Heterogeneous, Autonomous sources with distributed and decentralized control, and seeks to explore Complex and Evolving relationships among data." (Wu et al., 2013)

Questo teorema afferma che per parlare di Big Data ci sia bisogno come prima cosa di eterogeneità, caratteristica assente nei nostri dati.

Questo è il limite che ci siamo ritrovati ad affrontare nel corso di questo progetto, limite che chiaramente ci presupponiamo di superare suggerendo quale sensoristica debba essere implementata per gli scopi che si era prefissato il progetto SmartGeo.

In conseguenza di quanto appena detto, è stato possibile analizzare a fondo il fenomeno del pompaggio del compressore centrifugo di impianto, unico componente di cui disponevamo sufficienti dati in entrambe le classi di funzionamento (“regime” e “anomalia”), mentre per gli altri componenti critici di impianto è stata proposta e validata una metodologia avanzata che consente sia il controllo delle prestazioni dei macchinari, che la stima della sua presunta Remaining Useful Life (RUL) senza però potersi spingere oltre costruendo e validando modelli diagnostici e/o prognostici.

## 3.6 - Sensorizzazione dei componenti di interesse

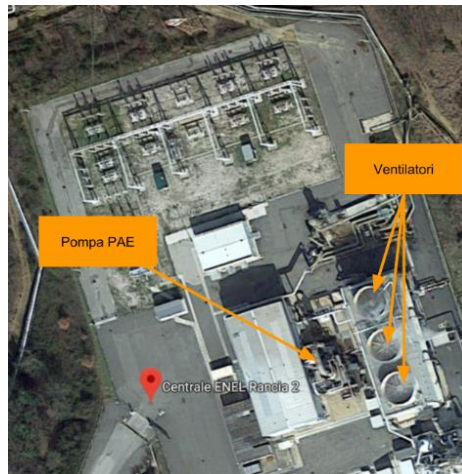
Con riferimento alla metodologia descritta nel capitolo 2.1 - ed in particolare alla Figura 2.1 a valle dell’analisi approfondita dei modi di guasto necessaria ad individuare i modi di guasto critici di impianto ed individuarne le variabili spia si passa alla fase di sensorizzazione dei componenti critici individuati andando ad installare quei sensori necessari allo sviluppo di modelli diagnostici e prognostici.

All’interno del caso studio si è occupato di questa fase il partner di progetto ISE s.r.l. che si è concentrato nell’installazione non solo dei sensori ma anche degli apparati ausiliari ai PLC di impianto necessari alla preelaborazione ed analisi dei dati raw al livello 1 del DCS di impianto.

L’installazione di tali sistemi è necessaria quando si ha a che fare con modelli hard o soft realtime che necessitano di una preelaborazione ed analisi dei dati raw per estrarne il contenuto informativo necessario ai modelli diagnostici e prognostici sviluppati.

I componenti analizzati, output della fase di analisi delle criticità sono i seguenti:

- Pompa PAE di estrazione della fase liquida dal condensatore;
- Gearbox dei ventilatori delle torri di evaporazione utilizzate per raffreddare la fase liquida estratta;



**Figura 3.43 Geolocalizzazione della Pompa PAE e dei Ventilatori all'interno dell'impianto pilota**

Dopo l'analisi dell'impianto è stata definita e progettata la catena di misura utile alla rilevazione delle grandezze fisiche caratterizzanti le utenze d'interesse:

- Scelta idonea dei trasduttori per il tipo di applicazione
- Progettazione hardware del sistema di acquisizione e trasmissione dati

In questa fase, inoltre, sono state messe a disposizione dal partenariato una serie di tecnologie e soluzioni per le quali le aziende coinvolte disponevano di un know-how preesistente da sfruttare e integrare con la specificità del progetto e in sinergia con le competenze delle realtà operanti nel progetto stesso. Questo ha comportato una fase di studio e di analisi condivisa, nella quale sono state studiate, testate e valutate, una serie di soluzioni tecnologiche utili all'implementazione della rete di monitoraggio.

I sistemi analizzati sono stati i seguenti:

- G&J M-System Mk II
- Rockwell– Allen-Bradly Dynamix 1444
- Piattaforme Dewesoft.

I sistemi selezionati hanno seguito una logica di scalabilità e applicabilità. In particolare, i sistemi sono stati suddivisi in tre diverse categorie in funzione dell'importanza e delle tempistiche di implementazione nel contesto del progetto. In particolare, si distinguono sistemi di categoria 1 da implementare all'interno del progetto come primo prototipo pilota; sistemi di categoria 2 da installare in parallelo al prototipo pilota una volta messa a punto l'architettura e i setup di misura del primo prototipo; sistemi di categoria 3 ritenuti d'interesse ma non prioritari e/o caratterizzati dalla scarsa fattibilità di messa in opera nel contesto del progetto.

- Sistemi di categoria 1:
  - Piattaforme Dewesoft;
  - Rockwell – Allen-Bradly Dynamix 1444;
- Sistemi di categoria 2:
  - G&J M-System Mk II;
- Sistemi di categoria 3
  - Piattaforme National Instrument;

Sulla base delle caratteristiche proprie delle utenze da monitorare e dei risultati dell'analisi FMECA, si descrivono di seguito le scelte progettuali relative alla catena di misura individuata per la Pompa PAE e i Ventilatori delle torri evaporative. In generale la catena di misura si compone dei seguenti elementi:

- Trasduttori. Questi sono stati scelti in base alle caratteristiche proprie e di funzionamento delle utenze d'interesse;
- Cablaggio analogico trasduttori. Questa tipologia di cablaggio è caratterizzata dallo studio del posizionamento dei sensori e delle relative condizioni ambientali al contorno;
- Schede di acquisizione. L'elemento deve rispondere a specifiche caratteristiche in risposta alla tipologia di sensori installati, alle informazioni che si vogliono estrarre da tali sensori e a peculiarità relative alla tipologia di installazione su campo scelta;
- Cablaggio tra schede di acquisizione e PC. Questa tipologia di cablaggio è caratterizzata dalla possibilità di raggiungere la sala controllo superando criticità legate alla velocità di trasmissione, lunghezza del mezzo trasmissivo, robustezza e affidabilità della soluzione;
- PC. All'interno della sala controllo si andranno ad installare una serie di elaboratori in grado di gestire i dati provenienti dalle schede di acquisizione con software dedicati e opportunamente configurati. Questi, inoltre, dovranno tener conto dell'interfacciamento con i sistemi già presenti nella centrale pilota.
- Di seguito si andranno a descrivere le scelte intraprese e le soluzioni individuate sulla base di quanto precedentemente descritto.

Nella figura seguente si riporta lo schema generale della rete di monitoraggio. In evidenza il posizionamento dei diversi quadri elettrici e dei cablaggi. In particolare, si descrivono:

- Quadro ventilatori e Quadro Pompa PAE. Sono i principali quadri elettrici nei quali vengono raccolti, dai sistemi di acquisizione dati, i segnali analogici provenienti dai trasduttori;
- Sala MCC e Quadro 220AC. Postazioni per la fornitura di energia elettrica;

- Quadro Bently. Quadro per la gestione e lo smistamento dei sensori di prossimità (probe) installati sulla Pompa PAE (Bently Nevada 3300XL);
- Sala Controllo. Centro di raccolta dati nel quale verranno installati i PC precedentemente descritti.

I collegamenti in viola indicano il collegamento in fibra ottica tra i quadri di misura e la sala di controllo. I collegamenti in verde indicano la rete di alimentazione degli stessi quadri.



Figura 3.44 Posizionamento quadri per punti di misura

In merito alla sensoristica utile al monitoraggio delle utenze, si è scelto di equipaggiare la Pompa PAE con i seguenti trasduttori:

- 2 Sensori assoluti radiali su motore (Cemb TA18S o equivalente Bently Nevada);
- 2 Sensori assoluti corpo pompa (Cemb TA18S o equivalente Bently Nevada);
- 2 probe Bently Nevada 3300XL 8mm;
- 1 TA per corrente motore;

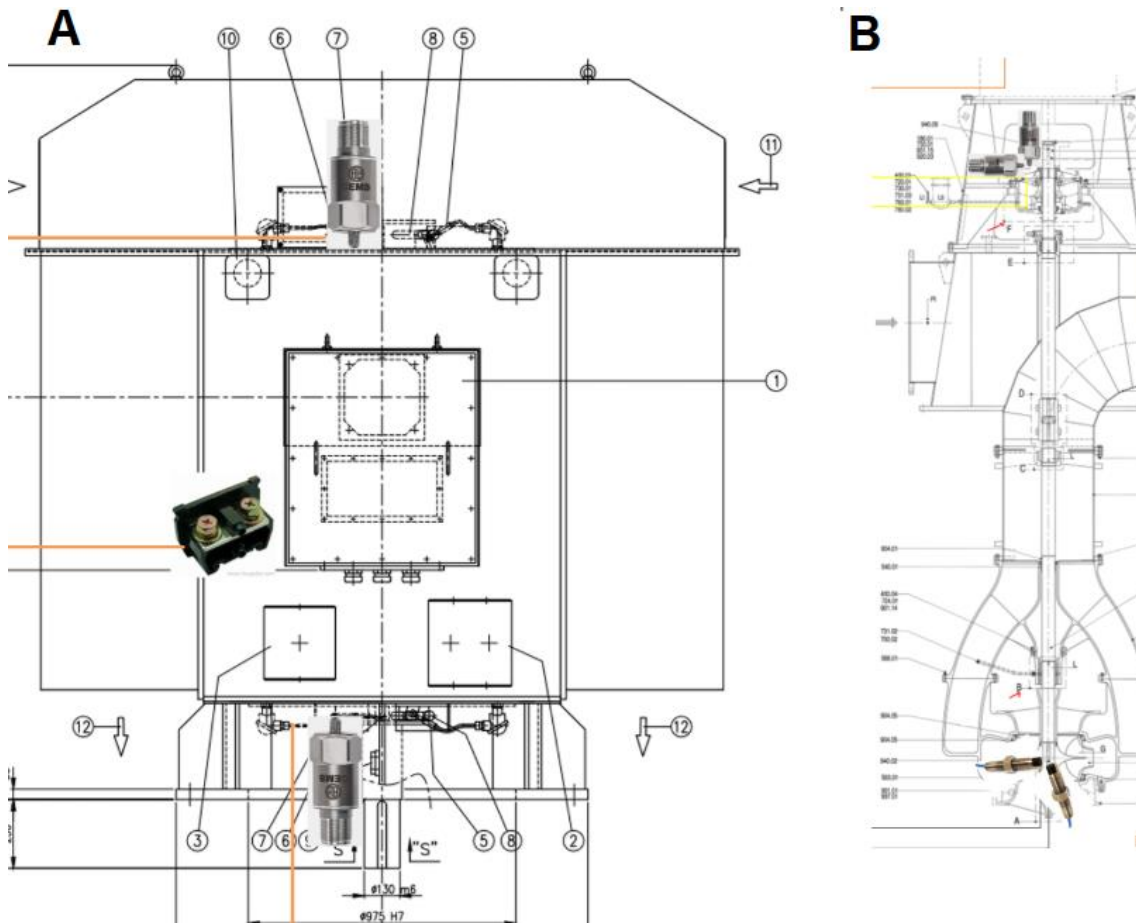


Figura 3.45 Esempio installazione trasduttori su Pompa PAE. (A) Motore, (B) Corpo pompa

Per ciascuno dei tre ventilatori è stata prevista la seguente installazione:

- 2 Sensori IEPE per motore (Cemb TA18S 100mV/g)
- 1 Sensore albero veloce del riduttore (Cemb TA18S 100mV/g)
- 1 Sensore albero lento del riduttore (Cemb TA18S 500mV/g)

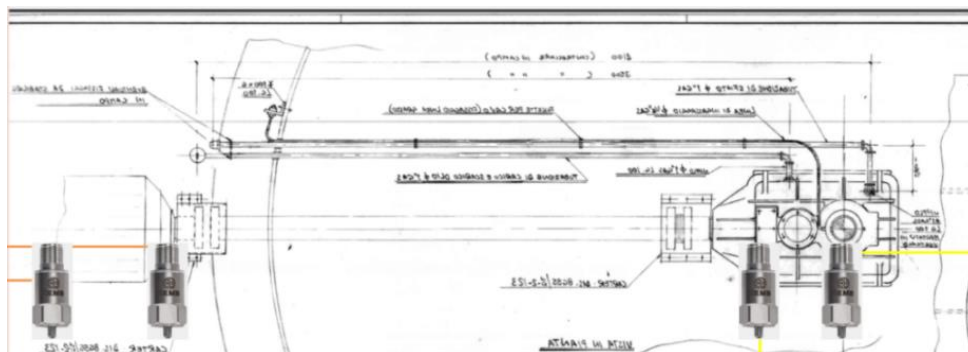


Figura 3.46 Esempio installazione trasduttori su ventilatore

Il monitoraggio della Pompa PAE e dei Ventilatori può essere schematizzato dalla seguente figura:

Ventilatore 1	Riduttore asse lento	Rockwell	Dewesoft Krypton ACC DB17018351
	Riduttore asse veloce		
	Motore lato asse		
	Motore lato ventola		
Ventilatore 2	Riduttore asse lento		Dewesoft Krypton ACC DB17018353
	Riduttore asse veloce		
	Motore lato asse		
	Motore lato ventola		
Ventilatore 3	Riduttore asse lento		Dewesoft Krypton ACC DB17018350
	Riduttore asse veloce		
	Motore lato asse		
	Motore lato ventola		
Pompa PAE 1	Cuscinetto motore superiore		Dewesoft Krypton ACC DB17018352
	Cuscinetto motore inferiore		
	Corpo pompa assiale		
	Corpo pompa radiale		
Pompa PAE 2	Girante X	Dewesoft Krypton ACC DB18003111	
	Girante Y		
	Corrente		
	(Disponibile)		

Figura 3.47 Schematizzazione monitoraggio Pompa PAE e Ventilatori

Per entrambe le utenze si è scelto un sistema ridondante di acquisizione dei dati e successiva analisi. La doppia piattaforma consentirà robustezza della misura ed efficienza nella disponibilità del dato, flessibilità nell'individuazione di algoritmi innovativi legati ai dati raccolti, individuazione della soluzione migliore da replicare sugli altri impianti. Dopo un'analisi di mercato su diversi prodotti e soluzioni, la scelta è ricaduta sui seguenti sistemi:

- Rockwell – Allen-Bradly Dynamix 1444 (di seguito nominato Sistema 1 - S1), caratterizzato dalla sua compattezza, scalabilità, protezione real-time legata alle misure effettuate e specificità rispetto a installazioni classiche di Condition Monitoring;
- Dewesoft Krypton (di seguito nominato Sistema 2 – S2), caratterizzato da elevate performance real-time, spiccate caratteristiche hardware e di processamento dati, significativa flessibilità nella gestione dei segnali e relativa implementazione di algoritmi non standard.

### 3.6.1 - Rockwell – Allen-Bradly Dynamix 1444

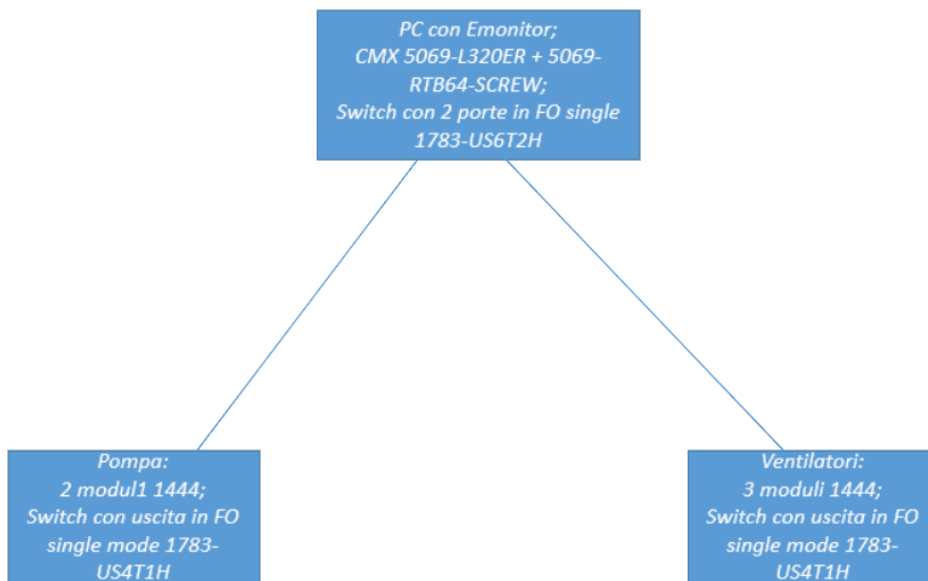


Figura 3.48 Schema generale architettura Rockwell per Pompa PAE e Ventilatori



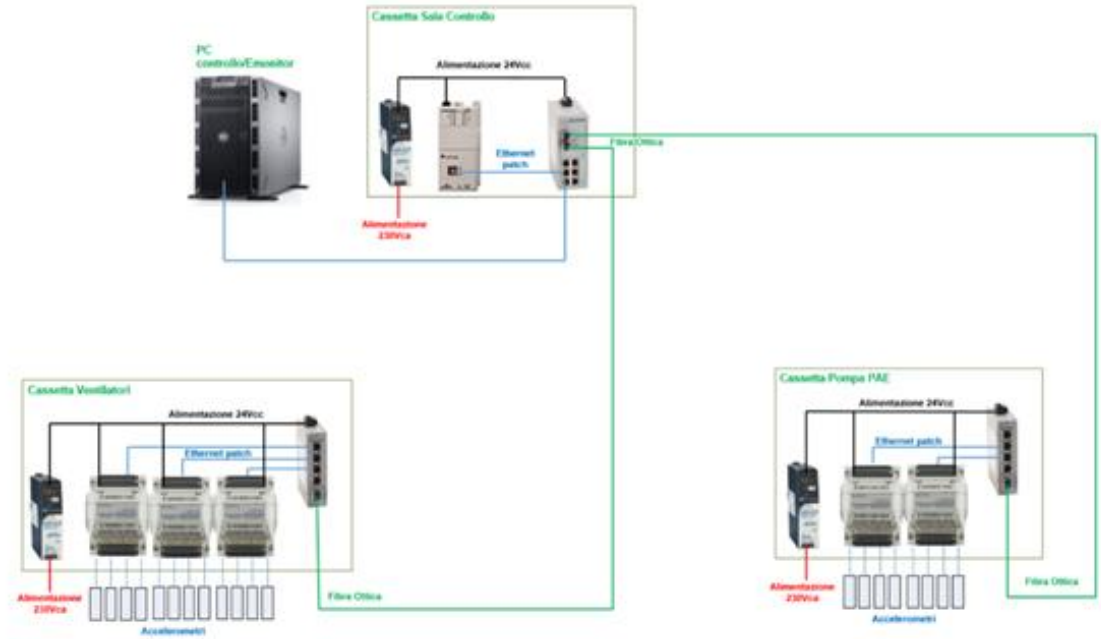


Figura 3.49 Schema di dettaglio architettura Rockwell per Pompa PAE e Ventilatori

### 3.6.2 - Dewesoft

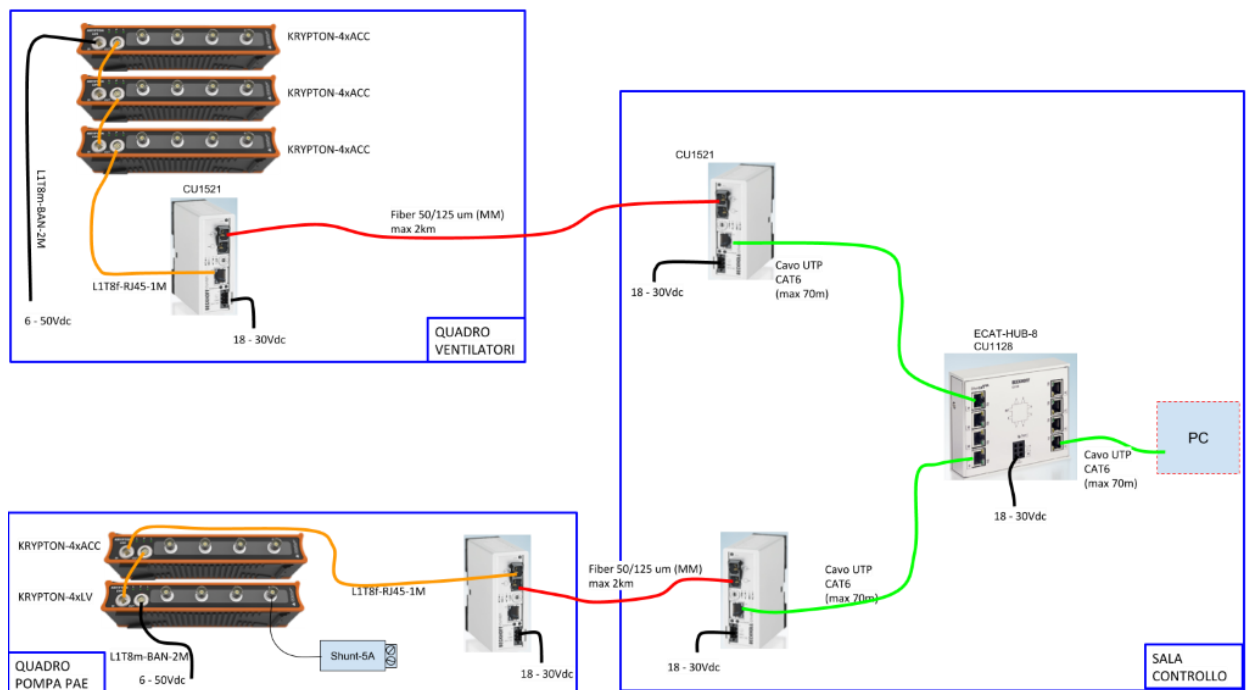


Figura 3.50 Schema di dettaglio architettura Dewesoft per Pompa PAE e Ventilatori

Per l'interfacciamento del PC con il sistema già presente in centrale, Dewesoft integra due soluzioni dedicate:

- OPC-UA Server, per i dati in uscita
- Plugin Siemens S7 per i dati in ingresso

In particolare, si è creato un meccanismo di aggiornamento del software Dewesoft attraverso client OPC-UA utile a sfruttare la potenza di calcolo di un Mainframe decentralizzato. I sistemi Dewesoft inviano in remoto i dati real-time attendendosi gli esiti degli algoritmi diagnostici e/o prognostici sviluppati. Per una prima analisi locale è stato utilizzato uno dei metodi poi implementati sul Mainframe centrale: il modello di analisi di serie temporali ARIMA descritto nel paragrafo 2.5.4 -. Questo software comunica con il software Dewesoft tramite il protocollo OPC UA: i dati vengono inviati al modello ARIMA che li elabora e stabilisce quando, dati i trend attuali e al netto del rumore, verrà superata una certa soglia ritenuta significativa ai fini della manutenzione.

Per quanto riguarda l'installazione hardware, viste le peculiarità delle due piattaforme, si è scelto di utilizzare due stadi di acquisizione in cascata, entrambe caratterizzati da ingressi analogici non campionati, elaborati e condizionati, ma direttamente collegati ai trasduttori in campo. Date le caratteristiche delle due piattaforme, S1 si occuperà in prima istanza di ricevere i segnali dai trasduttori e di fornire l'alimentazione necessaria al funzionamento degli stessi; successivamente, attraverso output dedicati, invierà gli stessi segnali a S2 senza far passare quest'ultimi dallo stadio di conversione A/D e quindi di elaborazione e condizionamento dei segnali. S2 riceverà in input ogni segnale come ingresso analogico da configurare opportunamente secondo le caratteristiche del relativo trasduttore, e non dovrà occuparsi in alcun modo di fornire a quest'ultimo l'alimentazione necessaria al suo funzionamento.

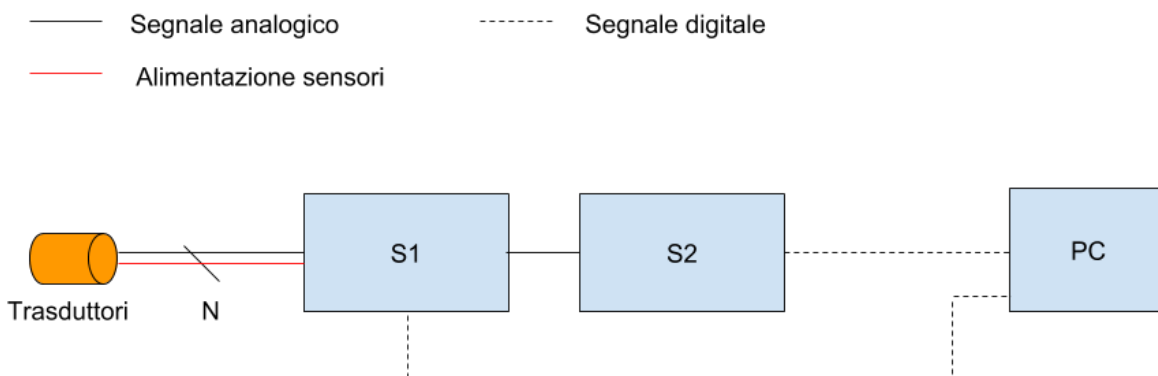


Figura 3.51 Architettura ridondante sistemi di acquisizione dati

## 3.7 - Applicazione di modelli diagnostici al compressore

Per la diagnostica del pompaggio del compressore centrifugo multistadio, sono stati costruiti due diversi modelli. Uno machine learning che sfrutta le *support vector machine* (SVM) ed uno statistico basato sul riconoscimento di forme d'onda tipiche per diagnosticare comportamenti riconoscibili al fenomeno in analisi.

Il fenomeno sotto analisi, per il quale avevamo alcuni dati a database anche relativi alla condizione di funzionamento anomalo, è il pompaggio del compressore, fenomeno dettagliato nel capitolo 3.4.2.5.

Questo fenomeno è da evitare in quanto causa forti vibrazioni al macchinario che potrebbero causare la rottura delle sue componenti meccaniche. Essendo dovuta a condizioni di funzionamento anomale, questo fenomeno si instaura molto rapidamente e non richiede nessun intervento manutentivo ma la gestione dell'apertura di valvole deputate all'incremento della portata in ingresso per scongiurare il fenomeno. Di conseguenza all'interno del caso studio non è stato sviluppato un modello prognostico (diagnostica più stima della vita utile rimanente) ma solo un modello diagnostico molto accurato sulla base del quale azionare le valvole di antipompaggio ed effettuare quindi controllo automatizzato. Per la tipologia di guasto, il modello sviluppato deve seguire la logica hard Real-time.

Come presentato nella metodologia, prima di procedere al training dei modelli diagnostici è necessario seguire una fase di acquisizione e preprocessing dei dati presenti a database.

### 3.7.1 - Modello diagnostico – SVM

Di seguito tutte le fasi necessarie al training e validazione del modello diagnostico sviluppato.

#### 3.7.1.1 Analisi ed elaborazione dei dati storici

Questa fase ha portato alla costruzione di un dataset in forma matriciale, composto da dati relativi alle condizioni di incipiente pompaggio e di regime, contrassegnati rispettivamente dall'etichetta 1 e 0.

#### 3.7.1.2 Importazione

Enel Green Power S.p.a. ha fornito a DIEF i dati storici d'impianto, raccolti tra Ottobre 2016 e Luglio 2017, organizzati in intervalli temporali di riferimento e con l'annotazione delle condizioni di funzionamento in tali intervalli: in totale sono stati importati 16 file relativi alle condizioni di regime e 12 file relativi alle

condizioni di pompaggio, oltre a 17 file sulle condizioni di anomalia, che tuttavia non sono stati presi in considerazione per l'applicazione della metodologia, in quanto non si riferivano ad anomalie connesse all'evento di pompaggio da prognosticare. Le condizioni segnalate come "pompaggio", contiene dati relativi a tutta l'evoluzione del macchinario fino all'insorgenza del pompaggio.

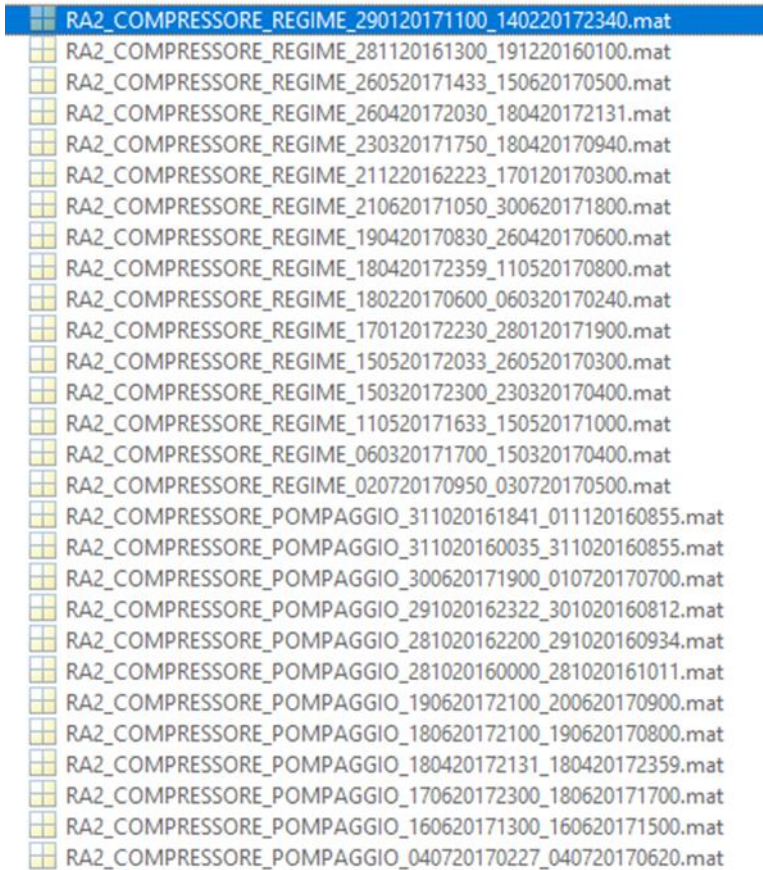


Figura 3.52 dati forniti da Enel Green Power S.p.a.

All'interno di ciascun file sono presenti 38 celle, una per ognuno dei sensori ritenuti rilevanti ai fini dello studio.

38x2 cell

	1	2
1	3x3825 double	'RA2_GS_TT_401A'
2	3x3798 double	'RA2_GS_TT_401B'
3	3x149229 double	'RA2_EL_W_801'
4	3x160650 double	'RA2_GS_DT_401'
5	3x159627 double	'RA2_GS_DT_402'
6	3x102152 double	'RA2_GS_DT_404'
7	3x156874 double	'RA2_GS_PT_402'
8	3x1339 double	'RA2_GS_DT_403'
9	3x3533 double	'RA2_GS_TT_400A'
10	3x3556 double	'RA2_GS_TT_400B'
11	3x2655 double	'RA2_GS_TT_402A'
12	3x2676 double	'RA2_GS_TT_402B'
13	3x2724 double	'RA2_GS_TT_403A'
14	3x2708 double	'RA2_GS_TT_403B'
15	3x5677 double	'RA2_GS_TT_404A'
16	3x5137 double	'RA2_GS_TT_404B'
17	3x5102 double	'RA2_GS_TT_405A'
18	3x5052 double	'RA2_GS_TT_405B'
19	3x4665 double	'RA2_GS_TT_422'
20	3x2752 double	'RA2_GS_TT_425A'
21	3x2723 double	'RA2_GS_TT_425B'
22	3x3174 double	'RA2_GS_TT_450'
23	3x2971 double	'RA2_GS_TT_451'
24	3x2929 double	'RA2_GS_TT_452'
25	3x2969 double	'RA2_GS_TT_453'
26	3x72957 double	'RA2_GS_BI_901'
27	3x81026 double	'RA2_GS_PT_404'
28	3x56499 double	'RA2_GS_PT_405'
29	3x25237 double	'RA2_GS_PT_406'
30	3x119352 double	'RA2_GS_PT_450'
31	3x40583 double	'RA2_GS_PT_451'
32	3x103905 double	'RA2_GS_PT_452'
33	3x53912 double	'RA2_GS_PT_453'
34	3x144358 double	'RA2_GS_PT_401'
35	3x39450 double	'RA2_TU_PT_301A'
36	3x10286 double	'RA2_GS_ZT_401'
37	3x4414 double	'RA2_GS_ZT_402'
38	3x144358 double	'RA2_GS_PT_401'

Figura 3.53 celle e sensori corrispondenti

Ogni cella è composta da 3 vettori riga:

1. Vettore dei valori misurati dal sensore,
2. Vettore degli indicatori della bontà del dato acquisito (pari a 192 se il sensore funziona bene),
3. Vettore degli istanti temporali di campionamento.

Ogni sensore ha la propria frequenza di campionamento, da ciò deriva la differenza tra le dimensioni delle varie celle.

output{1, 1}										
	1	2	3	4	5	6	7	8	9	10
1	28.4121	28.3790	28.3281	28.3014	28.2403	28.2229	28.2772	28.3551	28.4294	28.4660
2	192	192	192	192	192	192	192	192	192	192
3	7.3688e+05	7.3688e+05	7.3688e+05	7.3688e+05	7.3688e+05	7.3688e+05	7.3688e+05	7.3688e+05	7.3688e+05	7.3688e+05

Figura 3.54 contenuto delle celle

Le grandezze fisiche rilevate sono riportate in tabella, e comprendono misure di pressione, di portata, di temperatura, di potenza in ingresso al compressore, di posizionamento delle valvole di regolazione. Si può, dunque, osservare come si sia esteso lo studio rispetto alla determinazione della curva critica standard fornita dai produttori di compressori, non limitando l'analisi ai soli valori di pressione, portata e numero di giri. La scelta di questi sensori permette di monitorare le condizioni del sistema di estrazione e di mettere in luce gli effetti provocati dall'instabilità del compressore.

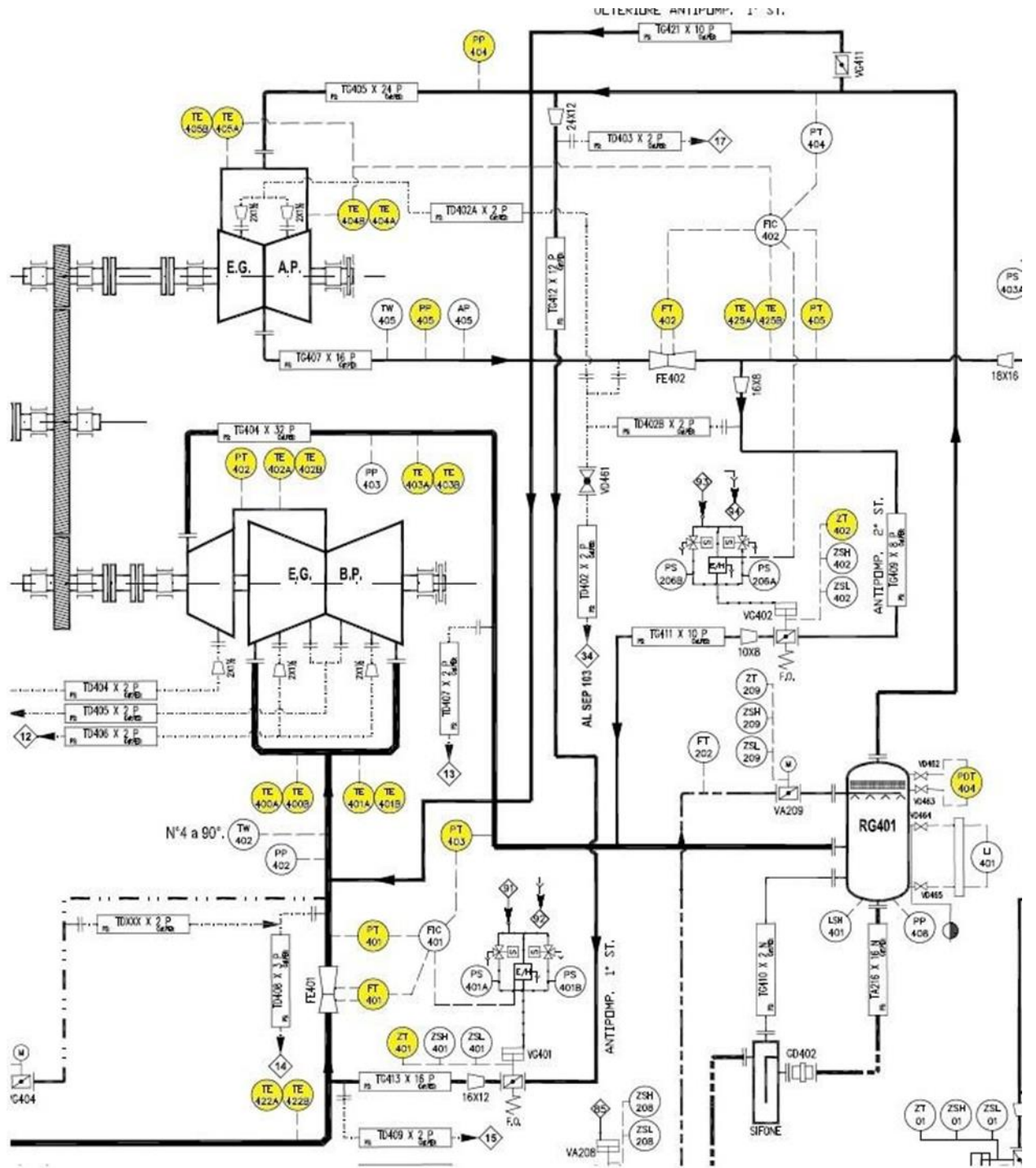


Figura 3.55 gran parte dei sensori a cui fanno riferimento i dati (evidenziati)

Tabella 3.6 sigla dei sensori e grandezze misurate.

Sensore	Descrizione
RA2_GS_TT_401A	TEMPERATURA ASPIRAZIONE GAS BP LATO OPPOSTO MOLTIPLICATORE
RA2_GS_TT_401B	TEMPERATURA ASPIRAZIONE GAS BP LATO OPPOSTO MOLTIPLICATORE
RA2_EL_W_801	POTENZA ATTIVA NETTA
RA2_GS_DT_401	PORTATA GAS STADIO BP

RA2_GS_DT_402	PORTATA GAS STADIO AP
RA2_GS_DT_404	PRESSIONE DIFFRENZIALE DEMISTER RG
RA2_GS_PT_402	PRESSIONE GAS ASPIRAZIONE MP
RA2_GS_DT_403	PORTATA GAS SCARICO
RA2_GS_TT_400A	TEMPERATURA ASPIRAZIONE GAS BP LATO MOLTIPLICATORE
RA2_GS_TT_400B	TEMPERATURA ASPIRAZIONE GAS BP LATO MOLTIPLICATORE
RA2_GS_TT_402A	TEMPERATURA GAS MANDATA 1 <sup>o</sup> STADIO EG
RA2_GS_TT_402B	TEMPERATURA GAS MANDATA 1 <sup>o</sup> STADIO EG
RA2_GS_TT_403A	TEMPERATURA GAS MANDATA 2 <sup>o</sup> STADIO EG
RA2_GS_TT_403B	TEMPERATURA GAS MANDATA 2 <sup>o</sup> STADIO EG
RA2_GS_TT_404A	TEMPERATURA GAS ASPIRAZIONE 3 <sup>o</sup> STADIO SX
RA2_GS_TT_404B	TEMPERATURA GAS ASPIRAZIONE 3 <sup>o</sup> STADIO SX
RA2_GS_TT_405A	TEMPERATURA GAS ASPIRAZIONE AP LATO MOLTIPLICATORE
RA2_GS_TT_405B	TEMPERATURA GAS ASPIRAZIONE AP LATO MOLTIPLICATORE
RA2_GS_TT_422	TEMPERATURA GAS USCITA CONDENSATORE
RA2_GS_TT_425A	TEMPERATURA GAS USCITA STADIO AP
RA2_GS_TT_425B	TEMPERATURA GAS USCITA STADIO AP
RA2_GS_TT_450	TEMPERATURA INTERSTADIO ESTRATTORE GAS
RA2_GS_TT_451	TEMPERATURA INTERSTADIO ESTRATTORE GAS
RA2_GS_TT_452	TEMPERATURA INTERSTADIO ESTRATTORE GAS
RA2_GS_TT_453	TEMPERATURA INTERSTADIO ESTRATTORE GAS
RA2_GS_BI_901	TEMPERATURA BULBO UMIDO
RA2_GS_PT_404	PRESSIONE GAS ASPIRAZIONE 3 <sup>o</sup> STADIO
RA2_GS_PT_405	PRESSIONE GAS USCITA STADIO AP
RA2_GS_PT_406	PRESSIONE SCARICO GAS
RA2_GS_PT_450	PRESSIONE INTERSTADIO ESTRATTORE GAS
RA2_GS_PT_451	PRESSIONE INTERSTADIO ESTRATTORE GAS
RA2_GS_PT_452	PRESSIONE INTERSTADIO ESTRATTORE GAS
RA2_GS_PT_453	PRESSIONE INTERSTADIO ESTRATTORE GAS
RA2_GS_PT_401	PRESSIONE GAS ASPIRAZIONE BP
RA2_TU_PT_301A	PRESSIONE ASSOLUTA CONDENSATORE
RA2_GS_ZT_401	POSIZIONE VALVOLA VG401
RA2_GS_ZT_402	POSIZIONE VALVOLA VG402
RA2_GS_PT_401	PRESSIONE GAS ASPIRAZIONE BP

### 3.7.1.3 -Sincronizzazione

Poiché i sensori hanno frequenze di campionamento diverse, è stato necessario sincronizzarli con un'interpolazione lineare. Sono stati individuati gli estremi temporali per l'interpolazione e, sull'intervallo risultante, è stata applicata la funzione `interp1` di MATLAB®. Tale procedura ha permesso di ottenere segnali sincronizzati e delle medesime dimensioni, all'interno dello stesso file di dati, con osservazioni intervallate da finestre temporali di un secondo.



	1	2
1	3x50261 double	'RA2_GS_TT_401A'
2	3x50261 double	'RA2_GS_TT_401B'
3	3x50261 double	'RA2_EL_W_801'
4	3x50261 double	'RA2_GS_DT_401'
5	3x50261 double	'RA2_GS_DT_402'
6	3x50261 double	'RA2_GS_DT_404'
7	3x50261 double	'RA2_GS_PT_402'
8	3x50261 double	'RA2_GS_DT_403'
9	3x50261 double	'RA2_GS_TT_400A'
10	3x50261 double	'RA2_GS_TT_400B'
11	3x50261 double	'RA2_GS_TT_402A'
12	3x50261 double	'RA2_GS_TT_402B'
13	3x50261 double	'RA2_GS_TT_403A'
14	3x50261 double	'RA2_GS_TT_403B'
15	3x50261 double	'RA2_GS_TT_404A'
16	3x50261 double	'RA2_GS_TT_404B'

Figura 3.56 sensori sincronizzati all'interno di un file

### 3.7.1.4 -Raggruppamento delle ridondanze

Tra i dati importati ci sono varie misure ridondanti. Ai fini di una preliminare riduzione delle variabili e per una maggior semplicità di analisi, si è calcolata la media aritmetica dei valori delle misure ridondanti, e se n'è memorizzato il risultato in celle aggiuntive. Sul momento, le misure ridondanti sono state mantenute, per poi essere eliminate nelle ultime fasi della metodologia. Il risultato è l'ampliamento delle celle da 38 a 45. Le ridondanze sono facilmente individuabili, in quanto caratterizzate dall'aggiunta della lettera A o B al termine della sigla.

Tabella 3.7 Tabella delle ridondanze.

Sensore	Descrizione
RA2_GS_TT_401A	TEMPERATURA ASPIRAZIONE GAS BP LATO OPPOSTO MOLTIPLICATORE
RA2_GS_TT_401B	TEMPERATURA ASPIRAZIONE GAS BP LATO OPPOSTO MOLTIPLICATORE
RA2_GS_TT_400A	TEMPERATURA ASPIRAZIONE GAS BP LATO MOLTIPLICATORE
RA2_GS_TT_400B	TEMPERATURA ASPIRAZIONE GAS BP LATO MOLTIPLICATORE
RA2_GS_TT_402A	TEMPERATURA GAS MANDATA 1Â° STADIO EG
RA2_GS_TT_402B	TEMPERATURA GAS MANDATA 1Â° STADIO EG
RA2_GS_TT_403A	TEMPERATURA GAS MANDATA 2Â° STADIO EG
RA2_GS_TT_403B	TEMPERATURA GAS MANDATA 2Â° STADIO EG
RA2_GS_TT_404A	TEMPERATURA GAS ASPIRAZIONE 3Â° STADIO SX
RA2_GS_TT_404B	TEMPERATURA GAS ASPIRAZIONE 3Â° STADIO SX

RA2_GS_TT_405A	TEMPERATURA GAS ASPIRAZIONE AP LATO MOLTIPLICATORE
RA2_GS_TT_405B	TEMPERATURA GAS ASPIRAZIONE AP LATO MOLTIPLICATORE
RA2_GS_TT_425A	TEMPERATURA GAS USCITA STADIO AP
RA2_GS_TT_425B	TEMPERATURA GAS USCITA STADIO AP

45x2 cell

	1	2
34	3x50261 double	'RA2_GS_PT_401'
35	3x50261 double	'RA2_TU_PT_301A'
36	3x50261 double	'RA2_GS_ZT_401'
37	3x50261 double	'RA2_GS_ZT_402'
38	3x50261 double	'RA2_GS_PT_401'
39	3x50261 double	'Media tra 9 e 10'
40	3x50261 double	'Media tra 11 e 12'
41	3x50261 double	'Media tra 1 e 2'
42	3x50261 double	'Media tra 13 e 14'
43	3x50261 double	'Media tra 15 e 16'
44	3x50261 double	'Media tra 17 e 18'
45	3x50261 double	'Media tra 20 e 21'

Figura 3.57 valori medi delle ridondanze

### 3.7.1.5 Filtro

I segnali sono risultati essere contraddistinti da valori anomali. A tal proposito è stato realizzato un filtraggio dei dati. Le misure fornite dal sensore RA2\_GS\_DT\_404 (PRESSIONE DIFFERENZIALE DEMISTER RG) e dal sensore RA2\_GS\_PT\_450 (PRESSIONE INTERSTADIO ESTRATTORE GAS) si sono rivelate poco leggibili, oltre che spesso caratterizzate da valori poco sensati. Il sensore RA2\_GS\_DT\_403 (PORTATA GAS SCARICO), invece, è risultato non funzionante, in quanto settato costantemente sul valore nullo. Perciò, dato lo scarso contenuto informativo da esse apportato e la marginalità del loro contributo, è stato deciso di eliminare dall'analisi queste variabili nelle ultime fasi della metodologia.

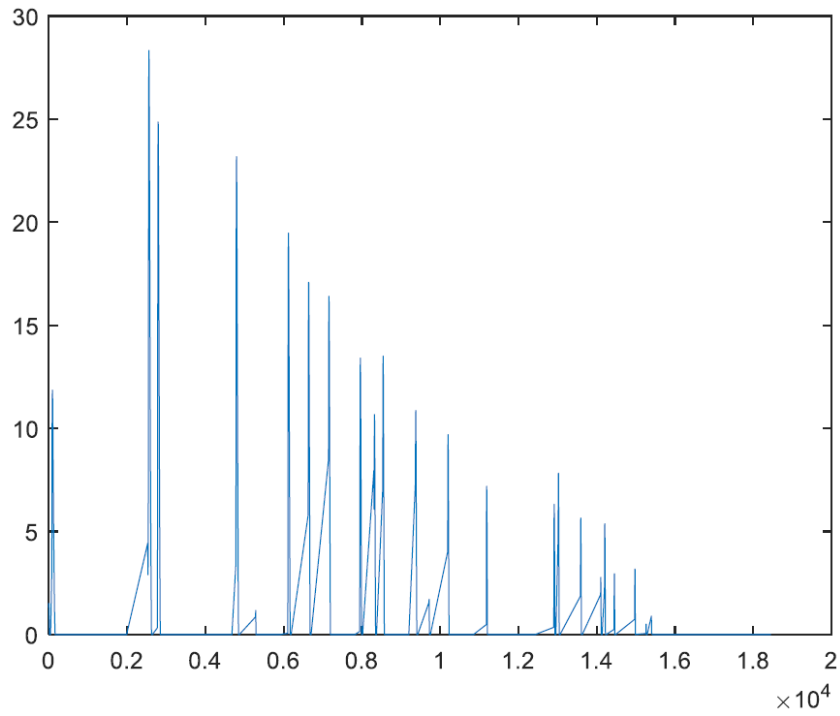


Figura 3.58 esempio di andamento delle misure rilevate dal sensore RA2\_GS\_DT\_404.

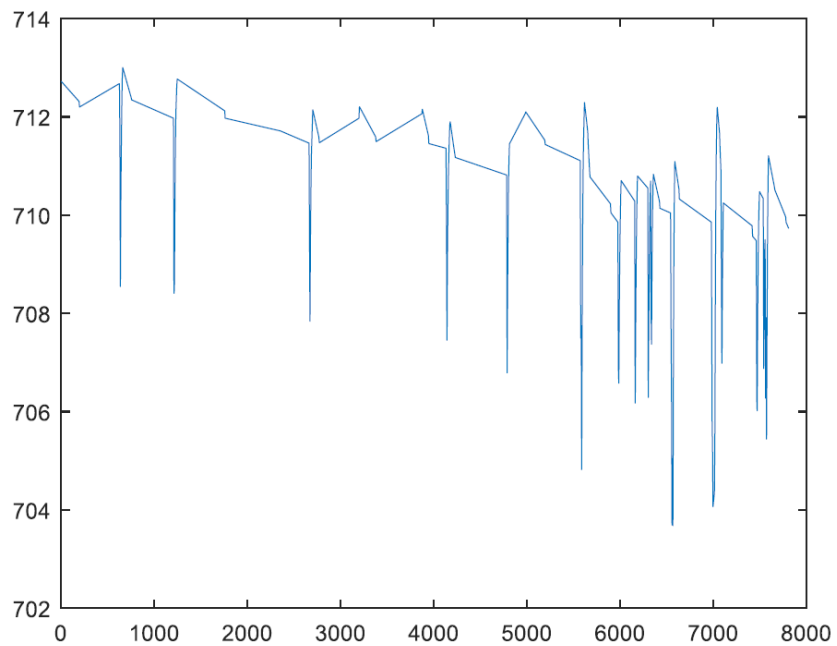


Figura 3.59 esempio di andamento delle misure rilevate dal sensore RA2\_GS\_PT\_450.

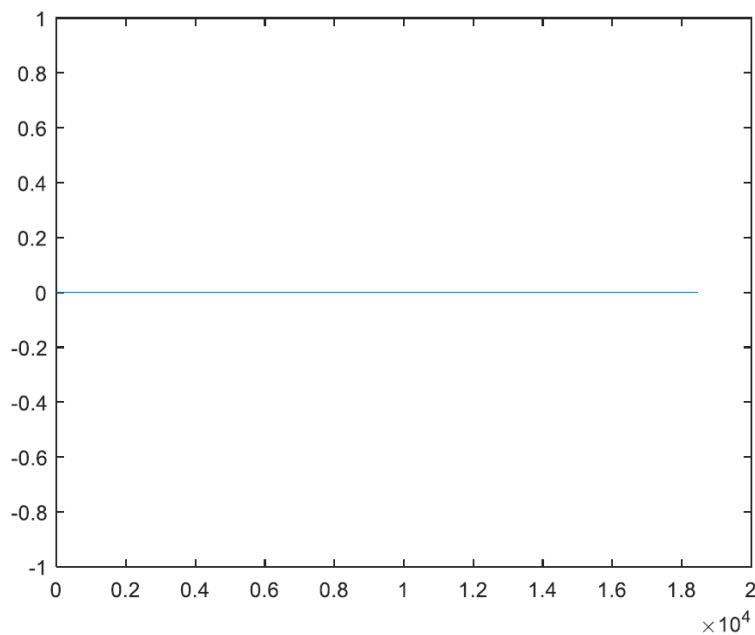


Figura 3.60 malfunzionamento del sensore RA2\_GS\_DT\_403.

Inoltre, è stato escluso dall'analisi il file RA2\_COMPRESSORE\_POMPAGGIO\_160620171300\_160620171500.mat, in quanto il sensore non ha campionato in tale intervallo una fondamentale grandezza per la definizione delle condizioni di pompaggio, quale la portata del gas nello stadio di bassa pressione (sensore RA2\_GS\_DT\_401).

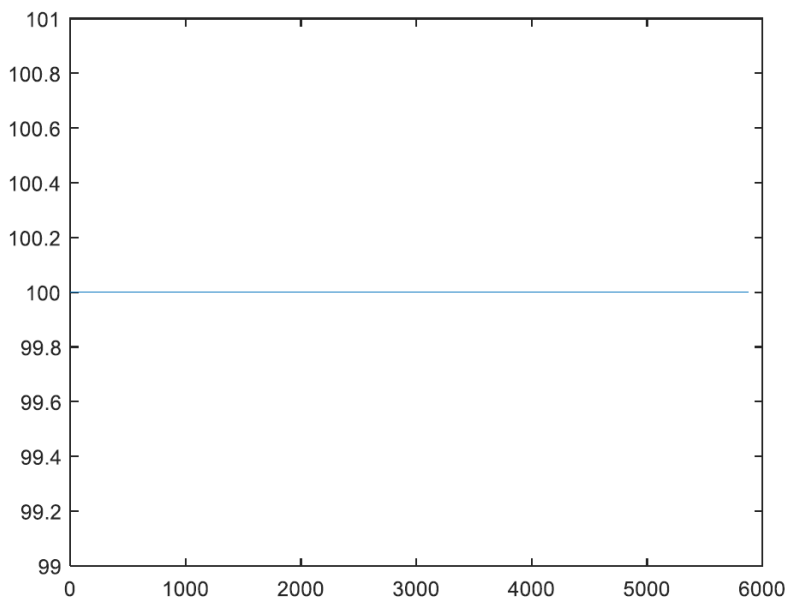


Figura 3.61 malfunzionamento del sensore RA2\_GS\_DT\_401 nel file RA2\_COMPRESSORE\_POMPAGGIO\_160620171300\_160620171500.mat.

Dall'analisi dell'andamento delle varie grandezze, sono state catalogate come oscillanti:

Tabella 3.8 grandezze oscillanti.

Sensore	Descrizione
RA2_EL_W_801	POTENZA ATTIVA NETTA
RA2_GS_DT_401	PORTATA GAS STADIO BP
RA2_GS_DT_402	PORTATA GAS STADIO AP
RA2_GS_PT_402	PRESSIONE GAS ASPIRAZIONE MP
RA2_GS_PT_404	PRESSIONE GAS ASPIRAZIONE 3Â° STADIO
RA2_GS_PT_405	PRESSIONE GAS USCITA STADIO AP
RA2_GS_PT_406	PRESSIONE SCARICO GAS
RA2_GS_PT_452	PRESSIONE INTERSTADIO ESTRATTORE GAS
RA2_GS_PT_401	PRESSIONE GAS ASPIRAZIONE BP
RA2_TU_PT_301A	PRESSIONE ASSOLUTA CONDENSATORE

Per queste grandezze si è, in primo luogo, applicato il codice di filtro descritto in precedenza, coi parametri ottimali trovati per via iterativa. Successivamente, nel caso fosse necessario, si è pulito ulteriormente il segnale, o riapplicando il codice o eliminando manualmente gli outliers.

Sono state individuate due tipologie di dati di pompaggio: un insieme di dati riguarda le condizioni di funzionamento del compressore precedenti allo sviluppo dell'instabilità, mentre un altro insieme comprende sia la parte antecedente sia le prime fasi di evoluzione del fenomeno. In quest'ultimo caso, è stato necessario fissare il limite di inizio dell'instabilità, mentre nel primo caso si è potuto considerare l'intero intervallo temporale monitorato.

Per quanto riguarda le grandezze lento-varianti, non è stato necessario alcun filtraggio.

Di seguito si riportano, a titolo di esempio, i risultati ottenuti dal filtraggio per le grandezze oscillanti, in termini di valori dei parametri ottimali del codice di filtro e di rappresentazione del segnale filtrato, relativamente ai dati contenuti nel file RA2\_COMPRESSORE\_POMPAGGIO\_311020161841\_011120160855.mat. Si riporta, inoltre, l'andamento delle grandezze lento-varianti (già sottoposte alla fase di interpolazione).

### Grandezze oscillanti

Tabella 3.9 parametri ottimali per il codice di filtro.

Sensore	alfa	beta	t	k	postcut	precut
RA2_GS_PT_401	6	100	200	1200	30	0
RA2_EL_W_801	6	6	400	1200	20	0
RA2_GS_DT_401	4	16	500	1200	25	0

RA2_GS_DT_402	4	6	200	1200	25	0
RA2_GS_PT_402	6	20	200	1200	30	0
RA2_GS_PT_404	8	16	200	1200	30	0
RA2_GS_PT_405	15	50	200	1200	20	0
RA2_GS_PT_406	15	50	200	1200	30	0
RA2_GS_PT_452	20	20	200	1200	30	0
RA2_TU_PT_301A	6	100	200	1200	30	0

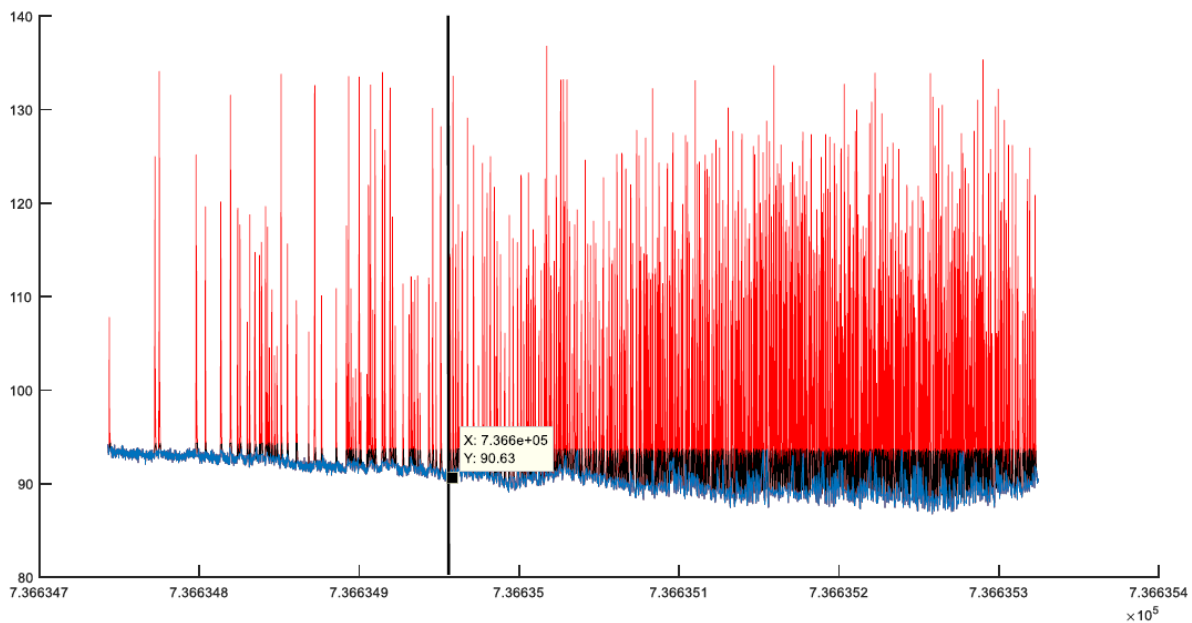


Figura 3.62 filtro delle misure rilevate dal sensore RA2\_GS\_PT\_401 con limite di inizio del pompaggio.

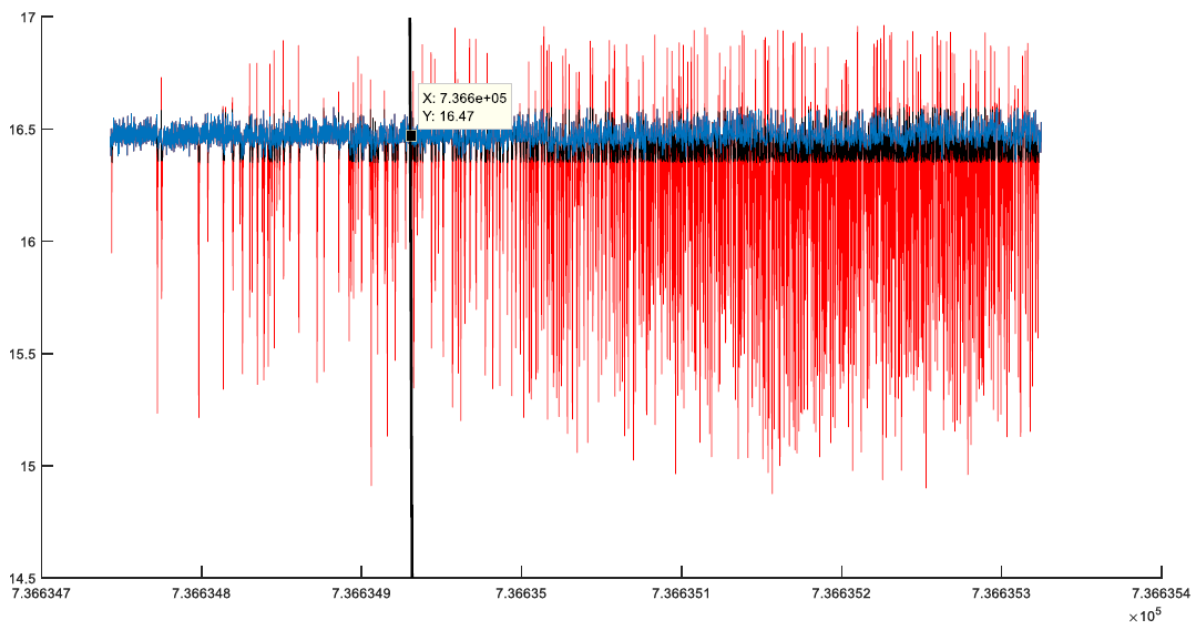


Figura 3.63 filtro delle misure rilevate dal sensore RA2\_EL\_W\_801 con limite di inizio del pompaggio.

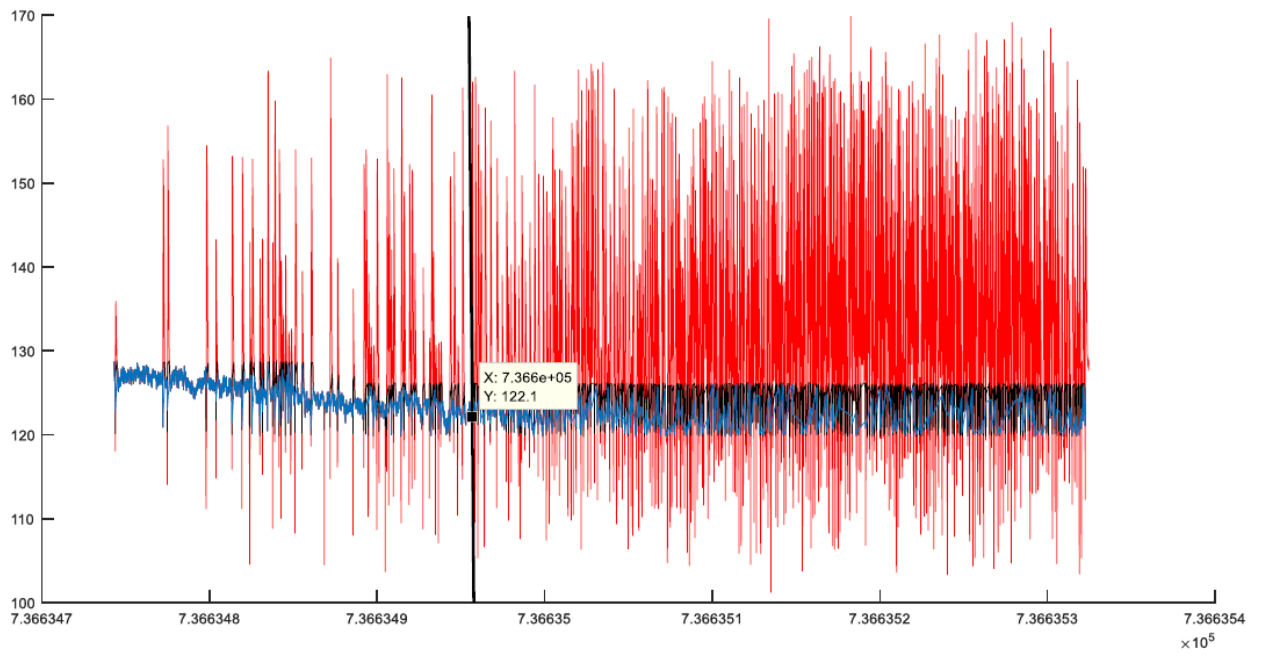


Figura 3.64 filtro delle misure rilevate dal sensore RA2\_GS\_DT\_401 con limite di inizio del pompaggio.

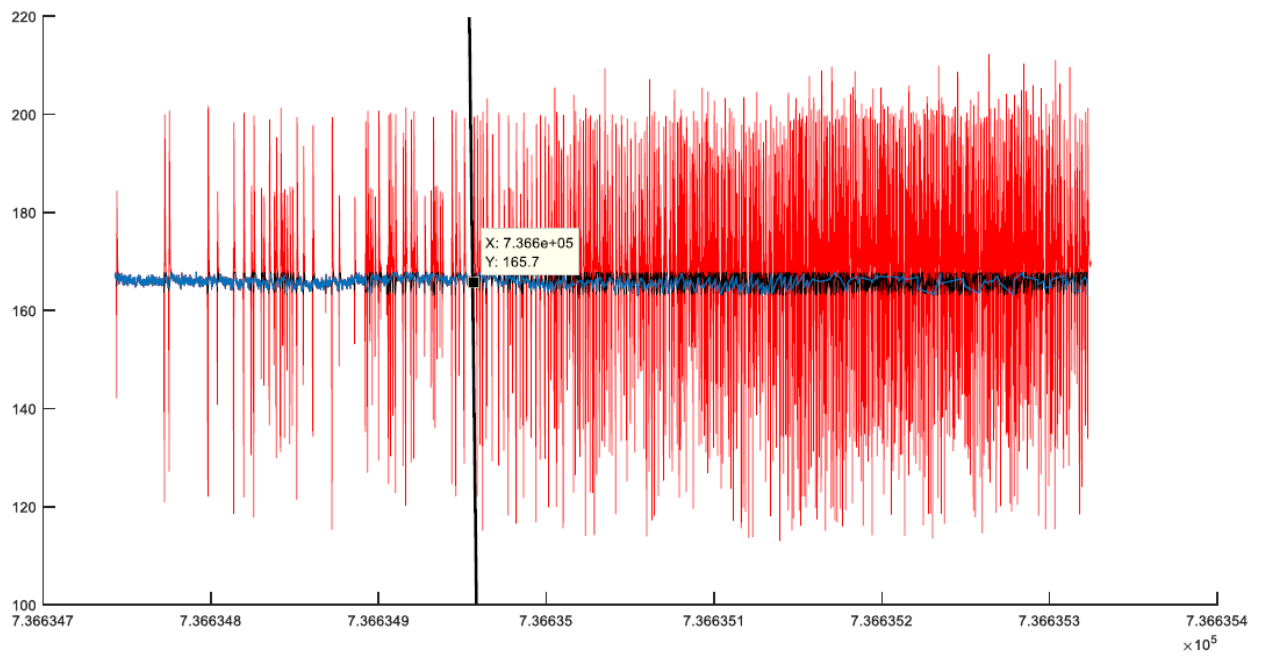


Figura 3.65 filtro delle misure rilevate dal sensore RA2\_GS\_DT\_402 con limite di inizio del pompaggio.

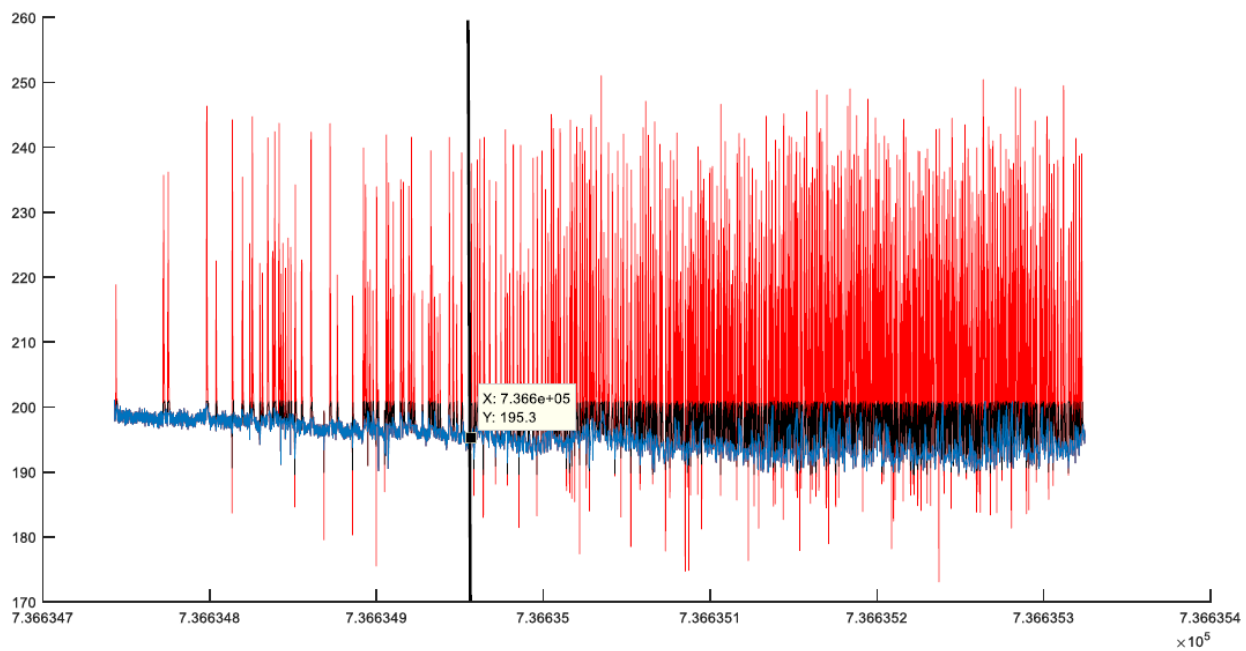


Figura 3.66 filtro delle misure rilevate dal sensore RA2\_GS\_PT\_402 con limite di inizio del pompaggio.

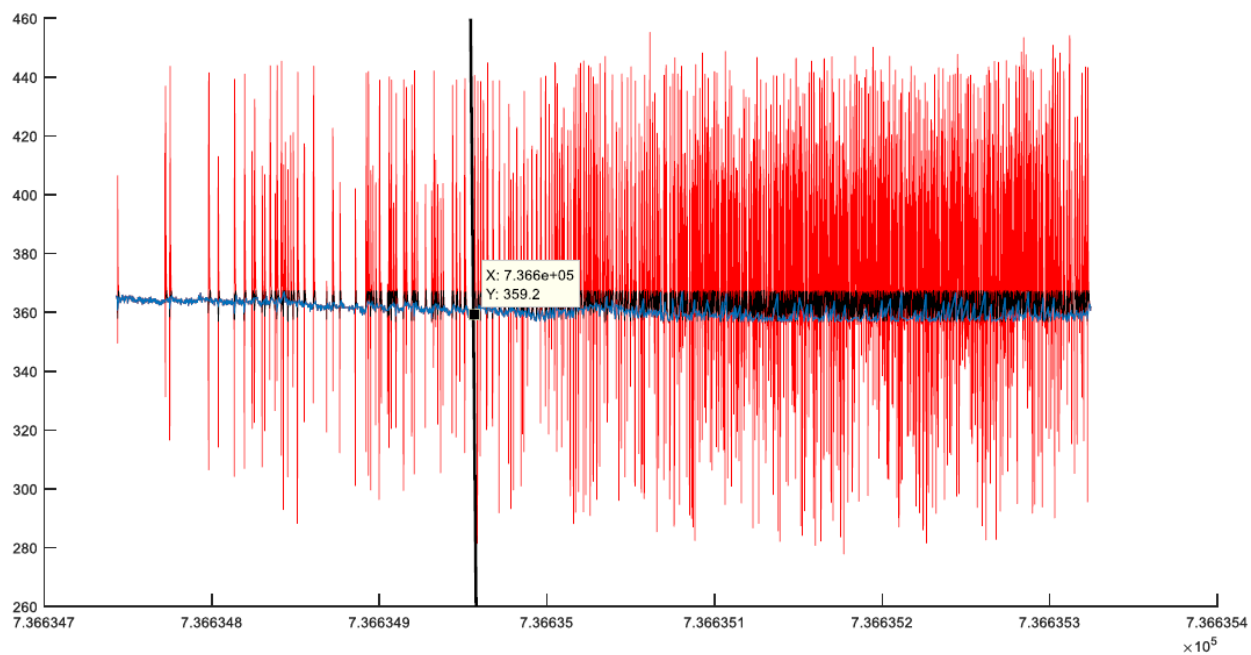


Figura 3.67 filtro delle misure rilevate dal sensore RA2\_GS\_PT\_404 con limite di inizio del pompaggio.



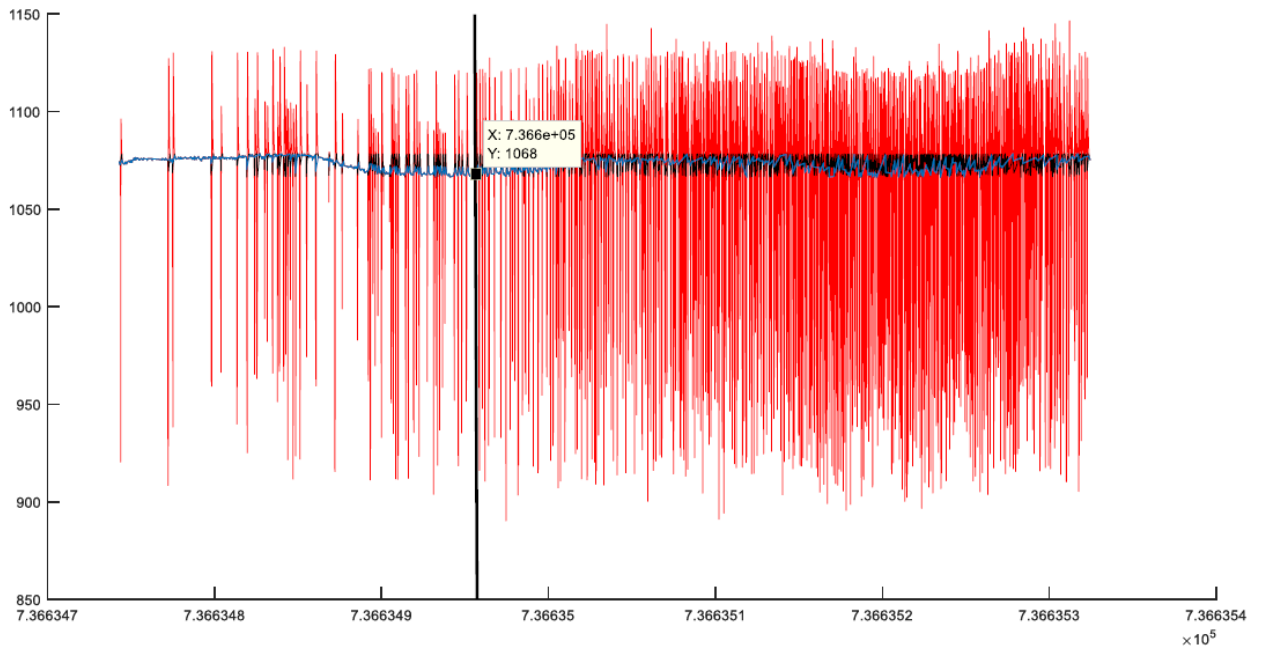


Figura 3.68 filtro delle misure rilevate dal sensore RA2\_GS\_PT\_405 con limite di inizio del pompaggio.

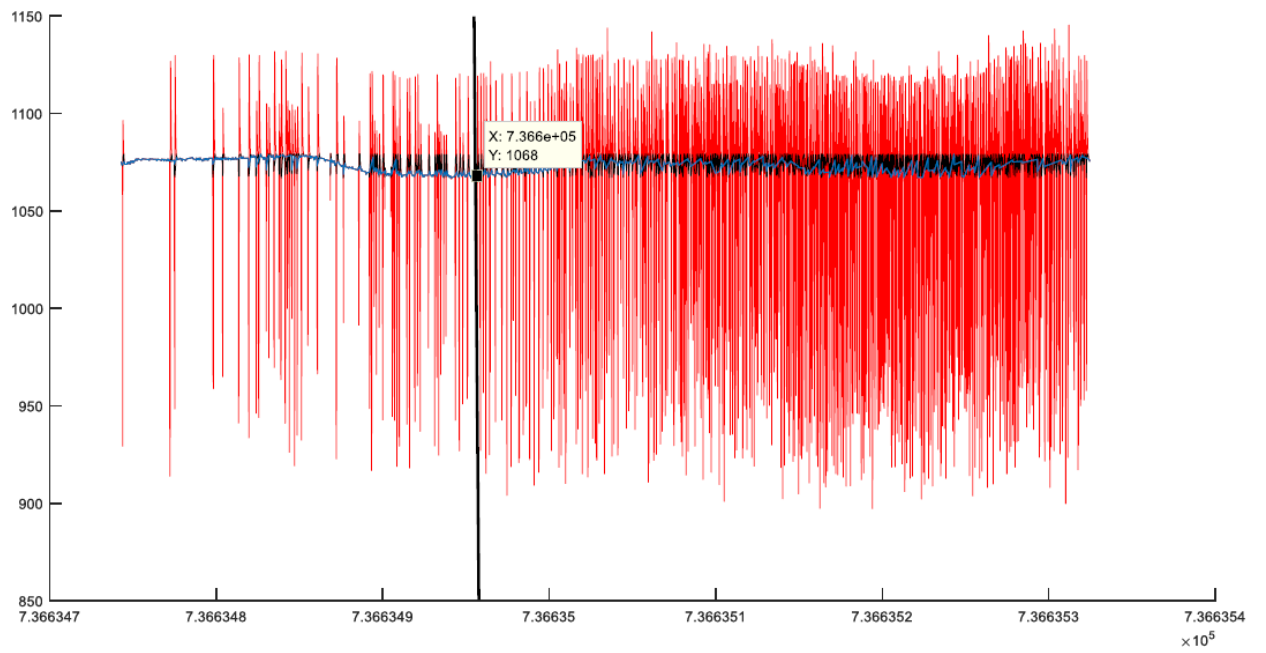


Figura 3.69 filtro delle misure rilevate dal sensore RA2\_GS\_PT\_406 con limite di inizio del pompaggio.

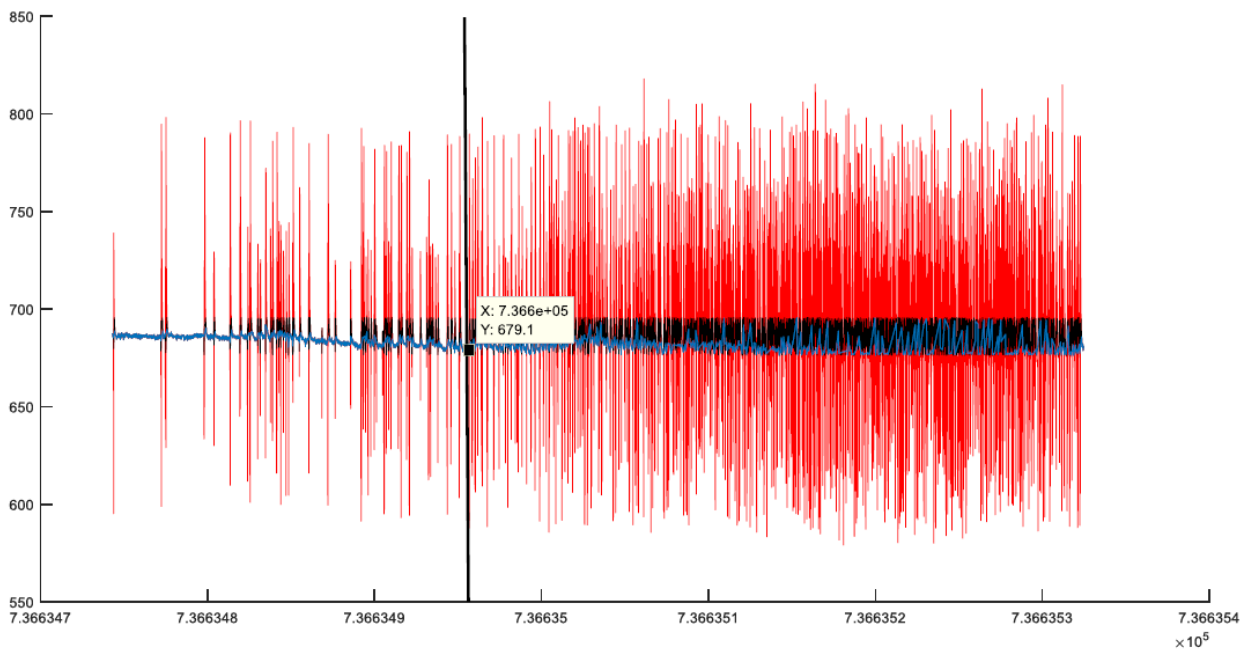


Figura 3.70 filtro delle misure rilevate dal sensore RA2\_GS\_PT\_452 con limite di inizio del pompaggio.

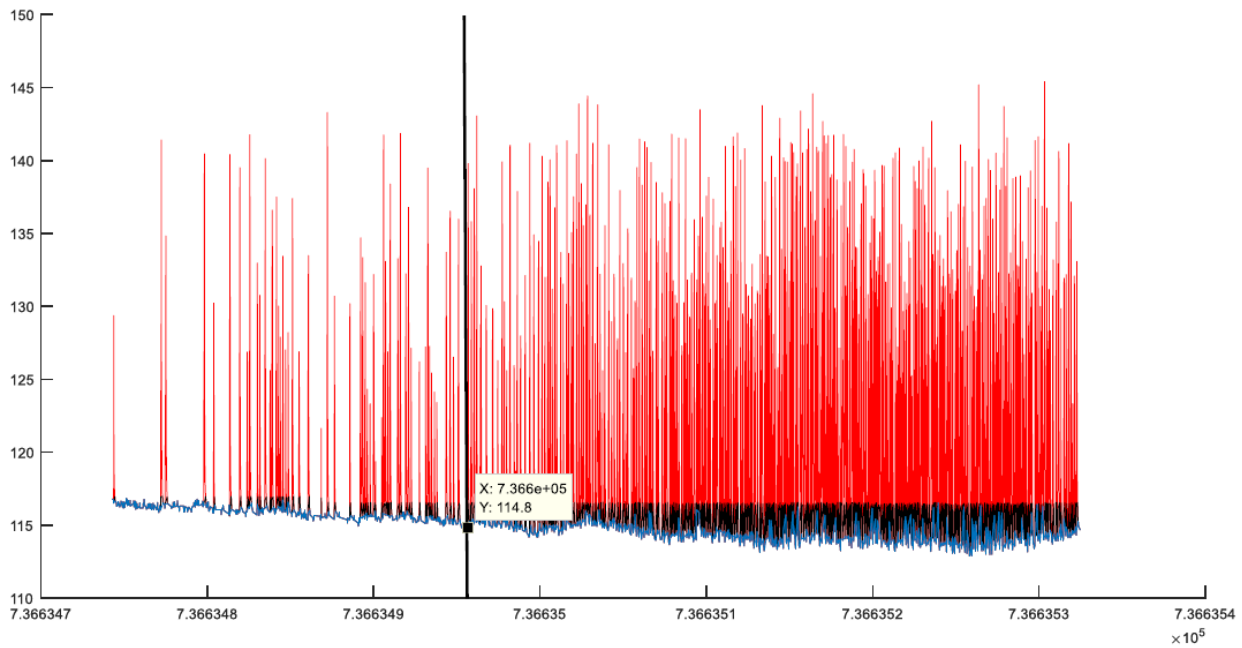


Figura 3.71 filtro delle misure rilevate dal sensore RA2\_GS\_PT\_301 con limite di inizio del pompaggio.

I dati relativi al periodo di “regime” hanno presentato alcune anomalie, come, ad esempio, il raggiungimento del fondo scala da parte di alcune misure. In questi casi si è agito manualmente per eliminare i rari errori di misura.

### 3.7.1.6 Interpolazione

Data l'ingente attività di filtraggio del segnale, è stato opportuno colmare i vuoti, dovuti all'eliminazione dei valori anomali, e risincronizzare tra loro i sensori. Per far ciò, una volta individuati gli estremi temporali di interpolazione, è stata applicata la funzione MATLAB® *interp1*, che ha risincronizzato i sensori e ristabilito il periodo di campionamento di un secondo.

Per i dati di pompaggio, è stato considerato come estremo superiore dell'intervallo:

1. il minimo dei limiti di inizio del pompaggio, nel caso di dati con sviluppo del pompaggio,
2. il minimo tra gli ultimi istanti temporali di osservazione, nel caso di dati relativi alla sola incipienza del fenomeno.

Ciò è stato fatto coerentemente con gli obiettivi dello studio qui presentato, in quanto interessano le condizioni di funzionamento del compressore nei momenti antecedenti lo sviluppo dell'instabilità.

### 3.7.1.7 Creazione del dataset

Una volta sincronizzati i sensori, è stato generato un dataset che rispettasse il formato descritto nel capitolo di presentazione della metodologia. Dopo aver organizzato i dati in una forma ordinata e più compatta, rispetto all'output della fase di interpolazione, questi sono stati fatti confluire in un'unica matrice, denominata *Dataset.mat*.

Tuttavia, data l'estrema abbondanza di dati di regime rispetto a quelli di pompaggio (18.120.861 osservazioni contro 307.169), è stato prima effettuato un bilanciamento: sono stati estratti in maniera casuale un numero di osservazioni all'interno dei dati di regime, pari alla dimensione del campione di dati di pompaggio, e l'estrazione è stata ripartita equamente su tutti i file relativi alle condizioni di regime. Quindi, avendo 307.169 osservazioni di instabilità e 16 file di regime, si sono estratte 19.198 osservazioni da ciascun file di regime.

Inoltre, sono state eliminate le colonne corrispondenti ai sensori esclusi dall'analisi, di cui si è parlato nel capitolo di filtraggio dei dati.

Di seguito si riporta l'ordine delle attività di creazione del dataset, con annesse immagini esemplificative.

### Fase 1: Unione dei dati

A partire dai dati di pompaggio sincronizzati, si è proceduto all'unione dei dati provenienti dai vari file in una matrice di celle, avente sulle righe i file e sulle colonne i sensori; sono stati memorizzati solo i valori delle misure e non gli istanti temporali di campionamento.

	1	2	3	4	5	6	7	8	9	10	11	12
1	1x18450 do...	1x18450 do...	1x18450 do...	1x18450 do...	1x18450 do...	1x18450 do...	1x18450 do...	1x18450 do...	1x18450 do...	1x18450 do...	1x18450 do...	1x18450 do...
2	1x63451 do...	1x63451 do...	1x63451 do...	1x63451 do...	1x63451 do...	1x63451 do...	1x63451 do...	1x63451 do...	1x63451 do...	1x63451 do...	1x63451 do...	1x63451 do...
3	1x8870 do...	1x8870 do...	1x8870 do...	1x8870 do...	1x8870 do...	1x8870 do...	1x8870 do...	1x8870 do...	1x8870 do...	1x8870 do...	1x8870 do...	1x8870 do...
4	1x14657 do...	1x14657 do...	1x14657 do...	1x14657 do...	1x14657 do...	1x14657 do...	1x14657 do...	1x14657 do...	1x14657 do...	1x14657 do...	1x14657 do...	1x14657 do...
5	1x42103 do...	1x42103 do...	1x42103 do...	1x42103 do...	1x42103 do...	1x42103 do...	1x42103 do...	1x42103 do...	1x42103 do...	1x42103 do...	1x42103 do...	1x42103 do...
6	1x34367 do...	1x34367 do...	1x34367 do...	1x34367 do...	1x34367 do...	1x34367 do...	1x34367 do...	1x34367 do...	1x34367 do...	1x34367 do...	1x34367 do...	1x34367 do...
7	1x6023 do...	1x6023 do...	1x6023 do...	1x6023 do...	1x6023 do...	1x6023 do...	1x6023 do...	1x6023 do...	1x6023 do...	1x6023 do...	1x6023 do...	1x6023 do...
8	1x35678 do...	1x35678 do...	1x35678 do...	1x35678 do...	1x35678 do...	1x35678 do...	1x35678 do...	1x35678 do...	1x35678 do...	1x35678 do...	1x35678 do...	1x35678 do...
9	1x36679 do...	1x36679 do...	1x36679 do...	1x36679 do...	1x36679 do...	1x36679 do...	1x36679 do...	1x36679 do...	1x36679 do...	1x36679 do...	1x36679 do...	1x36679 do...
10	1x38483 do...	1x38483 do...	1x38483 do...	1x38483 do...	1x38483 do...	1x38483 do...	1x38483 do...	1x38483 do...	1x38483 do...	1x38483 do...	1x38483 do...	1x38483 do...
11	1x7808 do...	1x7808 do...	1x7808 do...	1x7808 do...	1x7808 do...	1x7808 do...	1x7808 do...	1x7808 do...	1x7808 do...	1x7808 do...	1x7808 do...	1x7808 do...

Figura 3.72 risultato dell'unione.

### Fase 2: Creazione della matrice di dataset

Successivamente si passa alla riorganizzazione dei dati in una matrice, denominata Dataset\_pompaggio.mat, avente sulle righe le osservazioni e sulle colonne i sensori; si è colta l'occasione per eliminare i sensori poco informativi, di cui si è parlato in precedenza, ed è stata aggiunta una colonna per l'assegnazione dell'etichetta 1 ai dati di pompaggio.

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	27.5229	16.4653	128.4344	167.5401	201.0971	27.7972	119.4781	180.7281	31.0679	31.4998	31.3719
2	1	27.5259	16.4558	128.4851	167.3826	200.9177	27.7983	119.4791	180.7285	31.0680	31.4999	31.3714
3	1	27.5289	16.4463	128.5359	167.2250	200.7383	27.7995	119.4802	180.7288	31.0680	31.4999	31.3709
4	1	27.5319	16.4368	128.5866	167.0675	200.5590	27.8007	119.4812	180.7291	31.0681	31.5000	31.3704
5	1	27.5348	16.4273	128.6373	166.9099	200.3796	27.8019	119.4822	180.7294	31.0682	31.5000	31.3699
6	1	27.5375	16.4199	128.6880	166.7524	200.2002	27.8033	119.4832	180.7297	31.0682	31.5001	31.3695
7	1	27.5385	16.4269	128.6678	166.8027	200.1453	27.8055	119.4843	180.7300	31.0683	31.5002	31.3690
8	1	27.5395	16.4339	128.6465	166.8563	200.0924	27.8078	119.4853	180.7303	31.0684	31.5002	31.3685
9	1	27.5405	16.4408	128.6251	166.9100	200.0395	27.8101	119.4863	180.7307	31.0685	31.5003	31.3680
10	1	27.5414	16.4478	128.6038	166.9637	199.9866	27.8124	119.4873	180.7310	31.0685	31.5004	31.3675
11	1	27.5424	16.4529	128.5825	167.0173	199.9337	27.8144	119.4884	180.7313	31.0686	31.5004	31.3670
12	1	27.5434	16.4486	128.5369	166.9980	199.7695	27.8149	119.4894	180.7316	31.0687	31.5005	31.3665
13	1	27.5444	16.4442	128.4886	166.9706	199.6025	27.8154	119.4904	180.7319	31.0687	31.5005	31.3660
14	1	27.5454	16.4399	128.4403	166.9432	199.4356	27.8159	119.4914	180.7322	31.0688	31.5006	31.3655
15	1	27.5464	16.4356	128.3921	166.9159	199.2686	27.8164	119.4925	180.7325	31.0689	31.5007	31.3650
16	1	27.5474	16.4314	128.3438	166.8885	199.1017	27.8169	119.4935	180.7329	31.0689	31.5007	31.3645
17	1	27.5483	16.4289	128.2950	167.0304	198.9347	27.8174	119.4945	180.7332	31.0690	31.5008	31.3641
18	1	27.5493	16.4265	128.2465	167.1859	198.7677	27.8179	119.4955	180.7335	31.0691	31.5009	31.3636
19	1	27.5503	16.4241	127.8940	167.3413	198.6008	27.8184	119.4965	180.7338	31.0692	31.5009	31.3631
20	1	27.5513	16.4216	127.7415	167.4968	198.4339	27.8189	119.4975	180.7341	31.0692	31.5010	31.3626
21	1	27.5523	16.4187	127.5890	167.6522	198.2670	27.8194	119.4986	180.7344	31.0693	31.5010	31.3621

Figura 3.73 Dataset\_pompaggio.mat.

Gli stessi passaggi sono stati effettuati anche per i dati di pompaggio.

### Fase 3: Bilanciamento del dataset

Nella figura di seguito il codice sviluppato in MATLAB per il bilanciamento del dataset. In particolare, l'algoritmo procede all'estrazione casuale di un numero di dati pari a quelli contenuti nella classe meno popolosa (quella di pompaggio) dai dati di regime, in modo da ottenere un bilanciamento perfetto.

```

1 - load('Valori_regime.mat');
2 - k=19198;
3 - Estrazione_regime=cell(16,30);
4 - Indice_estrazione=cell(16,1);
5 - for i=1:size(Valori_regime,1)
6 -     for j=1:size(Valori_regime,2)
7 -         if j==1
8 -             [Estrazione_regime{i,j},idx]=datasample(Valori_regime{i,j},k,'Replace',false);
9 -         else
10 -            Estrazione_regime{i,j}=Valori_regime{i,j}(idx);
11 -         end
12 -     end
13 -     Indice_estrazione{i,1}=idx;
14 - end

```

Figura 3.74 codice di estrazione casuale.

	1	2	3	4	5	6	7	8	9	10	11	12
1	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...
2	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...
3	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...
4	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...
5	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...
6	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...
7	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...
8	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...
9	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...
10	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...
11	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...
12	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...
13	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...
14	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...
15	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...
16	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...	1x19198 da...

Figura 3.75 risultato dell'estrazione.

Analogamente a quanto fatto per i dati di pompaggio, si è effettuata una riorganizzazione dei dati in una matrice, denominata Dataset\_regime.mat, ed ai quali è stata assegnata l'etichetta 0.

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	30.7453	15.4456	178.1674	233.7285	222.5411	32.2169	125.2323	185.5860	36.7675	37.6308	34.8800
2	0	28.1127	15.4795	172.8628	230.5332	218.1168	30.5973	124.7674	184.2008	35.6139	36.4502	33.8495
3	0	28.0145	15.4812	170.9108	230.6444	218.6584	30.6860	124.8141	184.4061	35.8242	36.6625	34.0576
4	0	31.0175	15.3830	180.4958	237.3178	226.3779	32.5349	124.9123	184.8339	37.8786	38.7991	35.8211
5	0	29.5369	15.4837	171.9185	230.6343	219.2129	31.0673	124.8457	184.8028	35.8537	36.7076	34.1406
6	0	27.3285	15.5106	169.9540	228.6059	214.6642	30.2079	124.8939	184.5083	34.9490	35.8127	33.5124
7	0	30.4563	15.4003	181.1268	235.7551	223.5652	31.9947	124.8507	184.6687	37.2674	38.1924	34.9349
8	0	27.9514	15.4513	175.5940	231.0964	218.9824	30.6833	124.7778	184.3433	35.6744	36.5479	33.9023
9	0	27.3214	15.4504	179.4102	229.2249	216.7762	30.2077	124.8955	184.5084	34.9522	35.8218	33.5172
10	0	28.9599	15.4865	175.7496	229.8011	217.4964	30.6122	124.8723	184.6240	35.3996	36.2496	33.6335
11	0	30.3433	15.4479	179.8859	232.3732	219.7567	31.8025	125.1677	185.4488	36.3004	37.1712	34.4152
12	0	29.7729	15.4249	173.4757	231.2262	220.0504	31.3970	124.9405	184.7299	36.2445	37.1484	34.1892
13	0	31.1116	15.4301	185.5874	235.7524	224.8314	32.5672	125.1436	185.3139	37.6259	38.5035	35.7094
14	0	29.2314	15.4827	173.3072	229.0301	218.7602	30.8666	124.9408	184.7008	35.5515	36.4110	33.7855
15	0	30.6845	15.4040	186.8409	236.9563	223.6346	32.1905	124.8709	184.8703	37.3026	38.2296	34.9651
16	0	30.3004	15.5565	180.5277	230.3254	219.0305	31.7765	125.2475	185.4856	36.0803	36.9517	34.2013
17	0	30.1872	15.4961	179.9137	233.1408	218.8648	31.6619	125.1607	185.4085	36.2013	37.0168	34.3141
18	0	27.6071	15.4907	172.2474	229.7400	217.8906	30.4149	124.7883	184.2692	35.3863	36.2576	33.6940
19	0	30.7578	15.4009	181.1831	233.8364	224.2199	32.3321	124.8576	184.6709	37.3442	38.2395	35.2548
20	0	30.0111	15.4852	182.9343	231.7127	219.5886	31.4595	124.9822	184.9378	36.1239	36.9262	34.2633
21	0	29.8083	15.5029	175.6850	230.9072	219.2424	31.2796	124.9011	184.8594	36.0396	36.8757	34.2418

Figura 3.76 Dataset\_regime.mat.

Infine, i due dataset sono stati uniti per creare un `training_set` contenente i dati relativi ad entrambe le classi.

### 3.7.1.8 Selezione delle variabili – Neighborhood Component Analysis

Il dataset risultante è composto da 27 variabili, dimensione già molto inferiore rispetto alla situazione di partenza, per effetto della semplificazione delle ridondanze e dell'eliminazione dei sensori poco informativi, ma comunque troppo elevata per poter avere una apprezzabile comprensibilità del problema. Per questo motivo, è stata determinata l'importanza relativa di ciascuna variabile, in modo tale da poter eventualmente considerarne un sottoinsieme nella fase di classificazione.

La NCA ha portato alla definizione di pesi alle varie caratteristiche, e ha messo in evidenza la presenza di grandezze irrilevanti, o perché non discriminanti della condizione di funzionamento del compressore, o perché ridondanti rispetto ad altre misure. La NCA è stata condotta per mezzo della funzione `MATLAB® fscnca`, che è stata applicata sul dataset, in primo luogo in una fase di *cross-validation* e, in seguito, per la determinazione dei pesi.

La fase di *cross-validation* è stata eseguita per determinare il valore ottimale da assegnare al parametro di regolarizzazione  $\lambda$ : attraverso una validazione di tipo 4-fold, si è verificato che il valore di default del parametro (reciproco del numero di osservazioni) è già di per sé ottimale, in quanto porta a un valore nullo della funzione di perdita.

Infine, la determinazione dei pesi è stata realizzata imponendo alla funzione *fscnca* il valore di default di  $\lambda$  e la standardizzazione delle variabili, data la differenza di scala.

```

1 - load('Dataset.mat');
2 - X=(Dataset(:,2:28));
3 - Y=(Dataset(:,1));
4 - rng(1); % For reproducibility
5 - cvp = cvpartition(Y,'holdout',0.25);
6 - Xtrain = X(cvp.training,:);
7 - Ytrain = Y(cvp.training,:);
8 - Xtest = X(cvp.test,:);
9 - Ytest = Y(cvp.test,:);
10 - nca = fscnca(Xtrain,Ytrain,'Standardize',true);
11 - L = loss(nca,Xtest,Ytest);
12 - nca_all=fscnca(X,Y,'Standardize',true);

```

Figura 3.77 codice di NCA.

Il risultato della NCA è riassunto nell'immagine e nella tabella seguenti:

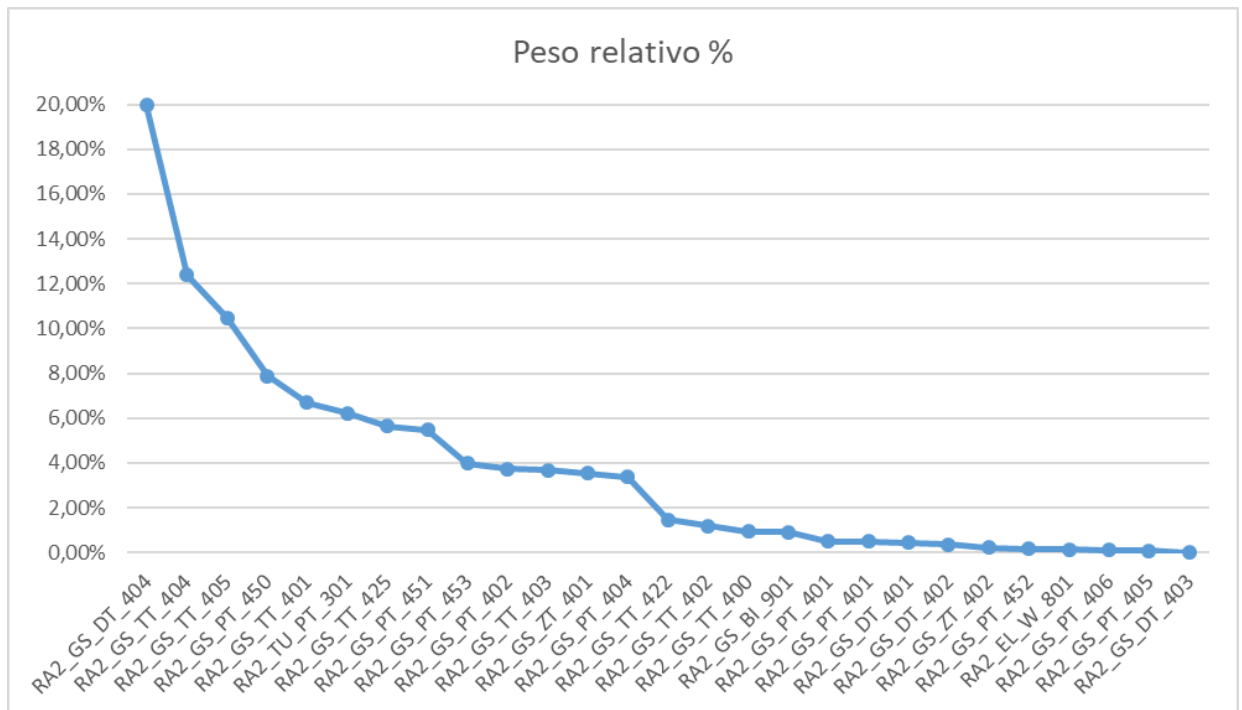


Figura 3.78 Risultato NCA: pesi delle variabili considerate.

Tabella 3.10 pesi delle variabili.

#	Sensore	Peso	Peso relativo %	Peso relativo cumulato %
1	RA2_GS_DT_404	8,38	19,95%	19,95%

2	RA2_GS_TT_404	5,20	12,39%	32,34%
3	RA2_GS_TT_405	4,40	10,48%	42,82%
4	RA2_GS_PT_450	3,31	7,88%	50,70%
5	RA2_GS_TT_401	2,81	6,68%	57,38%
6	RA2_TU_PT_301	2,61	6,21%	63,59%
7	RA2_GS_TT_425	2,37	5,65%	69,23%
8	RA2_GS_PT_451	2,29	5,46%	74,70%
9	RA2_GS_PT_453	1,67	3,98%	78,67%
10	RA2_GS_PT_402	1,57	3,74%	82,41%
11	RA2_GS_TT_403	1,55	3,68%	86,09%
12	RA2_GS_ZT_401	1,48	3,53%	89,62%
13	RA2_GS_PT_404	1,41	3,37%	92,99%
14	RA2_GS_TT_422	0,61	1,46%	94,45%
15	RA2_GS_TT_402	0,50	1,20%	95,65%
16	RA2_GS_TT_400	0,39	0,94%	96,59%
17	RA2_GS_BI_901	0,38	0,92%	97,51%
18	RA2_GS_PT_401	0,20	0,49%	97,99%
19	RA2_GS_PT_401	0,20	0,49%	98,48%
20	RA2_GS_DT_401	0,18	0,44%	98,92%
21	RA2_GS_DT_402	0,15	0,35%	99,27%
22	RA2_GS_ZT_402	0,09	0,22%	99,49%
23	RA2_GS_PT_452	0,07	0,16%	99,65%
24	RA2_EL_W_801	0,05	0,13%	99,78%
25	RA2_GS_PT_406	0,05	0,12%	99,90%
26	RA2_GS_PT_405	0,04	0,09%	99,99%
27	RA2_GS_DT_403	0,01	0,01%	100,00%

Si può osservare come le prime 7 variabili abbiano un peso superiore ai 2/3 del totale, mentre le ultime forniscano un contributo del tutto marginale ai fini della classificazione.

### 3.7.1.9 Validazione del modello diagnostico - SVM

Note le variabili più rilevanti, è stato possibile costruire dei modelli di classificazione per la separazione dei dati di pompaggio dai dati di regime. Si è deciso di escludere da questa fase le variabili relative alla



posizione delle valvole di regolazione (RA2\_GS\_ZT\_401 e RA2\_GS\_ZT\_402), in quanto queste non esprimono le caratteristiche fisiche del fluido, ma sono i parametri di controllo su cui si agisce a livello operativo per evitare di incorrere nell'instabilità della macchina. Inoltre, si può osservare che, dall'analisi NCA, non risultano essere variabili di grande rilevanza. Tutte le operazioni di calcolo sono state effettuate con un pc dotato di 4.00 GB di RAM e di un processore Intel® Core™ i7-6500U con frequenza di base pari a 2.50 GHz.

La classificazione di tipo SVM è stata realizzata attraverso l'utilizzo del *Classification Learner* di MATLAB®. In primo luogo, sono stati costruiti vari modelli SVM, di cui si è valutato l'accuratezza di previsione.

Dato l'output della NCA, il primo tentativo di addestramento ha coinvolto le sette variabili di maggior peso, di cui sono state estratte le osservazioni dal training-set creato da utilizzare per l'addestramento del modello SVM. Questa prima fase di addestramento ha permesso di definire il tipo ed i parametri del kernel più appropriati per il tipo di analisi

**Tabella 3.11**                      **variabili considerate nel primo tentativo di addestramento SVM.**

<b>Sensore</b>	<b>Descrizione</b>
RA2_GS_DT_404	PRESSIONE DIFFRENZIALE DEMISTER RG
RA2_GS_TT_404	TEMPERATURA GAS ASPIRAZIONE 3° STADIO SX
RA2_GS_TT_405	TEMPERATURA GAS ASPIRAZIONE AP LATO MOLTIPLICATORE
RA2_GS_PT_450	PRESSIONE INTERSTADIO ESTRATTORE GAS
RA2_GS_TT_401	TEMPERATURA ASPIRAZIONE GAS BP LATO OPPOSTO MOLTIPLICATORE
RA2_TU_PT_301	PRESSIONE ASSOLUTA CONDENSATORE
RA2_GS_TT_425	TEMPERATURA GAS USCITA STADIO AP

Sono stati provati tre modelli, tra quelli proposti dal Classification Learner, su cui si è effettuato l'addestramento con cross-validation di tipo *5-folds* e la successiva valutazione dell'accuratezza:

1. Kernel lineare,
2. Kernel gaussiano medio,
3. Kernel gaussiano fine.

Il modello con kernel lineare è stato provato per la maggior semplicità di trattazione che garantisce rispetto ai modelli non lineari. Tuttavia, la prova non ha dato esito positivo, in quanto l'accuratezza sul set di test è risultata piuttosto bassa. Inoltre, l'applicazione MATLAB® ha mostrato difficoltà nella ricerca dei

coefficienti ottimali dell'iperpiano di separazione, manifestando una evidente lentezza di calcolo, che ha comportato un addestramento di durata superiore a un giorno.

```
Model 1: Trained

Results
Accuracy          57.8%
Prediction speed  ~3000 obs/sec
Training time     99718 sec

Model Type
Preset: Linear SVM
Kernel function: Linear
Kernel scale: Automatic
Box constraint level: 1
Multiclass method: One-vs-One
Standardize data: true

Feature Selection
All features used in the model, before PCA

PCA
PCA disabled
```

Figura 3.79 risultati dell'addestramento del modello con kernel lineare sul dataset composto da 7 variabili.

Il kernel gaussiano, utilizzando per l'addestramento l'app dedicata di MATLAB, può essere grezzo, medio o fine e si differenzia per il valore del kernel scale, che è un parametro direttamente proporzionale al raggio  $\sigma$ . Si ha un incremento del kernel scale se si passa dal kernel fine a quello grezzo, e ciò comporta la definizione di una funzione gaussiana che considera come appartenenti alla stessa classe punti a distanza sempre maggiore. Da ciò consegue che la scelta del kernel scale ha un impatto sull'accuratezza del classificatore, col rischio di ottenere *underfitting*, per valori del parametro elevati, o *overfitting*, in caso contrario.

Il modello con kernel gaussiano medio proposto dal *Classification Learner* ha un kernel scale pari a  $2.65(\sqrt{p})$ , con  $p$  pari al numero di variabili), e l'addestramento sul dataset ha portato ai seguenti risultati:

```
Model 1: Trained  
  
Results  
Accuracy          99.8%  
Prediction speed  ~12000 obs/sec  
Training time     1098.7 sec  
  
Model Type  
Preset: Medium Gaussian SVM  
Kernel function: Gaussian  
Kernel scale: 2.65  
Box constraint level: 1  
Multiclass method: One-vs-One  
Standardize data: true  
  
Feature Selection  
All features used in the model, before PCA  
  
PCA  
PCA disabled
```

Figura 3.80 risultati dell'addestramento del modello con kernel gaussiano medio sul dataset composto da 7 variabili

Il modello con kernel gaussiano fine ha un kernel scale pari a  $0.66 \left(\frac{\sqrt{p}}{4}\right)$  e l'addestramento sul dataset ha portato ai seguenti risultati:

```
Model 2: Trained  
  
Results  
Accuracy          100.0%  
Prediction speed  ~91000 obs/sec  
Training time     237.36 sec  
  
Model Type  
Preset: Fine Gaussian SVM  
Kernel function: Gaussian  
Kernel scale: 0.66  
Box constraint level: 1  
Multiclass method: One-vs-One  
Standardize data: true  
  
Feature Selection  
All features used in the model, before PCA  
  
PCA  
PCA disabled
```

Figura 3.81 risultati dell'addestramento del modello con kernel gaussiano fine sul dataset composto da 7 variabili

Si può osservare che entrambi i modelli hanno fornito ottimi risultati in termini di accuratezza, riuscendo a classificare in maniera corretta la quasi totalità delle osservazioni del set di test. Il modello con *kernel* gaussiano fine risulta più performante, non solo in termini di accuratezza, ma anche in termini di velocità di addestramento e di previsione. Queste caratteristiche lo rendono il modello preferibile, poiché, in

un'ottica di implementazione *real-time* del sistema di classificazione, è importante non solo la validità del risultato di classificazione, ma anche la rapidità con cui un nuovo punto di funzionamento viene etichettato, così da poter riconoscere il prima possibile l'evoluzione di una situazione di pompaggio del compressore.

Data la tendenza delle prestazioni dei modelli SVM al variare del *kernel scale*, si è deciso di non mettere alla prova il modello grezzo, e si è approfondito la valutazione delle prestazioni del modello con kernel gaussiano fine, per irrobustirne la credibilità. Questa analisi è stata realizzata attraverso un'estensione del set di test, che ha riguardato in primo luogo i dati di regime e poi i dati di pompaggio.

Per verificare la validità del modello nella classificazione dei dati di regime, è stata calcolata l'accuratezza di previsione del classificatore su un dataset composto da tutte le osservazioni escluse nella fase di bilanciamento. Questa attività ha richiesto l'utilizzo della funzione MATLAB® *predictFcn*, che richiede in input il modello di classificazione col quale fare la previsione, e in output fornisce l'etichetta associata a ciascuna osservazione del dataset. Essendo il dataset composto da dati di regime, con etichetta pari a 0, l'accuratezza è stata calcolata nel seguente modo:

$$ACC_r = 1 - \frac{\# \text{ etichette con valore 1}}{\# \text{ totale di osservazioni}} = 1 - \frac{6.238}{17.813.692} = 99,96\%$$

Una volta valutato quale fosse il tipo di kernel da utilizzare sono state analizzate le prestazioni dei modelli variando il numero di parametri di input da 4 a 9 aggiungendo via via il successivo a peso più alto in base ai risultati dell'analisi NCA. Nella tabella seguente i risultati dell'analisi svolta:

**Tabella 3.12** Performance dei modelli SVM testati.

Numero variabili	Kernel Scale	Accuracy [%]	Prediction speed [obs/s]	Training time [s]
4	0,5	99,8	28000	432,42
5	0,56	100	85000	236,85
6	0,61	100	85000	264,4
7	0,66	100	91000	237,36
8	0,71	100	100000	228,45
9	0,75	100	93000	241,36

Come mostra la tabella, il modello a 4 variabili è l'unico a mostrare un'accuratezza inferiore al 100% e per questo è stato scartato. Inoltre, notiamo come la velocità di previsione, caratteristica fondamentale per il funzionamento *real-time* del modello SVM, ed il tempo di addestramento siano rispettivamente massimi e minimi per i modelli a 7, 8 e 9 variabili che sono quindi quelli su cui abbiamo svolto ulteriori analisi.

L'accuratezza del 100% dei modelli costruiti suggerisce un perfetto potere di classificazione della condizione attuale del compressore tra "regime" e "pompaggio". La tecnica di validazione del modello (*5-fold cross validation*) agisce suddividendo in maniera casuale il dataset in 5 parti, dopo di che ne utilizza 4 per l'addestramento ed una per verificare le previsioni con i valori effettivi. In questo modo, nella parte di dataset utilizzata per il training del modello saranno con alta probabilità contenuti dati relativi ad ognuno degli 11 eventi di pompaggio utilizzati. Per verificare la potenza classificatoria dei modelli SVM, a 7,8 e 9 variabili, simulando l'avvento di un nuovo evento di guasto è stata utilizzata la seguente procedura:

1. Eliminazione di un file di dati di pompaggio dal dataset di addestramento,
2. Addestramento del modello con kernel gaussiano fine sul dataset risultante,
3. Calcolo dell'accuratezza di previsione sul set composto dai dati esclusi, quindi sui soli dati di pompaggio.

Tale procedura è stata ripetuta, variando a ogni iterazione il file di dati trascurato, e variando oltretutto il numero di parametri del modello per trovare il miglior modello in termini sia di accuratezza che di performance. Essendo 11 gli eventi di pompaggio registrati a database, sono stati costruiti altrettanti modelli SVM, ognuno con la propria accuratezza di previsione sulle osservazioni trascurate nella fase di addestramento. Infine, è stata calcolata una accuratezza generale, che valutasse la capacità del modello di classificare correttamente i dati di pompaggio, come valore pesato delle 11 accuratezze sulla dimensione del set di test:

$$ACC_{pi} = 1 - \frac{\# \text{ etichette con valore } 0}{\# \text{ totale di osservazioni}}$$

$$ACC_p = \frac{\sum_{i=1}^{11} ACC_{pi} \cdot l_i}{\text{dimensione dataset di pompaggio}}$$

dove  $ACC_{pi}$  è l'accuratezza del modello valutata sull' $i$ -esimo file trascurato, e  $l_i$  è il numero di osservazioni che compongono tale file. Nella tabella seguente i risultati di accuratezza ottenuti da questa procedura di validazione.

**Tabella 3.13** Accuratezza previsionale sui singoli eventi di pompaggio.

Evento di pompaggio	Modello 7	Modello 8	Modello 9
1	44,10%	47,89%	59,60%
2	63,03%	67,84%	57,42%
3	68,67%	72,12%	74,36%

4	100,00%	100,00%	100,00%
5	48,47%	50,07%	52,17%
6	100,00%	100,00%	100,00%
7	39,83%	57,07%	84,45%
8	26,16%	38,89%	60,65%
9	92,58%	97,54%	98,03%
10	84,29%	77,80%	76,16%
11	0,00%	0,00%	0,00%
Accuratezza totale	60,65%	64,47%	69,35%
<b>REGIME</b>			
Accuratezza	99,96%	99,96%	99,96%

Come mostrato in tabella per tutti i modelli testati, le accuratezze sui vari eventi di pompaggio varia dal 100% al 26,16% e tutti mostrano un'accuratezza nulla per l'evento di pompaggio numero 11 che quindi non sarebbe stato riconosciuto dal modello SVM. Aggiungendo però i dati relativi a questo evento all'addestramento dell'SVM, nel caso il fenomeno di pompaggio si manifesti di nuovo con modalità simili sarà a questo punto riconosciuto dal modello. Valutando l'accuratezza totale, il modello a 9 variabili sembra essere il migliore per livello di accuratezza sui singoli fenomeni di pompaggio. L'accuratezza nel riconoscimento della classe di regime è invece invariata per tutti i modelli testati e pari al 99,96%. Per capire meglio quale sia il modello con le performance di accuratezza maggiore si riporta la seguente figura:

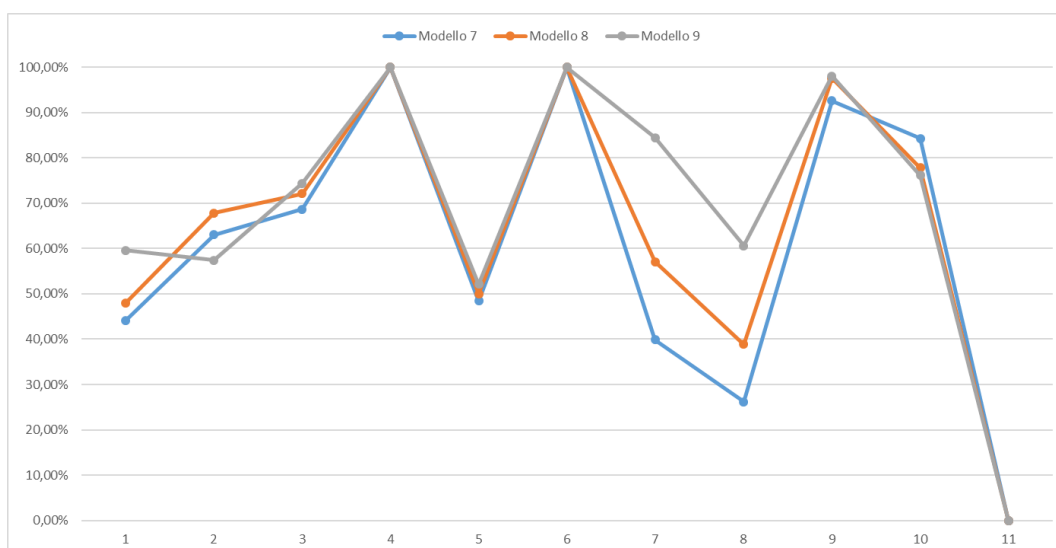


Figura 3.82 Accuratezza previsionale sui singoli eventi di pompaggio.

Come conferma anche la Figura 3.82 il modello a 9 variabili risulta essere più performante in termini di accuratezza per quasi tutti gli eventi di pompaggio analizzati ed è stato quindi scelto per essere implementato.

In definitiva le variabili da utilizzare per monitorare in real-time il compressore multistadio sono le seguenti:

**Tabella 3.14** Variabili necessarie per la prognostica del compressore.

Sensore	Descrizione
RA2_GS_DT_404	PRESSIONE DIFFRENZIALE DEMISTER RG
RA2_GS_TT_404	TEMPERATURA GAS ASPIRAZIONE 3° STADIO SX
RA2_GS_TT_405	TEMPERATURA GAS ASPIRAZIONE AP LATO MOLTIPLICATORE
RA2_GS_PT_450	PRESSIONE INTERSTADIO ESTRATTORE GAS
RA2_GS_TT_401	TEMPERATURA ASPIRAZIONE GAS BP LATO OPPOSTO MOLTIPLICATORE
RA2_TU_PT_301	PRESSIONE ASSOLUTA CONDENSATORE
RA2_GS_TT_425	TEMPERATURA GAS USCITA STADIO AP
RA2_GS_PT_451	PRESSIONE INTERSTADIO ESTRATTORE GAS
RA2_GS_PT_453	PRESSIONE INTERSTADIO ESTRATTORE GAS

### 3.7.1.10 Validazione del modello diagnostico

Oltre al calcolo dell'accuratezza del modello diagnostico sviluppato, si è cercato di valutare l'errore di prima e seconda specie commessi nel particolare caso studio simulandone il comportamento in real-time sui dati storici a disposizione.

Per quanto riguarda l'errore di seconda specie, definito come la probabilità che il modello effettui una errata segnalazione del fenomeno di pompaggio quando questo si trova in condizioni di regime, in prima analisi questa sarà uguale al complemento ad 1 dell'accuratezza del modello nella classe di regime, ovvero pari allo 0,04%.

Analizzando i secondi di permanenza nella classe di pompaggio nonostante il comportamento del compressore fosse in realtà a regime si ha:

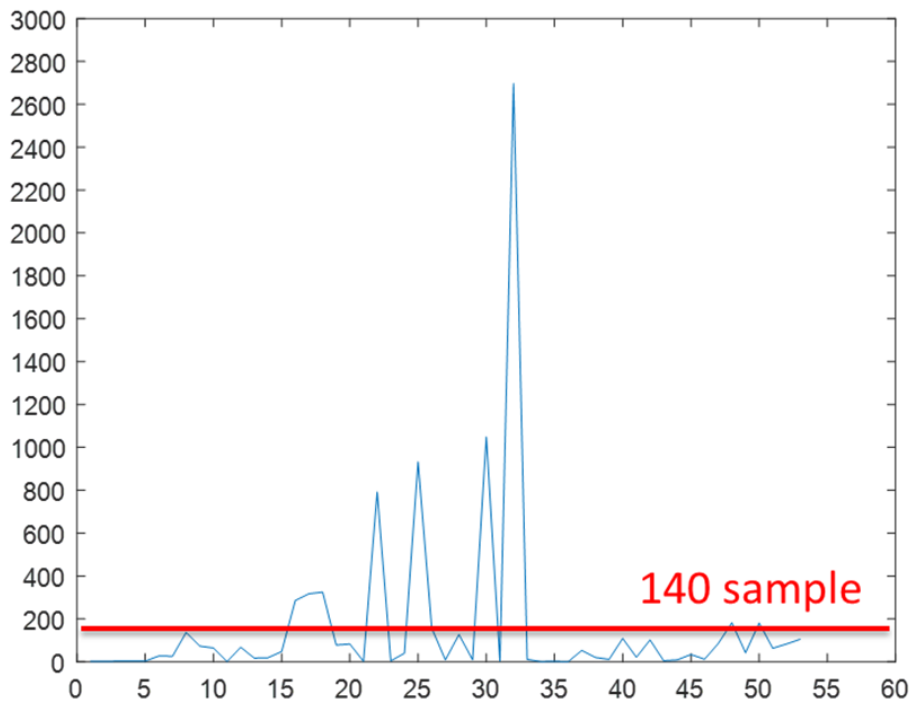


Figura 3.83 tempi di permanenza nella classe del pompaggio di dati relativi alle condizioni di funzionamento a regime.

Si evidenziano quindi 53 errori del classificatore nella classe di regime, su un periodo temporale di circa 2 anni e solo in 8 situazioni si ha una permanenza per un intervallo di tempo superiore a 140 secondi.

Impostando quindi come soglia per la segnalazione di un incipiente evento di pompaggio, la permanenza in quella classe per un tempo maggiore di 140 secondi, il classificatore mostra un errore di seconda specie pari ad 8 volte ogni 2 anni ovvero si avrà mediamente un'errata segnalazione di pompaggio una volta ogni 3 mesi.

Analizzando allo stesso modo anche la classe di regime, si evidenzia come anche nel caso dell'ultima validazione ad eventi di pompaggio effettuata, una logica di allarme di permanenza nella classe per 140s fa sì che tutti gli eventi di pompaggio tranne 1 (l'undicesimo valutato) siano riconosciuti dal classificatore mostrando quindi un errore di prima specie (mancata segnalazione dell'evento di pompaggio) del 9%.

Analizzando invece il modello completo, addestrato su tutti gli eventi di pompaggio e sul dataset di regime, questo mostra un errore sia di prima che di seconda specie nullo, senza dover impostare nessuna logica di segnalazione del guasto, dato che l'accuratezza del modello è pari al 100%.



### 3.7.2 - -Modello diagnostico – riconoscimento di forme d’onda (ShapeFind)

Per irrobustire il modello diagnostico presentato è stato affiancato anche un algoritmo più semplice che richieda un basso sforzo computazionale. Il requisito principale che si richiede all’algoritmo è la semplicità teorica, così da poter comprendere e studiare le risposte non banali dell’algoritmo stesso ai dati di verifica. Successivamente, una volta verificata l’efficacia della “versione base”, saranno possibili perfezionamenti, o adattamenti a utilizzi specifici.

Lo strumento generato dall’algoritmo è un riconoscitore di forma d’onda, ed è stato sviluppato dal Dottor Marco Strambi, ex assegnista di ricerca del Dipartimento di Ingegneria Industriale dell’Università degli Studi di Firenze che mi ha affiancato durante tutto lo svolgimento del progetto SmartGEO.

La teoria alla base del modello è stata affrontata nel paragrafo 2.5.3 -.

#### 3.7.2.1 Verifica dell’algoritmo

L’algoritmo è stato applicato ad un segnale di pressione proveniente dal sensore RA2\_GS\_PT\_401 (identificato dalla descrizione “PRESSIONE GAS ASPIRAZIONE BP”), per un totale di circa 20 milioni di dati, corrispondenti ad un periodo di circa 8 mesi in totale, campionato uniformemente con periodo pari ad 1 Hz. Il segnale stesso è stato suddiviso in tre tipologie di evento per un totale di 46 periodi:

1. Anomalie generiche (periodi 1-17)
2. Innesco di pompaggio o pompaggio sviluppato (periodi 18-30)
3. Situazione di regime (periodi 30-46)

La suddivisione è stata operata manualmente con l’aiuto del personale esperto dell’impianto caso studio, quindi potrebbe essere imprecisa; in particolare ci possiamo aspettare che:

- Durante le anomalie, per quanto non in pompaggio, possano essere presenti comunque dei segnali di innesco;
- Molto più raramente, anche durante i periodi di regime sono possibili indicazioni di innesco

Nel segnale registrato dal sensore scelto, il pompaggio si presenta come una forma d'onda pseudo-periodica di ampiezza limitata (Figura 3.84) e di periodo dell'ordine dei minuti, mentre l'innesco (incipiente pompaggio) presenta forme d'onda simili, ma sporadiche (Figura 3.85). Il funzionamento a regime è soggetto alla variazione della pressione dovuta ai cicli di temperatura giorno-notte (Figura 3.86). L'algoritmo ha un comportamento corretto se riesce a indicare sia i pompaggi che gli inneschi, senza

individuare altre forme d'onda dovute, ad esempio, al blocco della centrale, o comunque qualsiasi altro tipo di malfunzionamento non imputabile al pompaggio (periodi di anomalia generica, ed esempio Figura 3.87 e Figura 3.88), né segnalando firme durante i periodi di regime, con eventuali eccezioni dovute alle imprecisioni nella suddivisione dei periodi.

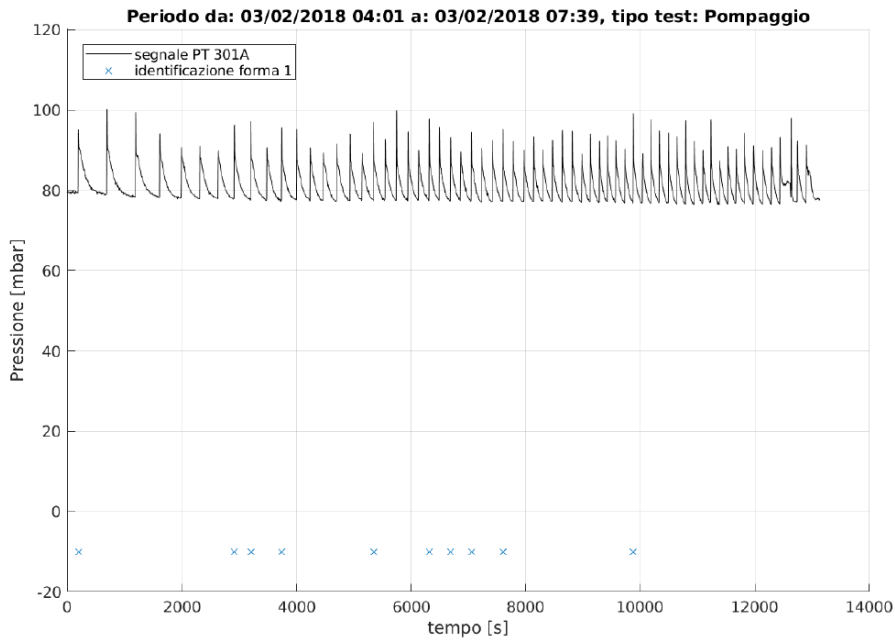


Figura 3.84 forma d'onda pseudo-periodica in condizioni di pompaggio.

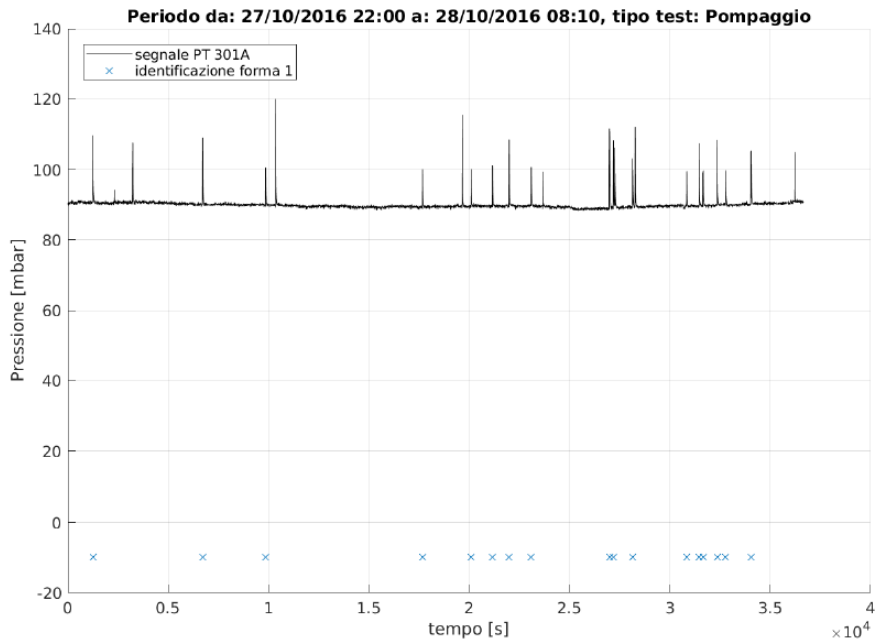


Figura 3.85 forma d'onda sporadica in condizioni di incipiente pompaggio.

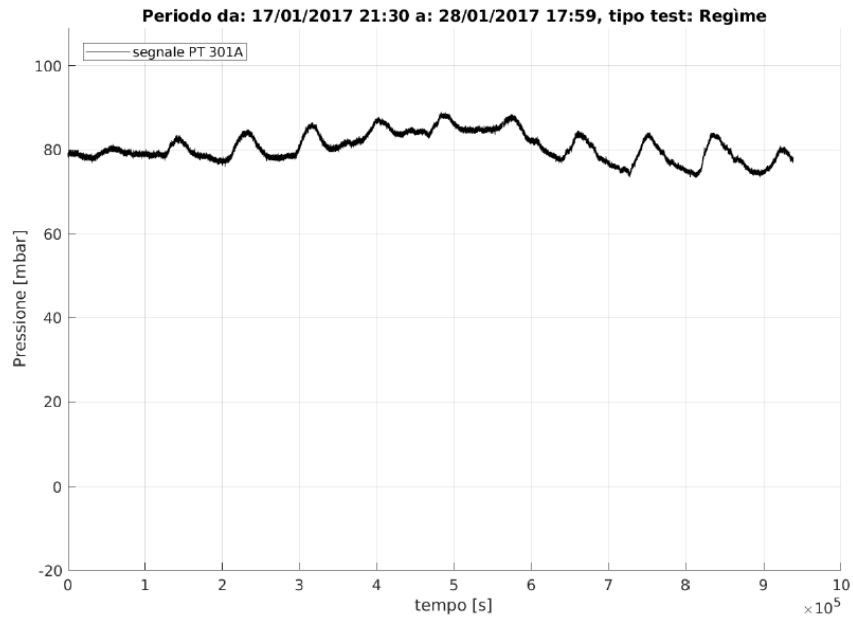


Figura 3.86 cicli giornalieri di pressione in condizioni di regime.

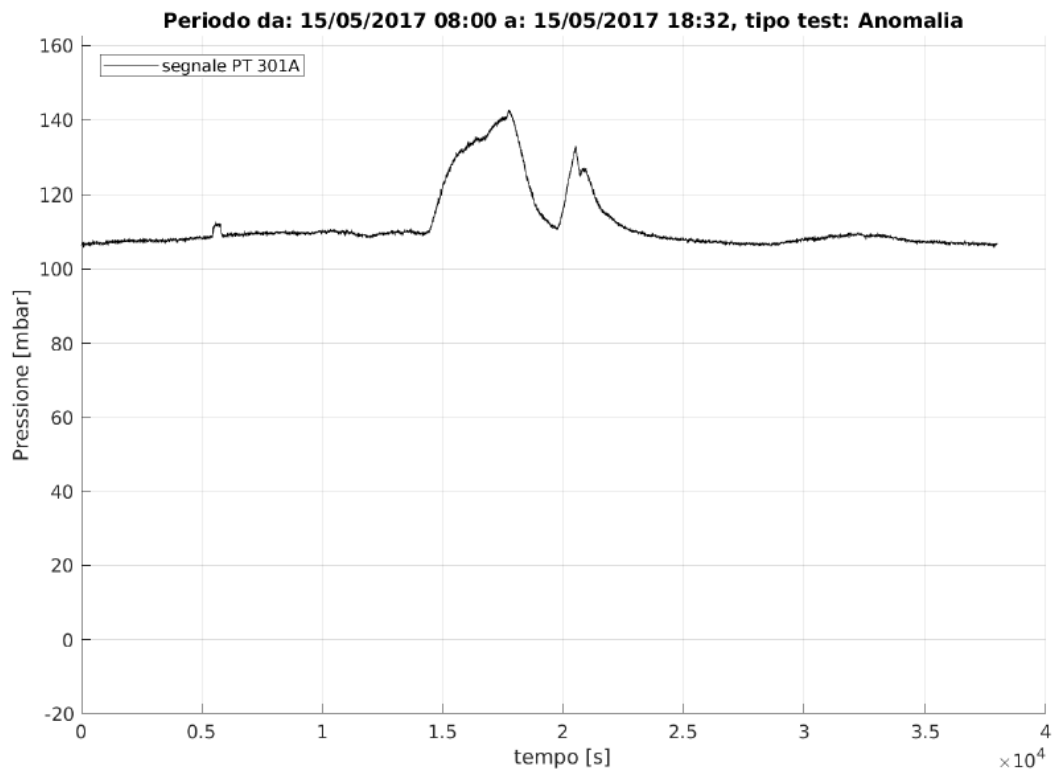


Figura 3.87 esempio di anomalia.

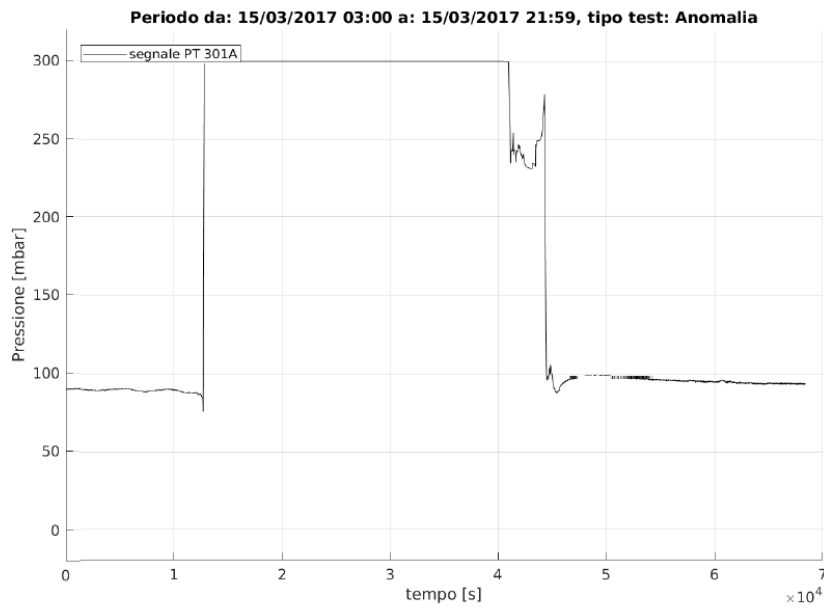


Figura 3.88 esempio di anomalia dovuta al blocco della centrale.

Come input dell'algoritmo sono state scelte quattro famiglie di parametri. Comune a tutte queste famiglie è la scelta dell'intervallo di picco-picco accettabile per il segnale da confrontare (tra 3 e 100 mbar). Per ciascuna famiglia di parametri sono state scelte una o due firme, ciascuna delle quali della durata di circa 100 s.

**Due firme:**

Le due firme scelte in input sono state rappresentate in Figura 3.89 e Figura 3.90

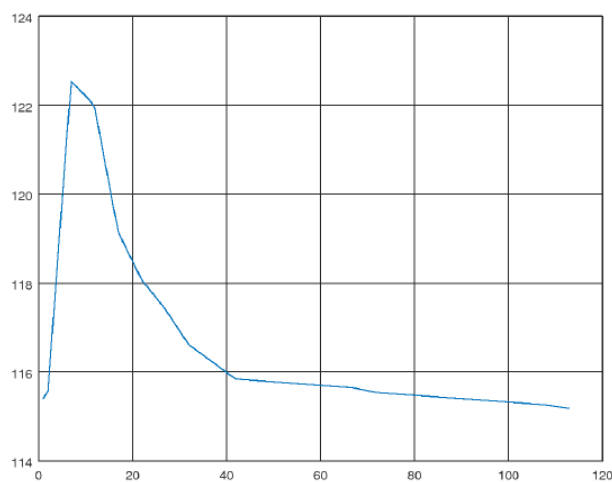


Figura 3.89 Firma tipica.

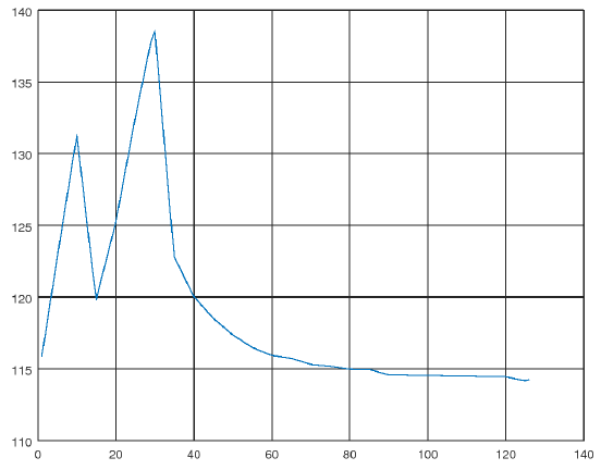


Figura 3.90 Firma bicornio.

L'insieme  $L$  scelto è il seguente:

$$L = \{0.2, 0.4, \dots, 1.8, 2, 3, 4, \dots, 20\}$$

e la soglia per la distanza quadratica è stata scelta pari a 0,07.

**Firma inversa:**

Per questo insieme di parametri, la firma in input (Figura 3.91) è stata scelta per poter individuare periodi di segnale pseudo periodici, contenenti una fase esponenziale discendente seguita da una crescita repentina. Con questa scelta si vuol tentare di isolare le risposte al pompaggio (fase periodica) dall'innesco (fase sporadica).

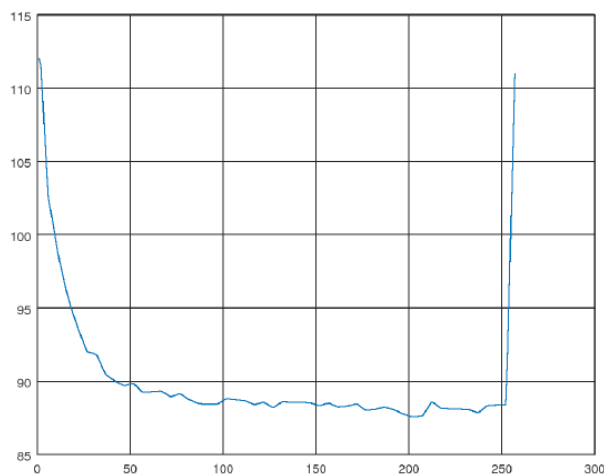


Figura 3.91 Firma inversa.

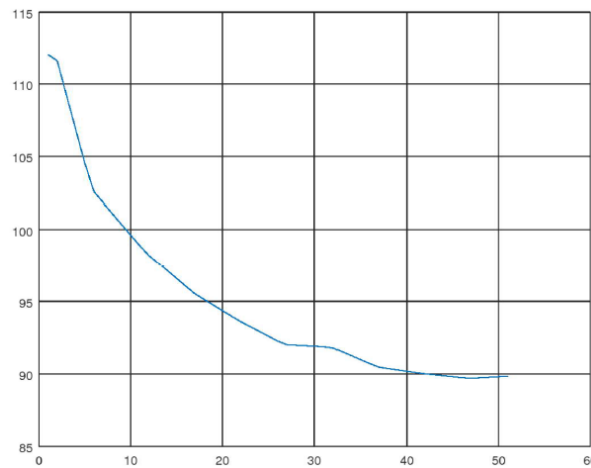
L'insieme  $L$  scelto è il seguente:

$$L=\{0.1,0.2,\dots,1.9,2,3,4,\dots,30\}$$

e la soglia per la distanza quadratica è stata scelta pari a 0.07.

**Firma discendente:**

La firma scelta è quella rappresentata in Figura 3.92



**Figura 3.92 Firma discendente esponenziale.**

L'insieme  $L$  scelto è il seguente:

$$L=\{0.1,0.2,\dots,1.9,2,3,4,\dots,30\}$$

e la soglia per la distanza quadratica è stata scelta pari a 0.07.

**Firma discendente, distanza ridotta:**

È stata scelta la stessa firma discendente esponenziale mostrata in Figura 3.92, tuttavia è stata impostata una nuova distanza massima pari a 0,025, così da limitare falsi positivi.

L'insieme  $L$  scelto è il seguente:

$$L=\{0.1,0.2,\dots,1.9,2,3,4,\dots,30\}$$

### 3.7.2.2 Prestazioni dell'algoritmo

L'algoritmo, per come è strutturato, ha prestazione lineare nei seguenti parametri:

- Lunghezza del segnale di input
- Numero totale di firme
- Somma di tutti gli elementi di L

Per valutarne le prestazioni computazionali è stato implementato e verificato su un PC portatile dalle prestazioni medie, completando ciascuna delle 4 analisi in circa un'ora, quindi è in grado di essere eseguito real-time in condizioni ragionevoli senza particolari ottimizzazioni.

### 3.7.2.3 Riconoscimento firme

Di seguito delle tabelle riepilogative dei risultati ottenuti:

Tabella 3.15 Periodi di regime o di anomalia, insieme di parametri "Due Firme".

id. periodo	tipo periodo	durata [s]	firme riconosciute (tipo 1, tipo 2)
1	Anomalia	107393	14 (11, 3)
2	Anomalia	51596	0
3	Anomalia	30771	0
4	Anomalia	281992	0
5	Anomalia	68399	0
6	Anomalia	37968	0
7	Anomalia	539386	38 (36, 2)
8	Anomalia	70187	0
9	Anomalia	82195	4 (4, 0)
10	Anomalia	8875	2 (2, 0)
11	Anomalia	249771	0
12	Anomalia	49791	0
13	Anomalia	52197	0
14	Anomalia	41579	0
15	Anomalia	57593	1 (1, 0)
16	Anomalia	33841	0
17	Anomalia	143390	1 (1, 0)
31	Regime	68997	1 (1, 0)
32	Regime	730791	0
33	Regime	321998	0
34	Regime	622797	0
35	Regime	887207	0
36	Regime	937799	0
37	Regime	1370399	0
38	Regime	1929655	1 (1, 0)
39	Regime	595795	0
40	Regime	803396	1 (1, 0)
41	Regime	2263009	0
42	Regime	2213399	0
43	Regime	687535	3 (3, 0)
44	Regime	1693615	0
45	Regime	1771200	0
46	Regime	1427998	0

Tabella 3.16 Periodi di pompaggio, insieme di parametri "Due Firme".

id. periodo	durata	firme riconosciute
18	13128	10 (10, 0)
19	13978	11 (8, 3)
20	7197	1 (1, 0)
21	64798	6 (6, 0)
22	8875	2 (2, 0)
23	39599	4 (3, 1)
24	43197	8 (7, 1)
25	36652	17 (17, 0)
26	41634	120 (89, 31)
27	35395	20 (15, 5)
28	43194	0
29	29995	67 (48, 19)
30	51225	94 (61, 33)



Tabella 3.17 Periodi di regime o di anomalia, insieme di parametri "Firma inversa".

id. periodo	tipo periodo	durata [s]	firme riconosciute (tipo 1, tipo 2)
1	Anomalia	107393	55
2	Anomalia	51596	0
3	Anomalia	30771	0
4	Anomalia	281992	1
5	Anomalia	68399	0
6	Anomalia	37968	0
7	Anomalia	539386	158
8	Anomalia	70187	19
9	Anomalia	82195	16
10	Anomalia	8875	0
11	Anomalia	249771	0
12	Anomalia	49791	1
13	Anomalia	52197	1
14	Anomalia	41579	1
15	Anomalia	57593	1
16	Anomalia	33841	1
17	Anomalia	143390	17
31	Regime	68997	1
32	Regime	730791	0
33	Regime	321998	0
34	Regime	622797	0
35	Regime	887207	1
36	Regime	937799	0
37	Regime	1370399	0
38	Regime	1929655	1
39	Regime	595795	0
40	Regime	803396	5
41	Regime	2263009	0
42	Regime	2213399	0
43	Regime	687535	17
44	Regime	1693615	0
45	Regime	1771200	0
46	Regime	1427998	2

Tabella 3.18 Periodi di pompaggio, insieme di parametri "Firma inversa".

id. periodo	durata	firme riconosciute
18	13128	4
19	13978	35
20	7197	17
21	64798	18
22	8875	16
23	39599	28
24	43197	51
25	36652	17
26	41634	91
27	35395	27
28	43194	9
29	29995	59
30	51225	64

Tabella 3.19 Periodi di regime o di anomalia, insieme di parametri "Firma discendente".

id. periodo	tipo periodo	durata [s]	firme riconosciute (tipo 1, tipo 2)
1	Anomalia	107393	165
2	Anomalia	51596	9
3	Anomalia	30771	1
4	Anomalia	281992	16
5	Anomalia	68399	7
6	Anomalia	37968	0
7	Anomalia	539386	344
8	Anomalia	70187	3
9	Anomalia	82195	51
10	Anomalia	8875	18
11	Anomalia	249771	13
12	Anomalia	49791	1
13	Anomalia	52197	3
14	Anomalia	41579	2
15	Anomalia	57593	241
16	Anomalia	33841	56
17	Anomalia	143390	27
31	Regime	68997	3
32	Regime	730791	0
33	Regime	321998	0
34	Regime	622797	0
35	Regime	887207	1
36	Regime	937799	0
37	Regime	1370399	2
38	Regime	1929655	26
39	Regime	595795	0
40	Regime	803396	8
41	Regime	2263009	4
42	Regime	2213399	2
43	Regime	687535	42
44	Regime	1693615	1
45	Regime	1771200	2
46	Regime	1427998	3

Tabella 3.20 Periodi di pompaggio, insieme di parametri "Firma discendente".

id. periodo	durata	firme riconosciute
18	13128	62
19	13978	134
20	7197	36
21	64798	25
22	8875	18
23	39599	45
24	43197	80
25	36652	28
26	41634	418
27	35395	44
28	43194	17
29	29995	241
30	51225	699

Tabella 3.21 Periodi di regime o di anomalia, insieme di parametri "Firma discendente, distanza ridotta".

id. periodo	tipo periodo	durata [s]	firme riconosciute (tipo 1, tipo 2)
1	Anomalia	107393	15
2	Anomalia	51596	2
3	Anomalia	30771	1
4	Anomalia	281992	0
5	Anomalia	68399	2
6	Anomalia	37968	0
7	Anomalia	539386	37
8	Anomalia	70187	0
9	Anomalia	82195	9
10	Anomalia	8875	2
11	Anomalia	249771	1
12	Anomalia	49791	0
13	Anomalia	52197	0
14	Anomalia	41579	2
15	Anomalia	57593	29
16	Anomalia	33841	4
17	Anomalia	143390	6
31	Regime	68997	1
32	Regime	730791	0
33	Regime	321998	0
34	Regime	622797	0
35	Regime	887207	0
36	Regime	937799	0
37	Regime	1370399	0
38	Regime	1929655	4
39	Regime	595795	0
40	Regime	803396	0
41	Regime	2263009	1
42	Regime	2213399	0
43	Regime	687535	8
44	Regime	1693615	0
45	Regime	1771200	0
46	Regime	1427998	0

Tabella 3.22 Periodi di pompaggio, insieme di parametri "Firma discendente, distanza ridotta".

id. periodo	durata	firme riconosciute
18	13128	8
19	13978	4
20	7197	3
21	64798	5
22	8875	2
23	39599	7
24	43197	20
25	36652	18
26	41634	230
27	35395	22
28	43194	3
29	29995	116
30	51225	352

L'insieme dei parametri "Firma discendente, distanza ridotta" ottiene dei buoni risultati, soprattutto per quanto riguarda l'identificazione dei pompaggi e la limitazione dei falsi positivi nei periodi di regime, ma la

soluzione migliore risulta dai parametri “due firme”, dove il comportamento dell’algoritmo è molto buono anche nei periodi di anomalia. Prendendo in considerazione tutti i casi considerati, le anomalie n.1, 7 e 9 risultano di difficile identificazione per l’algoritmo (eventuali falsi positivi), così come il periodo a regime n.43 (eventuali falsi positivi), e il pompaggio n.28 (eventuale non segnalazione).

Nelle figure seguenti, che fanno riferimento all’insieme dei parametri “Due firme”, si rappresentano in dettaglio questi insiemi di dati “difficili”. In ciascun grafico, le x blu e le x rosse indicano rispettivamente le porzioni di segnale in cui sono state identificate le firme 1 e le firme 2.

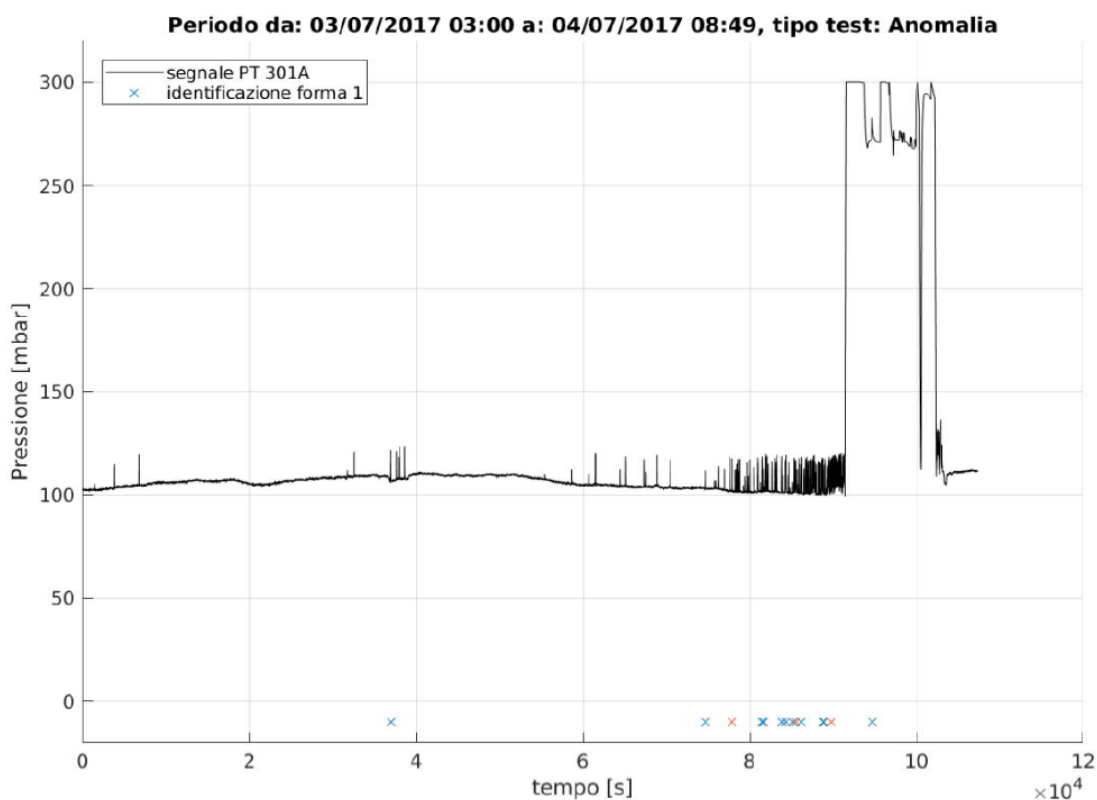


Figura 3.93 Anomalia 1: presenza di segnali di innesco pompaggio.

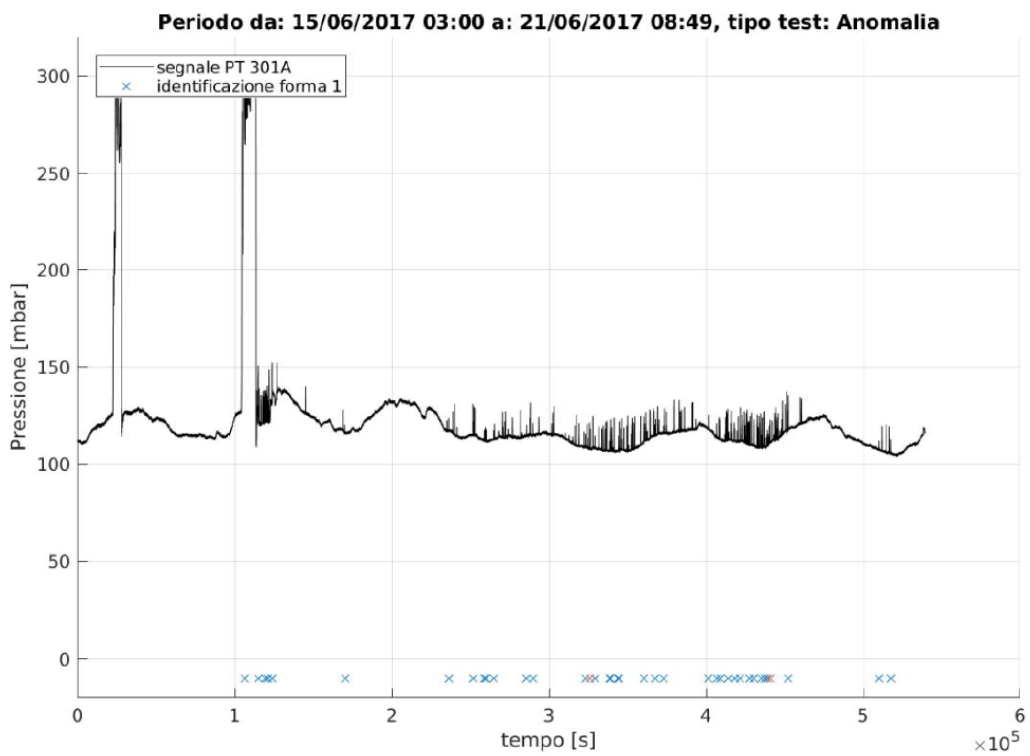


Figura 3.94 Anomalia 7: presenza di segnali di innesco pompaggio.

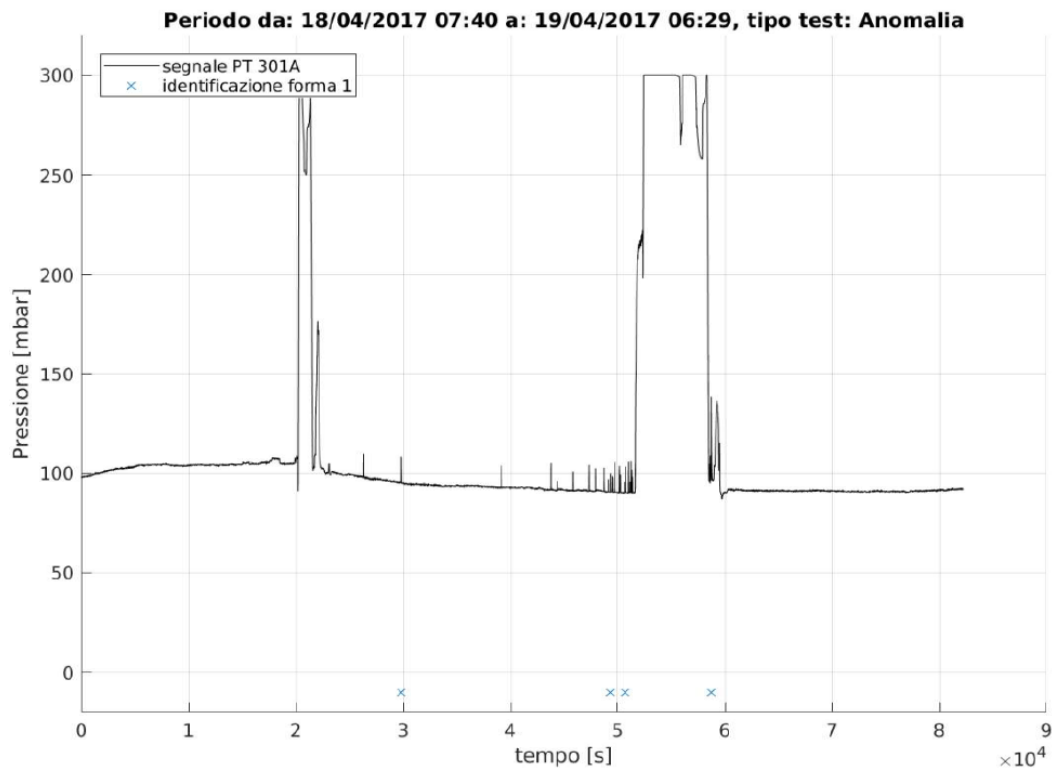


Figura 3.95 Anomalia 9: presenza di segnali di innesco pompaggio.

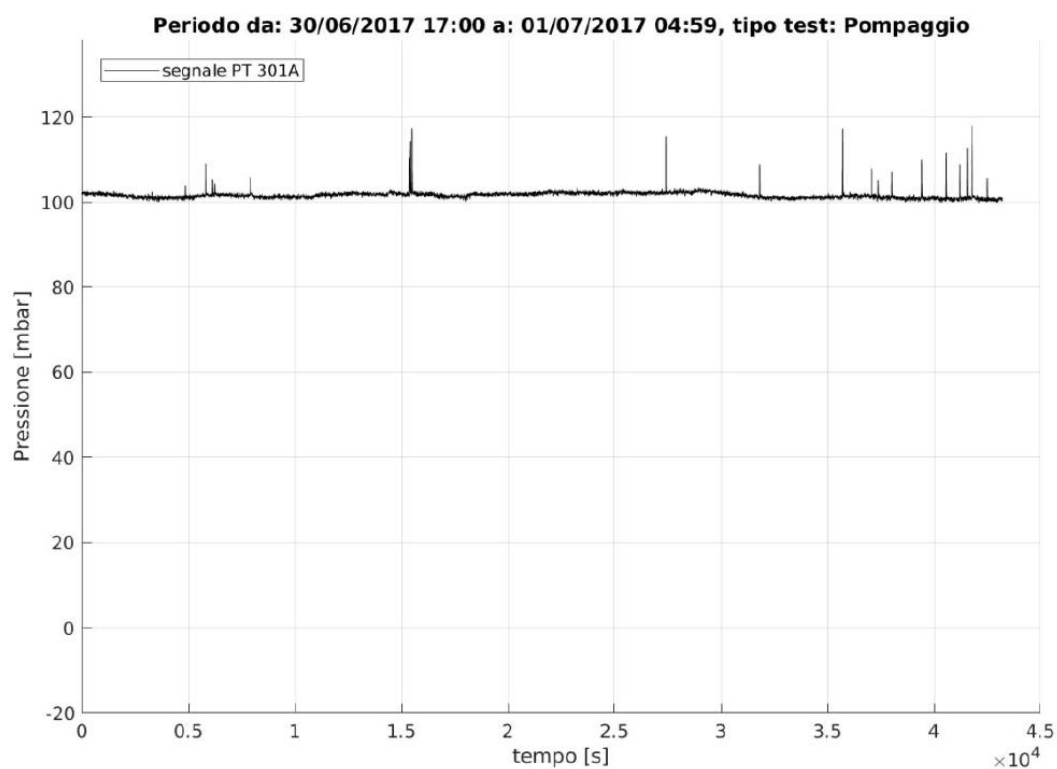


Figura 3.96 Pompaggio 28: Il pompaggio non viene riconosciuto.

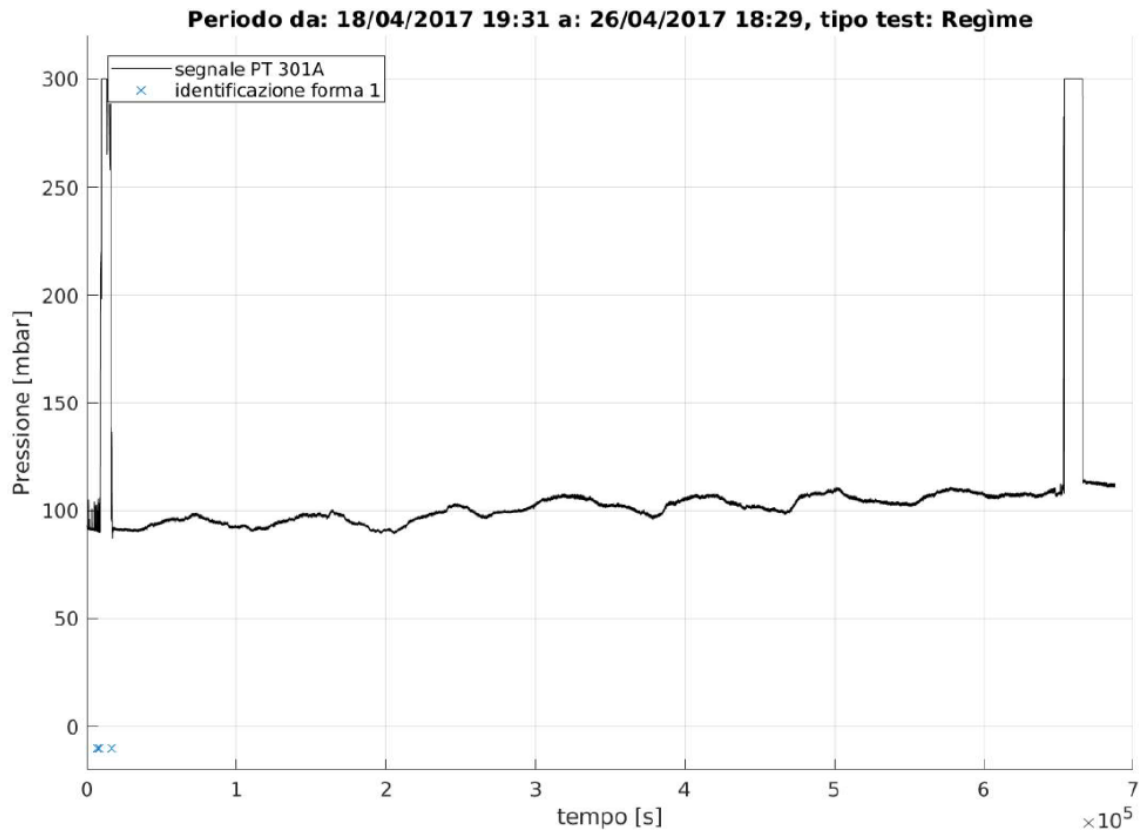


Figura 3.97 Regime 43: presenza di segnali di innesco pompaggio.

Come si può vedere dalle figure, tutti i falsi positivi si possono imputare alla presenza di segnali di innesco pompaggio, tuttavia, per quanto riguarda il periodo di pompaggio 28 (Figura 3.96), l'algoritmo fallisce nell'identificare la forma d'onda. Si può quindi supporre che la forma d'onda in questione sia diversa da quelle utilizzate nei dizionari, e che quindi vada aggiunta in input.

### 3.7.3 - Rilascio dei modelli su PLC Siemens SIMATIC STEP 7

Come descritto anche nel paragrafo 3.2.3 - i PLC acquistati nel progetto per implementare i modelli diagnostici hard-realtime necessitano di una trasformazione del codice MATLAB in progetti Simulink compatibili con la funzione PLC Coder di Simulink al fine di essere rilasciati sui PLC stessi.

#### 3.7.3.1 Limitazioni nell'integrazione Simulink con PLC Siemens

Molti degli algoritmi utilizzati, sia di supporto che finali, hanno necessitato di una traduzione, e spesso una riscrittura, dovuta alle limitazioni imposte dal compilatore PLC Siemens. In particolare, algoritmi scritti in linguaggio m, come nel nostro caso, devono essere soggetti alle seguenti restrizioni:

1. sono accettabili variabili di tipo booleano, virgola mobile, intero, stringa, ma non celle;
2. le variabili possono essere singole, o organizzate in array, ma le matrici presentano limitazioni;
3. la dimensione di un array deve essere conosciuta dall'algoritmo a tempo di compilazione, un array non può quindi essere a dimensione variabile;
4. non è possibile l'utilizzo di molte struct derivate da algoritmi nativi MATLAB;
5. non è possibile effettuare operazioni su array, ma solo su singoli elementi di array;
6. la maggior parte degli algoritmi di alto livello presenti in matlab utilizzano alcune delle feature dei punti precedenti, quindi non sono compatibili con il PLC Siemens.

Queste sono le principali costrizioni che un algoritmo in linguaggio .m deve soddisfare per poter essere compilato in un PLC Siemens, tuttavia esistono altre restrizioni che non sono state indagate e per le quali non esiste comunque documentazione di riferimento.

In particolare, per il punto 2 e 4, le strutture in output delle fasi di allenamento degli algoritmi hanno dovuto essere tradotte in un formato standard di array definito ad hoc e contenente tutte le informazioni necessarie alla fase di esecuzione su dati live. Inoltre, l'algoritmo SVM nativo MATLAB, a causa del punto 6, è stato riscritto completamente con un codice compatibile alle limitazioni imposte.



### 3.7.3.2 Organizzazione generale del progetto Simulink

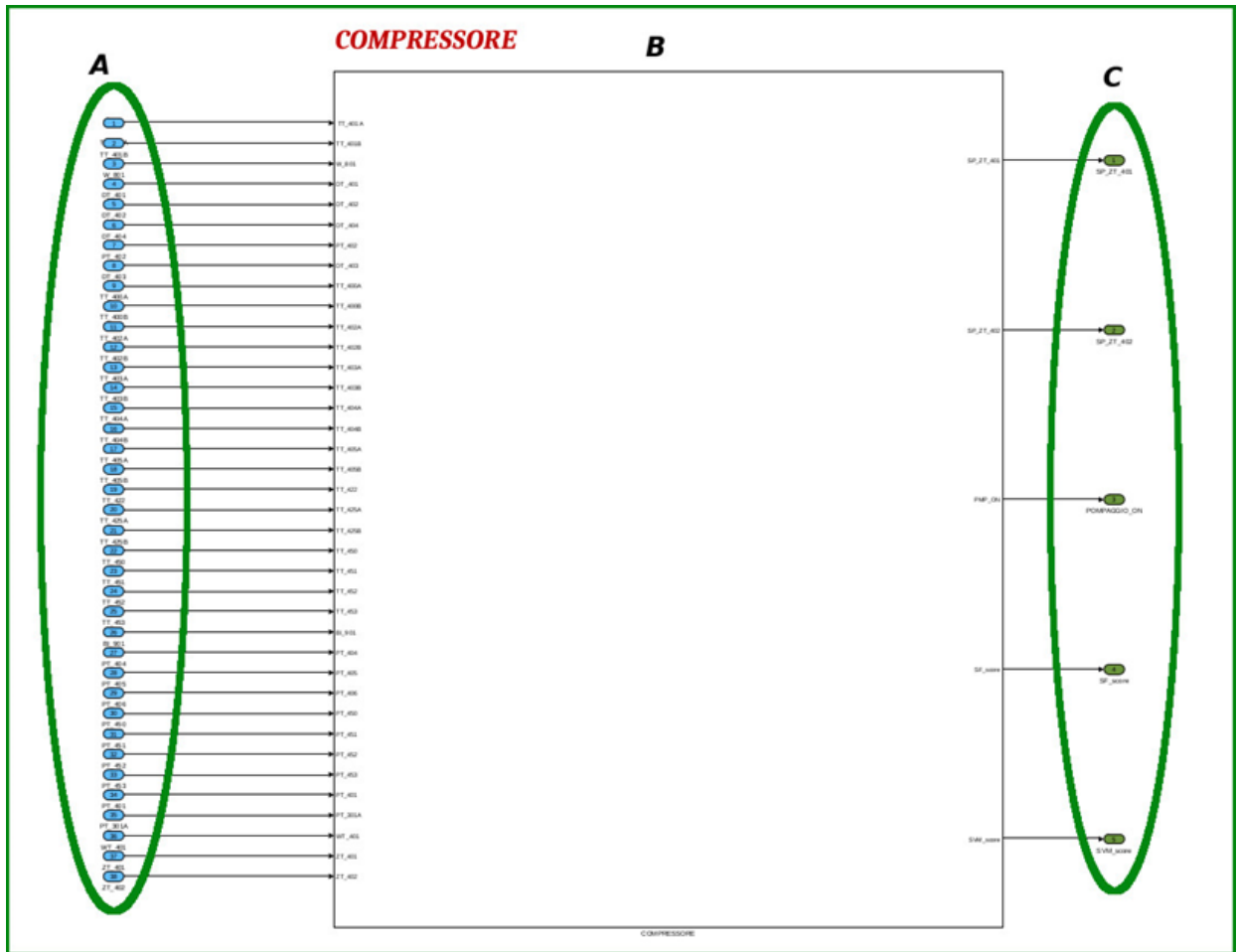


Figura 3.98 Organizzazione generale del progetto Simulink per PLC Siemens.

Come si può vedere dalla Figura 3.98, l'organizzazione del progetto Simulink a livello più alto è estremamente semplice. I sensori (A) sono input della sottostruttura generale (B), comprendente tra le varie cose, anche gli algoritmi sviluppati; dalla struttura B abbiamo quindi in output i parametri di controllo del PLC (C).

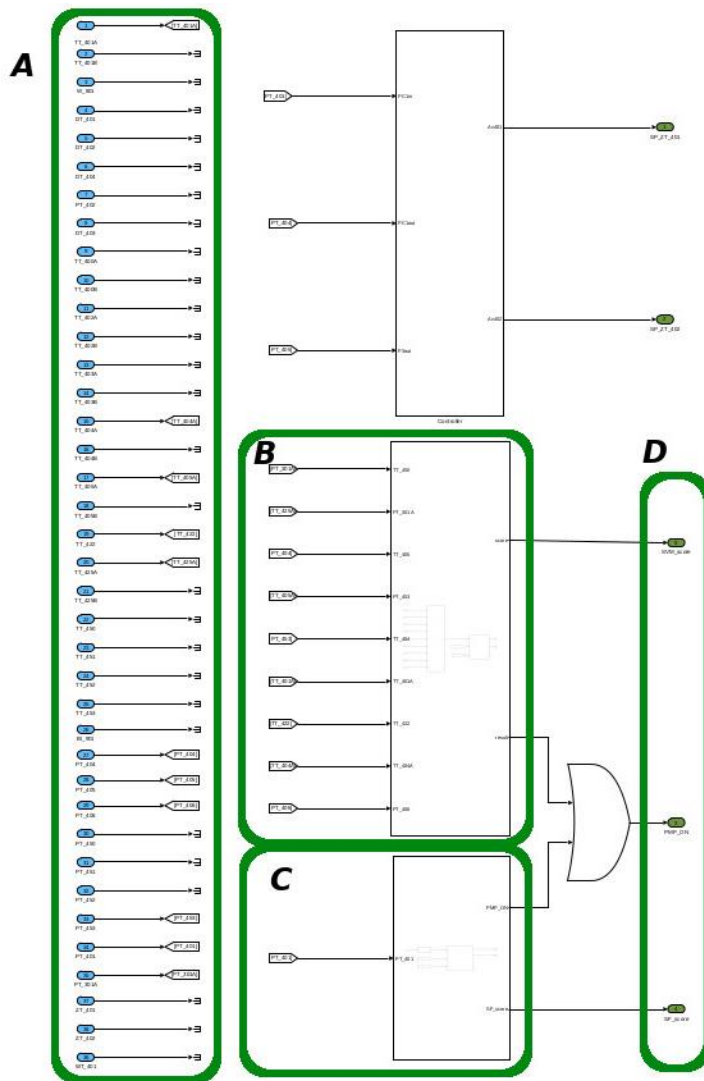


Figura 3.99 Sottostruttura "Compressore", identificazione delle parti di progetto dedicate agli algoritmi sviluppati.

Entrando nel dettaglio della struttura "Compressore" si può notare come la gerarchia si specializzi nelle singole parti dedicate all'algoritmo SVM (B) e Shapefind (C), che ricevono in input alcune delle variabili (A) derivate dai 39 sensori, selezionate tramite studio effettuato in fase di allenamento. Gli output degli algoritmi (D) sono limitati a un segnale di tipo booleano che indica lo stato di pompaggio, e alle variabili di controllo di due valvole.

### 3.7.3.3 Integrazione del modello diagnostico SVM nel progetto

L'algoritmo SVM identifica lo stato di pompaggio tramite i valori correnti di alcuni dei sensori in input.

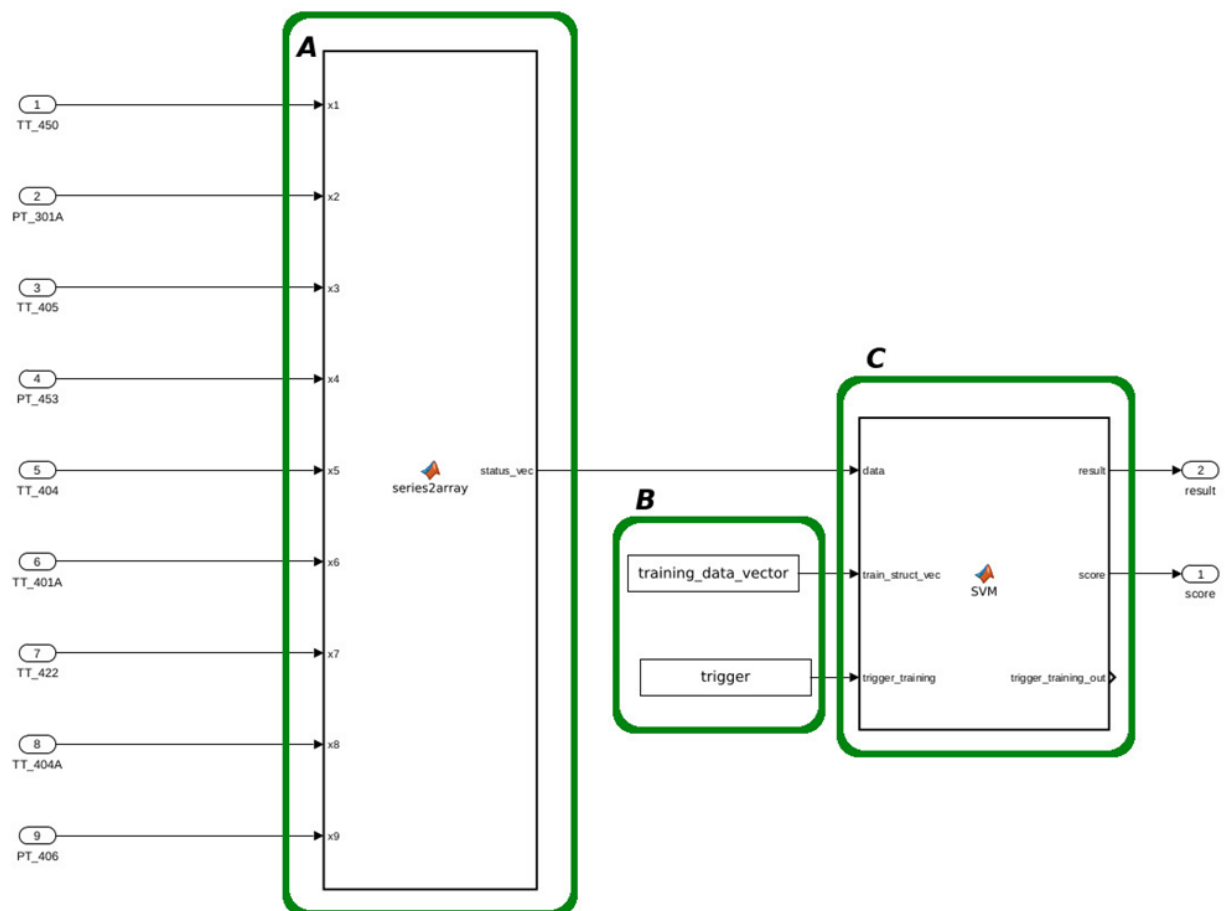


Figura 3.100 Struttura dell'algoritmo SVM.

In figura, sulla sinistra, i sensori identificati come significativi sono immessi nell'algoritmo di supporto (A) che li riorganizza come array e li comunica all'algoritmo SVM (C).

Oltre allo stato corrente dei sensori significativi, gli altri input dell'algoritmo (B) sono:

- una variabile (`training_data_vector`) contenente le informazioni della struttura ottenuta dalla fase di allenamento su dati storici e riorganizzata sotto forma di array a lunghezza fissa (per compatibilità PLC Siemens) tramite un algoritmo di supporto sviluppato ma non presente nel progetto;
- un indicatore booleano (`trigger`) forza l'algoritmo SVM alla lettura dei parametri di allenamento.

Gli output dell'algoritmo sono:

- un indicatore booleano (`result`) indicante lo stato di pompaggio
- un valore reale compreso tra 0 e 1 di valutazione quantitativa dello stato di pompaggio (`score`)

L'algoritmo SVM è presente nelle librerie native Matlab, tuttavia in tale formato risulta incompatibile con le limitazioni imposte dal PLC Siemens, quindi è stata necessaria una re-implementazione ad un più basso livello, evitando le tecniche di programmazione e le strutture non consentite.

La formula utilizzata per calcolare la valutazione quantitativa, o score, dello stato di pompaggio è la seguente:

$$x_{i,j} = k \left( \left( \frac{d_j - \mu_j}{\sigma_j} \right) - V_{i,j} \right)$$
$$s = b + \sum_{i=1}^l \alpha_i \theta_i e^{-|x_i|^2}$$

dove:

$d_j$  è il valore corrente del j-esimo sensore

$\mu_j$  e  $\sigma_j$  indicano i valori medi e le deviazioni standard per il j-esimo sensore

$V_{i,j}$  è il valore dell'i-esimo *Support Vector* relativo al sensore j

$k$  è un fattore di scala

$x_i$  è il vettore costituito dalla i-esima riga della matrice  $x_{i,j}$

$b$  è un valore di scostamento iniziale

$l$  è il numero di *Support Vector*

$\alpha_i$  è il parametro di peso specifico di ciascun *Support Vector*

$\theta_i$  è l'etichetta di ciascun *Support Vector*, e vale 1 se il *Support Vector* specifico è relativo a una fase di pompaggio, -1 altrimenti

$s$  è lo score risultante

I parametri  $m, l, b, \alpha, \theta, \mu, \sigma$  e  $V$  sono calcolati in fase di allenamento e passati all'algoritmo tramite la variabile "training\_data\_vector".

Se lo score  $s$  risulta superiore ad una soglia definita in fase di allenamento, l'algoritmo segnala lo stato di pompaggio effettivo imponendo alla variabile result il valore true.

### 3.7.3.4 Integrazione del modello diagnostico riconoscimento di forme d'onda (ShapeFind)

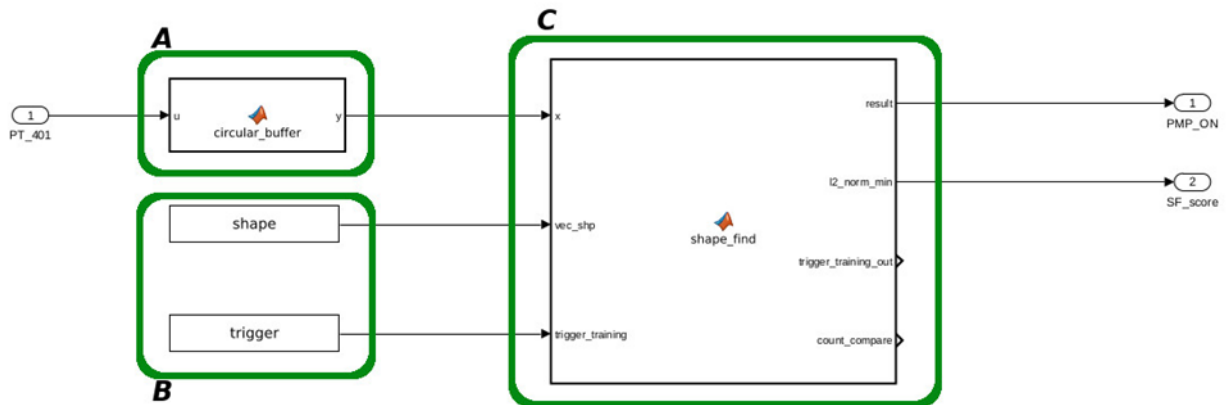


Figura 3.101 Struttura dell'algoritmo ShapeFind.

La struttura che integra l'algoritmo *Shapefind* è simile a quella utilizzata per l'SVM, ma a differenza di quest'ultimo, *Shapefind* necessita della storia continua di un singolo sensore per effettuare la propria analisi. Il sensore in questione viene immesso in un buffer circolare e cumulato in un *array* che quindi ne conterrà la storia fino al momento corrente. L'algoritmo di supporto "*circular\_buffer*" (A) si prende carico di questo compito prima di comunicare l'intero *array* all'algoritmo *Shapefind* (C) come input.

Come per SVM, anche *shapefind* ha altri due input (B): un *array* contenente le "firme" identificate in fase di allenamento degli stati storicamente riconosciuti come pompaggio, e un parametro booleano che forza la lettura dell'*array* di allenamento.

Anche gli output di *shapefind* sono simili a quelli SVM, e consistono in un indicatore booleano di stato di pompaggio (PMP\_ON), e di una valutazione dello stesso con un numero reale compreso tra 0 e 1 (SF\_score).

### 3.7.3.5 Migrazione su PLC di diagnostica hard-realtime

Entrambi gli algoritmi, modellati come blocchi *Simulink* con la logica appena descritta saranno poi importati ed utilizzati a fini diagnostici sul PLC Siemens per controllo del compressore.

## 3.8 - Modello di stima della RUL in assenza di dati di guasto

La metodologia elaborata e presentata nel paragrafo 2.3 - è stata applicata al caso studio ed in particolare all'unico modo di guasto di cui avevamo a disposizione sia dati relativi al comportamento a regime che nell'intera evoluzione fino al momento del guasto. In questo modo è stato possibile confrontare i risultati previsionali del modello con quelli effettivamente registrati al fine di validare la metodologia e le analitiche proposte.

Per il caso studio, abbiamo messo a confronto due diversi modelli di *state detection*:

1. Carte di controllo multivariate o carte di Hotelling;
2. *One class SVM*.

Sugli output di questi modelli è stato utilizzato un modello di analisi di serie temporali ARIMA per procedere con la stima della RUL e valutare le performance dei due modelli costruiti.

L'analisi dei dati storici del compressore assieme al personale specializzato dell'impianto caso studio, ci ha permesso di selezionare i dati relativi al funzionamento a regime del macchinario per l'addestramento dei modelli di *state detection* e di utilizzare i dati relativi al fenomeno di pompaggio per verificare che questi diano un valore di performance del compressore significativamente al di sotto di quelle utilizzate per l'addestramento del modello. L'applicazione del modello di analisi di serie temporali, ci permette inoltre di valutare l'accuratezza della stima della RUL dei due modelli.

Data la natura del macchinario sono stati selezionati tutti i sensori relativi alle condizioni termofluidodinamiche (pressione, temperatura e portata) del gas lavorato in corrispondenza di ingresso ed uscita di tutti gli stadi di compressione per un totale di 16 variabili di controllo correlate alle performance del compressore. Il periodo storico analizzato, corrispondente a circa 8 giorni di acquisizione dati con campionamento ad 1 secondo, è stato suddiviso insieme al personale di impianto in 3 differenti periodi:

1. Regime;
2. Anomalia;
3. Pompaggio.

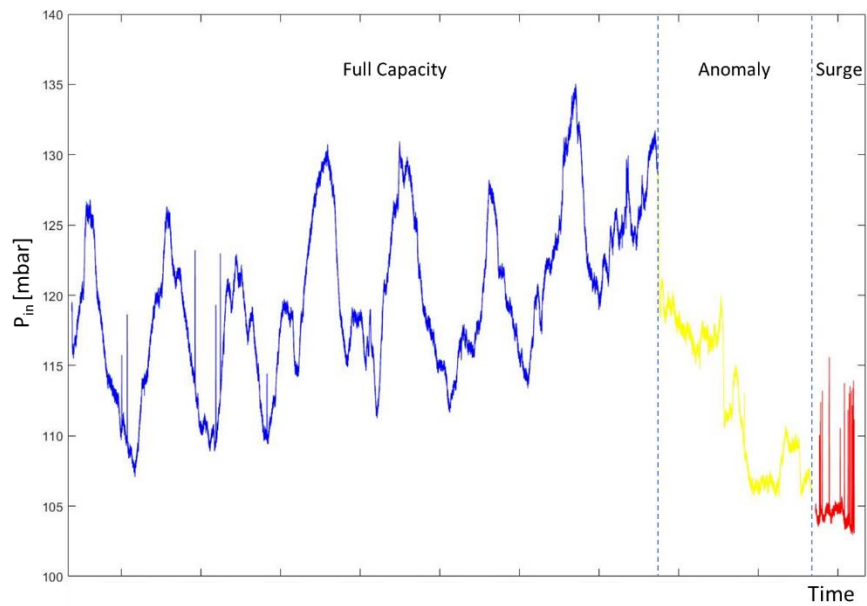


Figura 3.102 Esempio di divisione delle condizioni operative del macchinario.

I dati relativi al periodo di regime contano 635782 elementi per ognuna delle 16 variabili di controllo, quelli relativi al periodo di funzionamento anomalo 166473 e quelli del periodo di pompaggio 42103. La grande quantità di dati nel periodo di regime, utilizzato per addestrare i due modelli di controllo statistico di processo ne garantisce la robustezza.

### 3.8.1 - Applicazione delle carte di controllo multivariate

Sui dati di regime è stata costruita la carta di controllo multivariata utilizzando la funzione “*pca*” del software Matlab tra i cui output compare la statistica  $T^2$ , variabile di controllo correlata alle prestazioni del compressore. Il modello così addestrato è stato poi applicato anche ai dati del periodo di anomalia e di pompaggio. Data l’elevato rumore della statistica  $T^2$ , si è deciso di effettuarne una media mobile con *bucket* temporale pari a 300 secondi in modo da rendere più stabile la previsione ARIMA e più robusto il controllo statistico del processo. In Figura 3.103 è riportato l’andamento della media mobile della statistica  $T^2$  nel tempo con il relativo limite superiore di controllo (UCL) pari a 25,06 con confidenza del 95% e limite inferiore di controllo relativo al periodo di pompaggio (SCL) pari a 77,36 con confidenza del 5%:

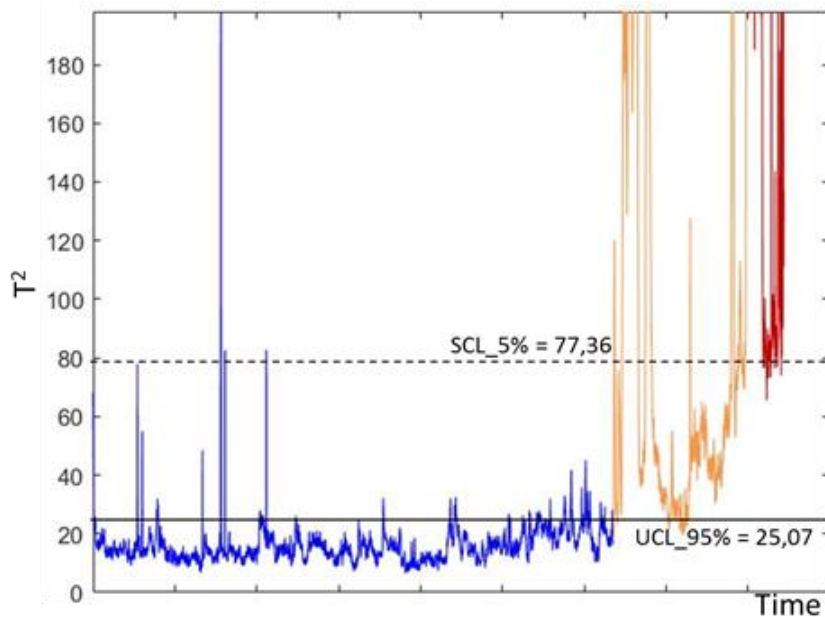


Figura 3.103 Evoluzione nel tempo della media mobile della variabile di controllo  $T^2$ .

Dalla Figura 3.103 Evoluzione nel tempo della media mobile della variabile di controllo  $T^2$ , risulta evidente che i valori di  $T^2$  relativi al periodo contrassegnato come pompaggio (in rosso) sono nettamente superiori al limite superiore di controllo, in particolare il valore minimo di  $T^2$  nel periodo di pompaggio risulta pari a 65,61 ed il valor medio pari a 190,66. L'errore di prima specie del modello, ovvero la mancata segnalazione di pompaggio è quindi nullo per il periodo selezionato. Avendo scelto un limite superiore di controllo pari al 95%, l'errore di seconda specie, ovvero l'errata segnalazione di funzionamenti regolari come anomali risulta pari al 5%. Per ottenere anche un errore di seconda specie nullo è necessario utilizzare come logica di segnalazione del funzionamento anomalo del compressore la permanenza al di sopra della soglia UCL per un tempo maggiore di 2587 secondi (circa 43 minuti). Utilizzando questa logica il modello è in grado di segnalare una anomalia di funzionamento 45,34 ore prima dell'insorgenza del modo di guasto in esame.

### 3.8.2 - Applicazione del One-class SVM

Come nel precedente caso il modello è stato addestrato sui dati storici relativi al periodo di funzionamento di regime. Per l'addestramento è stata utilizzata la funzione *fitcsvm* di Matlab con kernel di tipo gaussiano e ottimizzazione automatica del parametro di scala.

Dato l'elevato rumore della variabile di controllo output del modello addestrato, si è deciso di effettuare una media mobile con bucket temporale di 300 secondi come nel caso precedente.



In Figura 3.104 è riportato l'andamento della media mobile della variabile di controllo Likelihood nel tempo con il relativo limite inferiore di controllo (LCL) pari a 5883 con confidenza del 95% e limite superiore di controllo relativo al periodo di pompaggio (SCL) pari a -1,148 con confidenza del 5%:

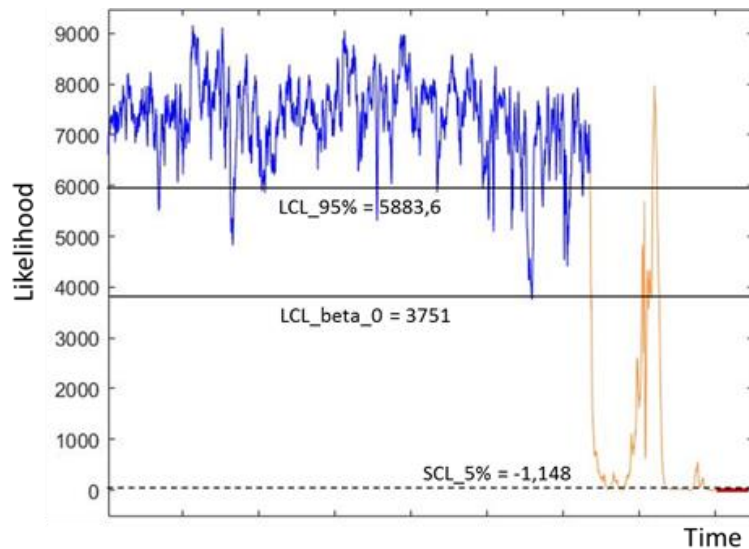


Figura 3.104 Evoluzione nel tempo della media mobile della variabile di controllo Likelihood.

Anche in questo caso è evidente come i dati di pompaggio (in rosso in figura) siano estremamente al di sotto del limite di controllo LCL; in particolare il massimo valore della variabile di controllo nel periodo di pompaggio è di -1,06 ed ha un valor medio di -1,19. Il modello ha quindi una elevata capacità diagnostica di pompaggio del compressore analizzato. L'errore di prima specie sul periodo di analisi è nullo ed avendo scelto un limite inferiore del 95% l'errore di seconda specie è pari al 5%. Per ottenere un errore di seconda specie nullo, è necessario utilizzare come logica di segnalazione del funzionamento anomalo del compressore la permanenza al di sotto della soglia LCL per un tempo maggiore di 146732 secondi (40 ore circa), vanificando il potere diagnostico del modello. In alternativa è possibile ottenere lo stesso risultato abbassando il limite di soglia LCL al valore di 3751 (LCL\_beta\_0). Utilizzando questo valore di soglia il modello è in grado di segnalare un'anomalia di funzionamento del componente 45,96 ore prima dell'insorgere del guasto.

### 3.8.3 - Applicazione del modello ARIMA

Una volta verificato che entrambi i modelli hanno un elevato potere diagnostico per il caso studio analizzato, abbiamo costruito un metodo di valutazione della qualità della stima della RUL del compressore immaginando un suo utilizzo in real-time sull'impianto. È stata utilizzata quindi la logica descritta nel paragrafo 2.5.4.1.

Il modello ARIMA è stato testato sui dati di output dei due modelli diagnostici costruiti, ovvero sui valori di  $T^2$  per le carte di controllo multivariate e sui valori di Likelihood output del modello One-class SVM, per confrontarne le performance predittive. Nel caso del modello che sfrutta le carte di Hotelling si iniziano a marcare i dati di  $T^2$  come anomali dopo che questi risultano oltre la soglia UCL\_95% per un tempo superiore a 2857s. Nel caso del modello diagnostico one-class si considerano anomali tutti i dati al di sotto della soglia LCL\_beta\_0. In entrambi i casi il modello ARIMA è stato inizializzato sui primi 1000 valori anomali. L'inizializzazione del modello calcola i parametri che meglio si adattano ai dati tramite ottimizzazione automatica basata sulla minimizzazione dell'AIC e del BIC. L'output del modello è il valore atteso e la relativa banda di confidenza del 95% dell'istante futuro in cui il valore della variabile di controllo proxy delle performance del compressore oltrepassa la soglia limite SCL\_5%. Il modello ARIMA inizializzato è per prima cosa adattato alla serie temporale di cui vogliamo la previsione utilizzando la funzione *estimate* e successivamente si effettua la previsione tramite la funzione *forecast* su software MATLAB, come descritto nel dettaglio nel paragrafo 2.5.4 -. La forte instabilità del parametro di controllo  $T^2$  nel periodo anomalo, fa sì che la previsione ARIMA sia estremamente instabile e altalenante per tutti i dati anomali fino all'insorgere dell'evento di pompaggio, rendendo assolutamente inaffidabile la previsione.

Per quanto riguarda il parametro Loglikelihood output del modello One-class SVM, questo presenta inizialmente un andamento discendente molto stabile. La previsione con il modello ARIMA infatti risulta anch'essa stabile e robusta fin dalle prime previsioni come mostrato in Figura 3.105 Prima previsione ARIMA della RUL..

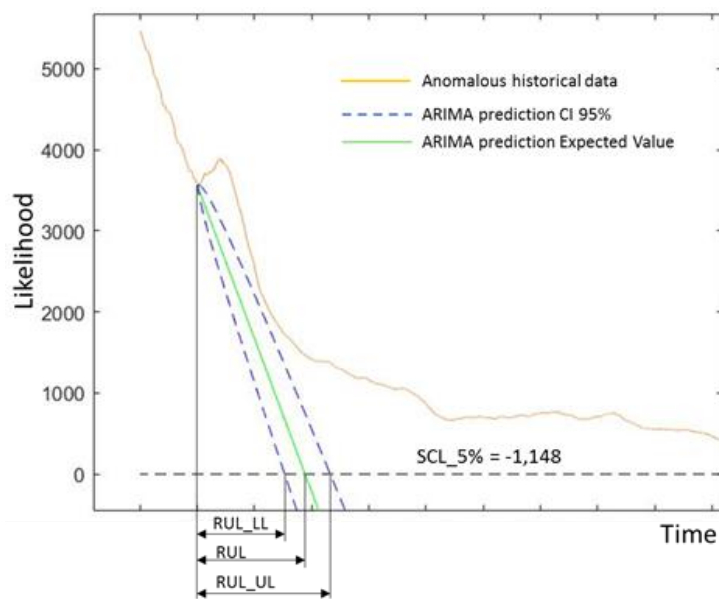


Figura 3.105 Prima previsione ARIMA della RUL.

Il modello ARIMA, stima una RUL attesa del componente di 31,4 minuti, e considerando un intervallo di confidenza del 95% sulla previsione, un limite inferiore di 25,5 minuti ed uno superiore di 38,8. Confrontando questo valore con l'arrivo effettivo del periodo di pompaggio che volevamo prevedere, la RUL risulta sottostimata di circa 45,68 ore.

Successivamente, come mostrato in Figura 3.104 Evoluzione nel tempo della media mobile della variabile di controllo Likelihood., il valore di *Loglikelihood* diviene molto instabile ed addirittura ritorna all'interno del limite di controllo LCL\_95%, rendendo impossibile continuare la previsione con il modello ARIMA.

In definitiva, entrambi i modelli di controllo di processo (carte di controllo multivariate e One-class SVM) applicati al caso studio presentano elevate prestazioni in termini di *state detection* di funzionamenti anomali del compressore studiato. Non possiamo dire lo stesso per quanto riguarda il modello ARIMA applicato alla variabile di controllo *proxy* delle prestazioni del compressore output dei due modelli diagnostici. In un caso infatti, l'andamento estremamente instabile delle prestazioni del compressore nel transitorio tra funzionamento a regime e pompaggio, non permette di effettuare alcuna previsione della RUL tramite il modello ARIMA. Nel secondo caso, invece, la previsione è estremamente stabile nel periodo iniziale, ma la stima della RUL è fortemente sottostimata.

La tabella di seguito riporta una sintesi delle prestazioni dei due modelli prognostici studiati.

Tabella 3.23 Performance dei modelli di state detection e previsione della RUL.

	Hotelling + ARIMA	One-Class SVM +ARIMA
Errore di prima specie	0%	0%
Errore di seconda specie	0%	0%
Diagnosi Precoce	45,34 ore	45,96 ore
Stima della RUL	n.d	31,4 minuti
RUL reale – RUL stima	n.d	45,68 ore

Il comportamento estremamente instabile del parametro *proxy* delle performance del compressore in entrambi i modelli diagnostici potrebbe essere dovuto alla particolare scelta del fenomeno di guasto in analisi. Il fenomeno del pompaggio infatti, non è un vero e proprio evento di guasto, ma piuttosto un

fenomeno che, se protratto nel tempo, porta alla rottura del compressore, per le elevate vibrazioni che ne comporta. Il pompaggio infatti non è causato da una cattiva condizione meccanica del compressore, ma dalle particolari condizioni termofluidodinamiche del gas lavorato e dalle condizioni del circuito in cui esso è inserito. In definitiva, dipendendo non solo dalle prestazioni del compressore ma complessivamente da tutto il circuito, questo fenomeno risulta molto difficile da prevedere.

Nel momento in cui si applica questa metodologia prognostica sul campo, l'assenza di dati relativi alle condizioni di guasto, fa sì che il limite di controllo SCL\_5% non possa essere calcolato a priori, ma ipotizzato ad una certa distanza dalla popolazione di regime, utilizzata per addestrare i modelli diagnostici. Dal singolo caso studio analizzato risulta che per il fenomeno del pompaggio, la popolazione di regime e di guasto sono estremamente distanti, soprattutto utilizzando il modello One-Class SVM.

Inoltre, stesse condizioni di bassa performance del macchinario monitorato, potrebbero essere causate da diversi eventi di guasto. La scelta delle corrette variabili di controllo di addestramento dei modelli di *state detection* può mitigare in parte questa incertezza. Nel momento in cui si hanno a disposizione dati relativi alle variabili spia del fenomeno di guasto anche nelle condizioni di cattivo funzionamento, è possibile e preferibile implementare un vero e proprio sistema prognostico più robusto sfruttando ad esempio un modello SVM a due classi come descritto nel paragrafo 3.7.1 -.

## 3.9 - Interfaccia modelli prognostici soft real-time

Come descritto all'inizio del capitolo 3 -il progetto SmartGEO, caso studio utilizzato per la validazione dei modelli diagnostici e prognostici sviluppati, si è avvalso di 4 partner di progetto per lo svolgimento dello stesso. Tra questi il compito di costruire un'interfaccia accessibile ed intuitiva per la visualizzazione dei risultati è stato assegnato al partner SDI automazione S.p.A.

Riporto nella tesi una sintesi del loro lavoro svolto perché lo ritengo interessante per rispondere in maniera completa alle domande di ricerca e proporre così una soluzione pratica per le aziende che intendono seguire lo stesso percorso affrontato da EGP.

Per il progetto SmartGeo non si è reso necessario lo sviluppo del lato *backend* per l'interfacciamento delle interfacce utente con i Server PI/AF OSIsoft perché questo è reso disponibile da OSIsoft attraverso lo strato software delle PI Web API che costituiscono un'interfaccia *RESTful* verso il sistema PI per rendere disponibili ai client accessi in lettura e scrittura ai dati AF e PI attraverso protocollo di comunicazione HTTPS.

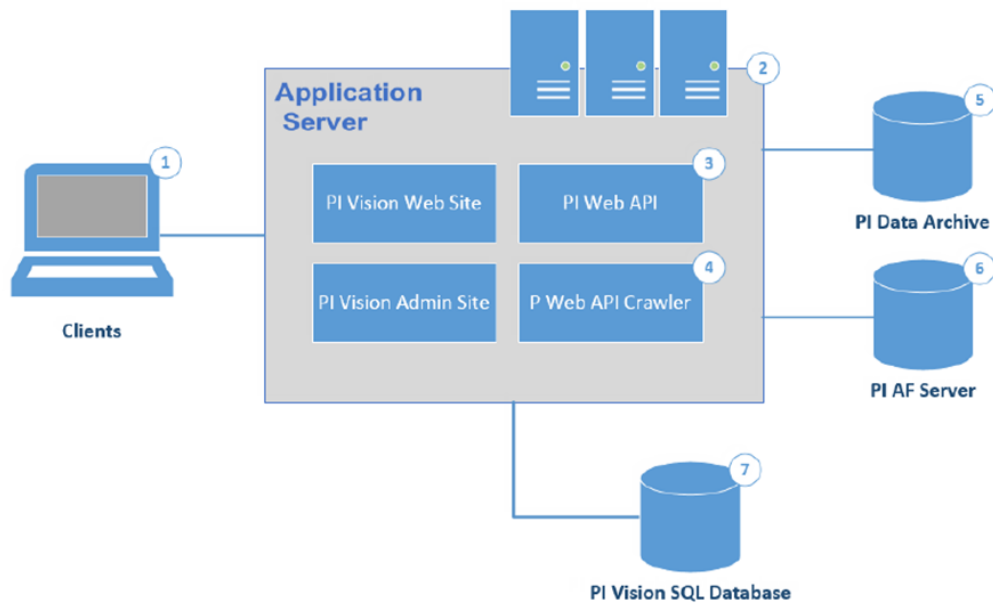


Figura 3.106 Servizi PI Web API.

All'interno dell'*application server* messo a disposizione nativamente da OSIsoft è presente il servizio *PI Web API* che supporta le funzionalità di recupero e manipolazione delle serie temporali dei dati, degli *asset* e degli *event frame*, il *PI Web Crawler* che raccoglie i metadati dal Sistema PI e fornisce gli *item* per velocizzare la ricerca indicizzata nei database da parte delle richieste HTTPS dai *Client* ed il *PI Vision Web Site* ed *Admin Site* che permettono la creazione dell'interfaccia *web* per la visualizzazione dei dati. Data la limitata elasticità degli strumenti *web PI Vision* si è infine optato per la costruzione di un sito *web* ad hoc chiamato "SmartGeoDiana" che permette, attraverso le *PI Web API* la visualizzazione intuitiva delle *dashboard* richieste dall'ingegneria di manutenzione.

Per la realizzazione del sito SmartGeoDiana sono state testate e selezionate librerie grafiche *web open source* per l'implementazione di oggetti complessi di presentazione dati real time, storici e diagnostici integrabili in ambiente OSIsoft, a partire da piattaforme hardware e software di tipo commerciale disponibili sul mercato, limitando al minimo le attività di personalizzazione software.

Dopo un'analisi approfondita sono state approvate le seguenti librerie *web open source*:

1. ECharts, framework per consente di creare velocemente visualizzazioni *Web-based* (integrata nel sito per la visualizzazione di grafici grafici trend, scatter...);
2. vis.js, libreria dinamica di visualizzazione *browser based*, semplice da usare e progettata per gestire, manipolare ed interagire con grandi quantità di dati dinamici (integrata nel sito per la rappresentazione del sinottico della rete vapore).

Per l'interfacciamento del sito web SmartGeoDiana con i server AF/PI è stata implementata una libreria con metodi pubblici e privati basati sulle *PI Web API* di OSISOFT. La libreria *piwebapi\_wrapper.js* del progetto *piwebapi-js-SMARTGEO* costituisce lo strato software di interfaccia tra *PI Web API* e le pagine *Web* di progetto ed implementazione delle richieste *ajax* di accesso in lettura/scrittura ai dati dei database dei Server AF e PI.


La landing page del sito web SmartGeoDiana sviluppato si apre con il link:

[https://<AF\\_Server\\_IP\\_address>/SmartGeoDiana/](https://<AF_Server_IP_address>/SmartGeoDiana/)

ed è riportata nella figura seguente.



Figura 3.107 Landing page del sito SmartGeoDiana.

Nella landing page del sito SmartGeoDiana è visualizzato il pulsante  che apre la *dialog box* dedicata ad effettuare il login al sito.

La pagina multistep a cui si accede consente l'interazione dell'utente con il sistema di diagnostica e prognostica messo a punto per il progetto SmartGeo.

Gli step consentono di creare e configurare nuove istanze di elementi diagnostici (*AF Diagnostics*), di monitorare lo stato di elementi già istanziati dall'utente loggato e di eseguire dei comandi di schedulazione degli stessi. La funzione di cancellazione delle istanze di elementi diagnostici è abilitata

solo per utenti con autorizzazioni di amministratore. La presentazione dei risultati dell'esecuzione degli algoritmi è demandata a pagine web dedicate (trend, scatter).

Nella figura seguente è visualizzata la pagina dello *Step 1* con la tabella delle istanze di elementi prognostici definiti, presenta dati di configurazione e dati aggiornati dinamicamente (*refresh* al minuto).

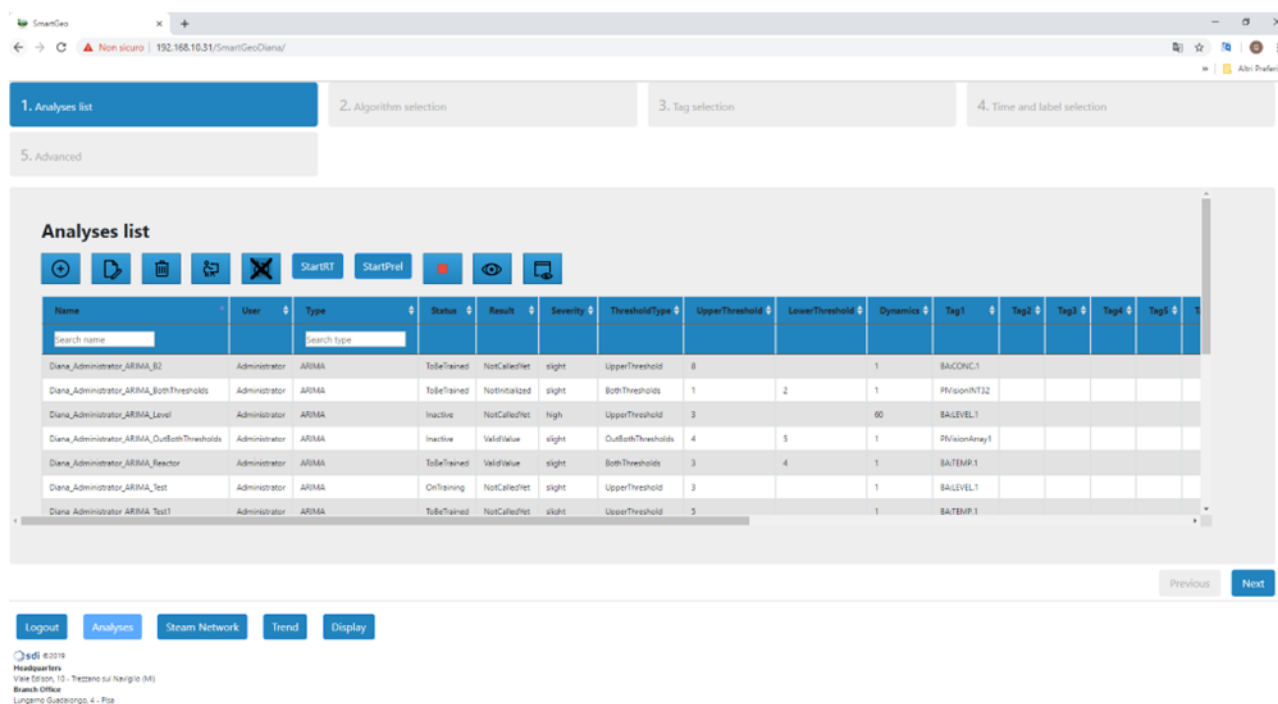


Figura 3.108 Pagina multistep del sito SmartGeoDiana (Lista analisi).

Il sito SmartGeoDiana realizza, tramite una pulsantiera dedicata, le seguenti operazioni di schedulazione sulle istanze di analisi definite nel sistema:

- Attiva la fase di *training* dell'elemento diagnostico e/o prognostico selezionato;
- Ferma la fase di *training* dell'elemento diagnostico e/o prognostico selezionato;
- Attiva l'esecuzione real time dell'elemento diagnostico e/o prognostico selezionato;
- Attiva la fase di preloading dei dati di input dell'elemento diagnostico e/o prognostico selezionato;
- Ferma l'esecuzione dell'elemento diagnostico e/o prognostico selezionato.

Nella figura seguente è visualizzata la tabella degli eventi associati all'istanza di analisi selezionata.

### Event frame table

Name	Description	Template	Severity	Start Time	End Time	Categories	Label	Tag01	Tag02	Tag03	Tag04
Diana_Administrator_ARIMA_Reactor_1	Event frame from user Administrator	DianaFrameARIMA	None	2019-07-08 02:39	2019-07-17 03:44	Diagnostics	good functioning	BA,TEMP,1			

Figura 3.109 Pagina multistep del sito SmartGeoDiana (Lista event frame).

Nella figura seguente è visualizzata la pagina dello Step 2 per consentire all’utente di creare una nuova istanza di elemento diagnostico. Gli step successivi presentano dati differenti in funzione del tipo di analisi selezionato nello Step 2. Come si vede dalla figura l’utente sceglie il modello diagnostico e/o prognostico che intende utilizzare in funzione di una breve descrizione dell’algoritmo scelto.

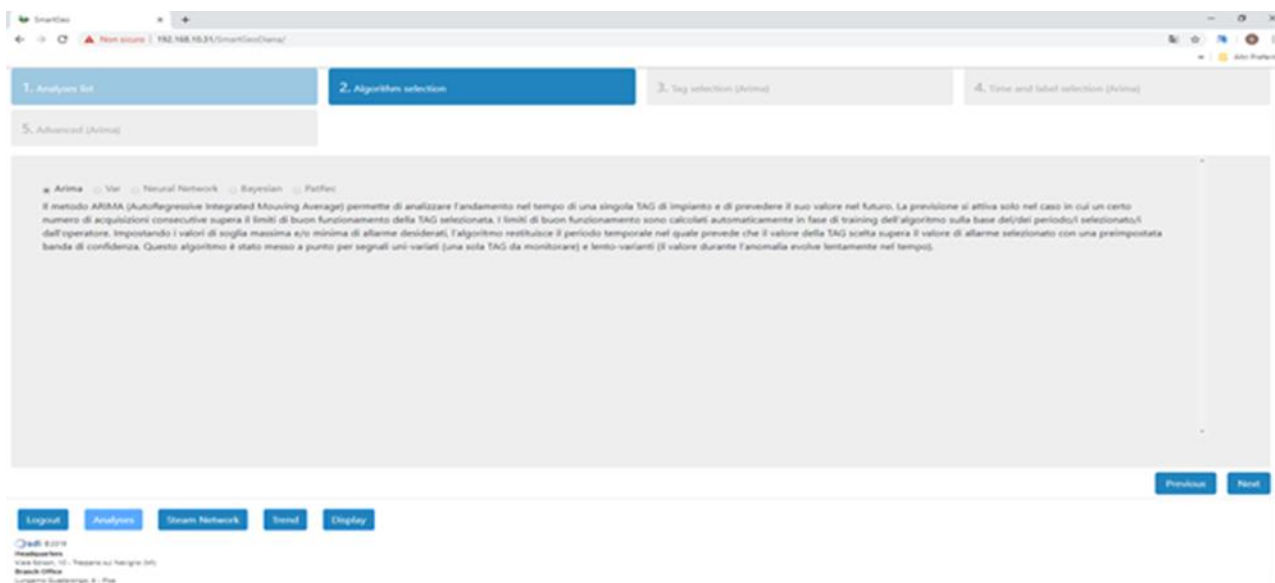


Figura 3.110 Pagina multistep del sito SmartGeoDiana (Definizione tipo analisi).

Nella figura seguente è visualizzata la pagina dello Step 3 per la selezione della lista dei PI point che costituiscono un subset di dati di input per l’elaborazione dell’elemento diagnostico.



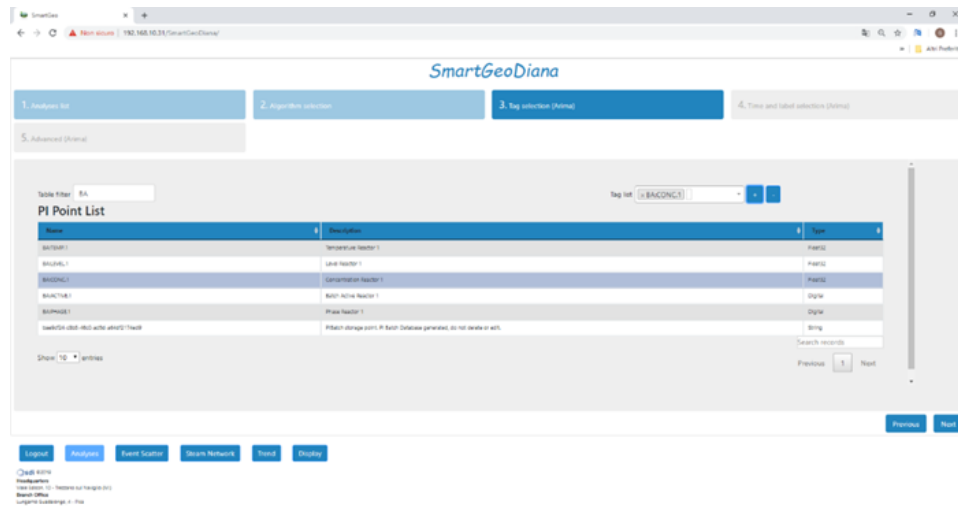


Figura 3.111 Pagina multistep del sito SmartGeoDiana (Selezione tag PI)

Nella figura seguente è visualizzata la pagina dello Step 4 per la selezione dei range temporali e delle label (regime o tipo di guasto) da associare all’elaborazione dell’elemento diagnostico.

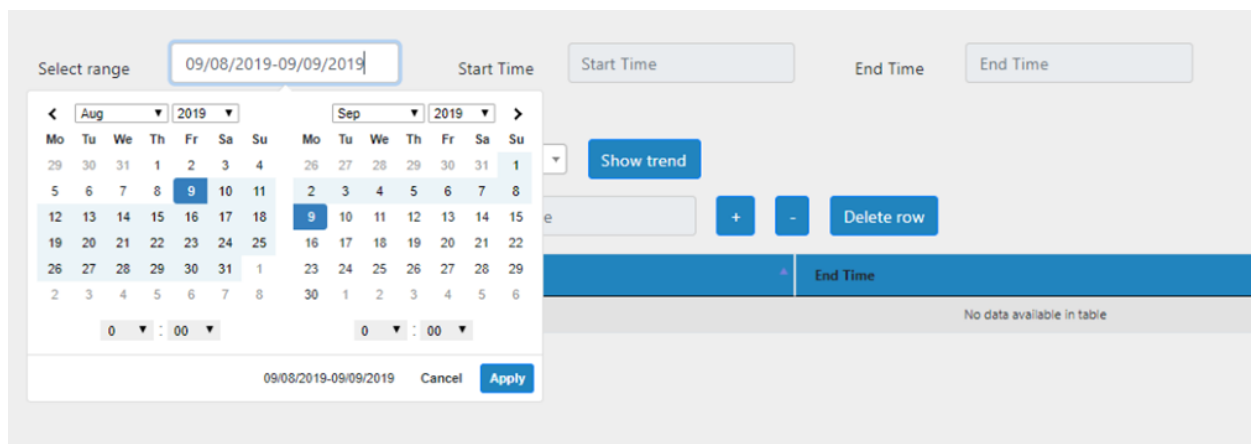


Figura 3.112 – Pagina multistep del sito SmartGeoDiana (Selezione range temporale)

Nella figura seguente è visualizzata la pagina dello Step 5 di definizione dei parametri avanzati di configurazione della nuova istanza di elemento diagnostico.

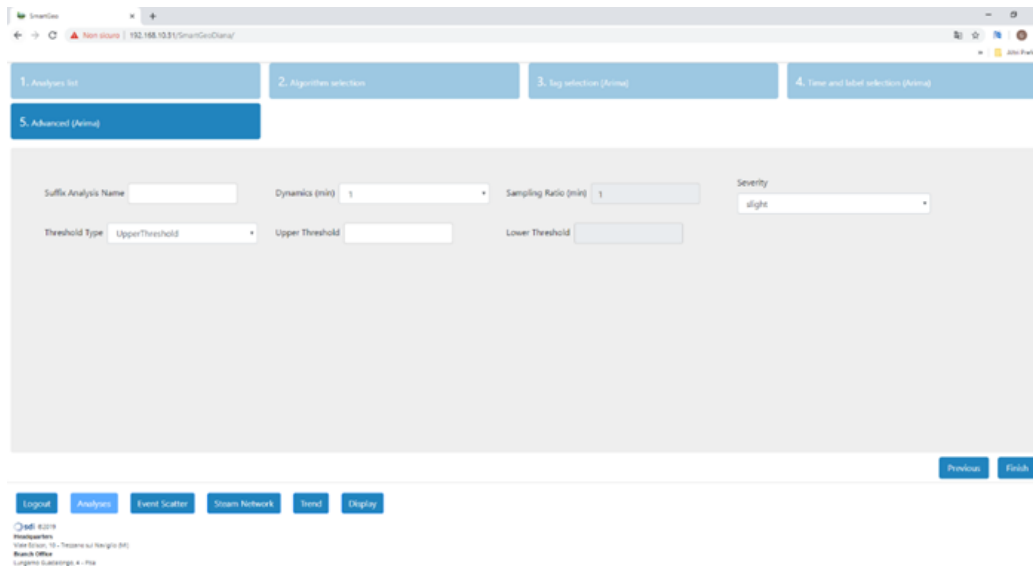


Figura 3.113 – Pagina multistep del sito SmartGeoDiana (Configurazione dati advanced)

Una volta istanziate le analisi queste saranno visibili nella homepage dello specifico utente.

I risultati delle analisi sono visualizzabili in tempo reale grazie alla schermata chiamata “scatter plot”.

Come possiamo vedere dalla figura seguente, questo tipo di grafico mostra degli eventi futuri identificandoli per gravità (colore), risultato della severity dell’analisi FMEA iniziale, e probabilità di accadimento. Il risultato dello scatter è ottenuto dagli output dei modelli prognostici basati sulla previsione ARIMA presentati. Grazie a questo tipo di grafico gli operatori di manutenzione potranno accedere a informazioni altamente informative e fruibili per permettere il buon esercizio dei sistemi di impianto.

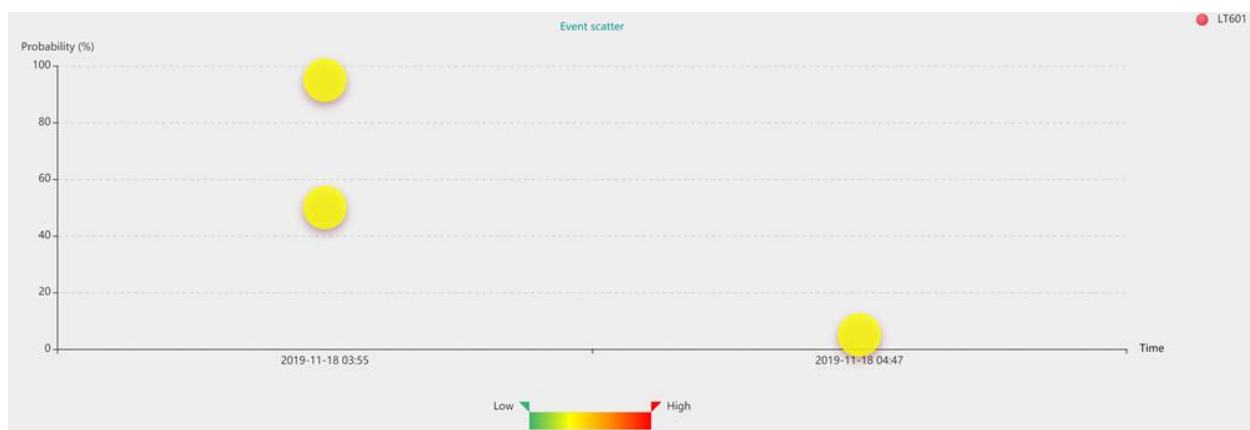


Figura 3.114 – Pagina grafico scatter dei risultati

In definitiva grazie a questa interfaccia interattiva ogni utente può creare agevolmente la propria analisi diagnostica e/o prognostica utilizzando i modelli parametrici rilasciati nell'architettura informatica:

1. Diagnostico: SVM per la classificazione del comportamento a regime o a guasto di un componente;
2. Diagnostico: ShapeFind per l'individuazione di particolari forme d'onda in un segnale;
3. Stima della RUL: ARIMA per prevedere la RUL di un macchinario utilizzandola in combinazione con un modello diagnostico o di state detection;
4. State detection: One-class SVM per riconoscere comportamenti anomali di un macchinario le cui prestazioni dipendono da numerose variabili;
5. State-detection: Carte di controllo uni e multivariate per riconoscere comportamenti anomali di un macchinario le cui prestazioni dipendono da una o più variabili.

### 3.10 - Prototipo di applicativo per prognostica soft real-time

Al fine di ottenere uno strumento a basso costo che potesse implementare il modello prognostico più promettente tra quelli sviluppati, ho costruito un prototipo di applicativo che sulla base di quanto descritto nel capitolo precedente è in grado, una volta connesso al database di effettuare prognostica basandosi sul modello diagnostico SVM (classificazione) e di analisi di serie temporali ARIMA. La scelta è stata dettata dagli ottimi risultati in termini di accuratezza del modello diagnostico applicato al pompaggio del compressore.

Il prototipo di applicazione (il cui codice sviluppato è in Appendice 2) si interfaccia con un DB *OpenSource InfluxDB*, database di serie temporali sviluppato da *InfluxData*. Per utilizzare il prototipo è quindi necessario storicizzare i dati di impianto su questo tipo di DB.

Il prototipo è stato sviluppato in MATLAB utilizzando la funzione *AppDesigner*, in modo da essere agevolati nella scrittura dei modelli che erano stati già sviluppati in linguaggio .m.

Si compone di due schermate:

1. Addestramento del modello diagnostico SVM;
2. Previsione real-time della RUL.

La prima schermata, raffigurata in Figura 3.115 – Schermata di addestramento riportata in alto a sinistra i parametri necessari alla connessione dell'applicativo al DB Influx ed in particolare:

1. URL: indirizzo http della macchina su cui risiede il DB Influx
2. USER: username per l'accesso al DB;
3. PASSWORD: password per l'accesso al DB
4. DATABASE: nome del database al quale vogliamo accedere.

In alto a destra si ha invece la finestra di dialogo.

Il pulsante "Prova connessione" permette di testare i parametri di connessione inseriti e la conferma di avvenuto accesso al DB è riportata nella finestra di dialogo.

Una volta testata la connessione, l'app chiede all'utente se esista o meno un modello SVM già addestrato. Se la risposta è affermativa cliccando il pulsante "Invia" si apre una finestra di dialogo che permette di ricercare all'interno del PC il modello addestrato. In caso questo non sia presente si procede con la compilazione delle tabelle.

Nella tabella a sinistra si indicano i dati necessari all'individuazione nel DB delle variabili di addestramento necessarie; in quella sulla destra si indicano invece, inserendo l'istante di inizio e di fine del periodo, gli slot temporali in cui il componente che vogliamo analizzare si trovava nella condizione di regime o di guasto. Ciò che permettere di discernere tra le due classi di funzionamento è il corretto inserimento del "Label":

- Pari a 0 se nel periodo di riferimento il macchinario stava funzionando a "regime"
- Pari ad 1 se nel periodo di riferimento il macchinario aveva iniziato il degrado dovuto al guasto.

Una volta compilate le tabelle e premuto il tasto "Crea Training\_set" l'algoritmo provvede alla creazione del dataset di training del modello SVM effettuando anche una sincronizzazione delle serie temporali indicate. Una volta creato il *training\_set* restituisce due informazioni su quanto appena creato che aiutano a decidere se effettuare o meno un bilanciamento perfetto (sbilanciamento *training\_set* pari al 50%) del *training\_set* tramite eliminazione casuale di dati dalla classe più numerosa. Le informazioni fornite sono il numero di campioni del *training\_set*, necessario per valutare se una sua riduzione ulteriore è accettabile e lo sbilanciamento del *training\_set* che è un valore percentuale che indica quanti più dati ho in una classe rispetto all'altra (solitamente accettabile se < 70%).

A questo punto si procede scegliendo il *Kernel* del modello SVM che può essere:

1. Gaussiano;
2. Lineare;

3. Polinomiale di secondo grado;
4. Polinomiale di terzo grado.

Premendo il pulsante “Addestra” viene addestrato il modello SVM a due classi con il kernel selezionato e parametro di scala ottimizzato in automatico dall’algoritmo. Una volta terminato l’addestramento, se l’accuratezza è di gradimento, premendo il pulsante “Fine Addestramento” il modello diagnostico SVM è pronto per essere utilizzato in real-time.

### Connessione al db Influx

URL

USER

PASSWORD

DATABASE

connessione effettuata  
ping =162.2355  
Controllare i dati inseriti nelle tabelle! Dot indexing is not supported for variables of this type.

Esiste un modello SVM addestrato?

SI  NO

### Variabili di addestramento

Database	Serie	Field
mydb	RA2_GS	TT_450
mydb	RA2_GS	PT_301A
mydb	RA2_GS	TT_405
mydb	RA2_GS	PT_453

### Periodi di addestramento

Formato data: gg/mm/aaaa hh:mm:ss    Regime: Label = 0  
Anomalia: Label = 1

Inizio (Time)	Fine (Time)	Label
02/07/2017 09:50:00	03/07/2017 05:00:00	0
06/03/2017 17:00:00	15/03/2017 04:00:00	0
11/05/2017 16:33:00	15/05/2017 10:00:00	0
15/03/2017 23:00:00	26/05/2017 03:00:00	0

N° campioni training\_set

Sbilanciamento training\_set

Kernel

- Gaussiano
- Lineare
- Polinomiale 2
- Polinomiale 3

Accuratezza

Figura 3.115 – Schermata di addestramento

Una volta terminato l’addestramento, l’applicativo apre una seconda finestra di dialogo, la finestra di previsione real-time della RUL del macchinario sotto analisi e riportata in Figura 3.116.

Questa finestra presenta due schede, una di “impostazioni” ed una di visualizzazione dei risultati. Nella prima scheda si inseriscono tutti i parametri necessari all’inizializzazione del modello ARIMA di previsione, che entrerà in funzione solo quando si registrerà un valore all’interno della classe di funzionamento a guasto del macchinario. I parametri da inserire sono gli stessi descritti nel paragrafo 2.5.4 -. Vorrei focalizzare l’attenzione sul parametro “soglia”, che indica la soglia massima di accettabilità della distanza normalizzata tra 0 e 1 del punto attuale di funzionamento dalla classe di Regime.

Oltre a questi è necessario impostare il “Tempo di campionamento”, ovvero ogni quanti secondi attivare la lettura ed analisi di un nuovo dato real-time a database. Questo parametro dovrà essere coerente con il tempo scelto di storicizzazione a database. Infine, nella finestra “output della previsione su Influx” è necessario indicare dove vogliamo registrare gli output della previsione sul *database* stando attenti a non utilizzare un nome per la “Series” già in uso. Sulla destra anche in questo caso abbiamo la finestra di dialogo ed in basso i pulsanti di “Start” e “Stop” dell’analisi prognostica.

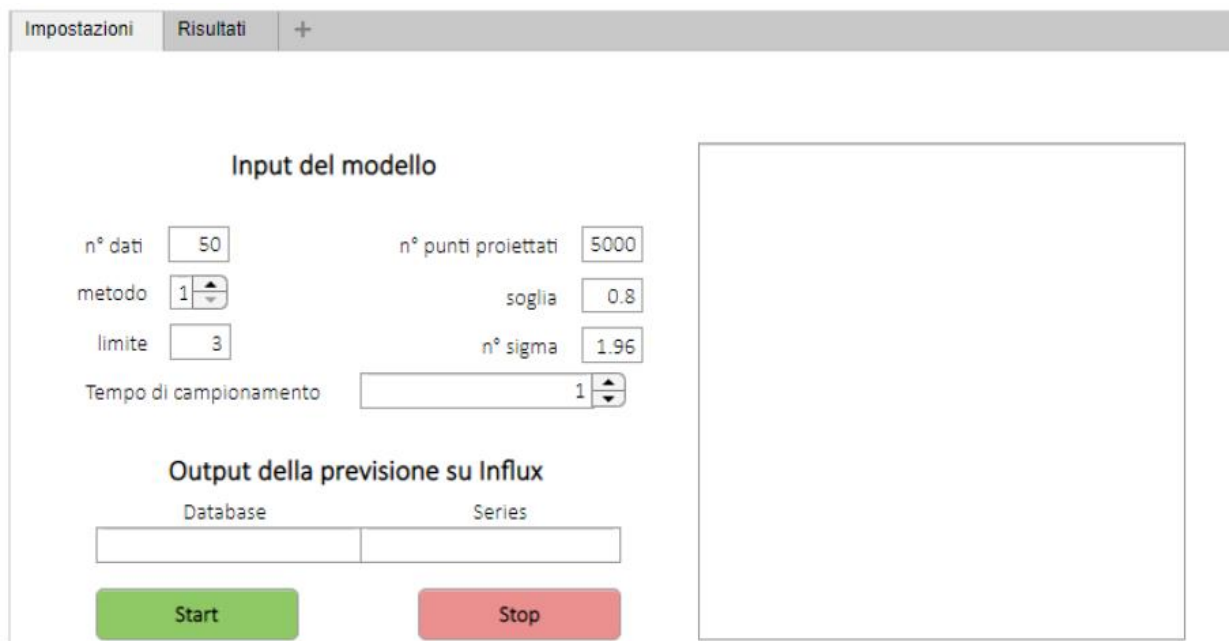


Figura 3.116 – Finestra “Impostazioni” della schermata di previsione realtime della RUL

Nella scheda “Risultati”, riportata in Figura 3.117, si possono invece visualizzare i risultati dell’analisi. In alto a destra sono riportati gli output del modello diagnostico, ovvero il label attuale (0 se funziona a regime, 1 se a guasto) e la distanza dalla superficie di separazione delle classi normalizzata tra 0 ed 1, dove 0 significa che il punto è a cavallo tra le classi ed 1 che si trova alla distanza massima.

Il grafico in alto a sinistra riporta il punto di funzionamento attuale rispetto alle classi di buon funzionamento (regime) e guasto a cui è aggiunta una fascia di anomalia che delimita la soglia di

accettabilità imposta nella scheda "Impostazioni". Il punto di funzionamento cambia colore in base alla zona di appartenenza (blu se nell'area di buon funzionamento, giallo se in quella di anomalia e rosso se a guasto).

Quando il modello ARIMA sarà in grado di prevedere il superamento della soglia di accettabilità, il risultato sarà visibile nel grafico in basso che altro non è che una specie di calendario. Il triangolo rosso indica l'istante temporale futuro in cui si ha il valore atteso di superamento della soglia ed i due triangoli blu il suo intervallo di confidenza dettato dal parametro "n° sigma impostato".

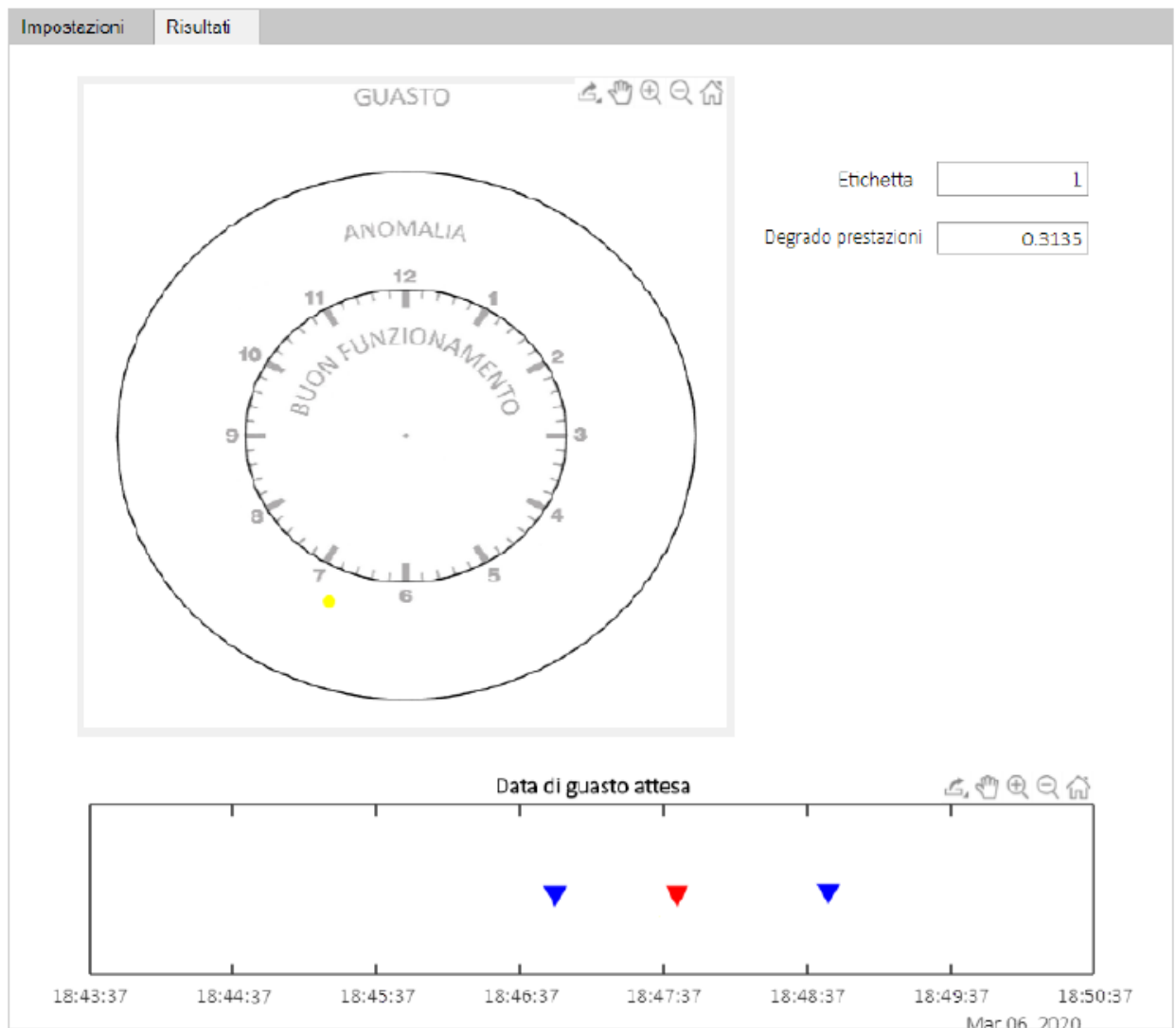


Figura 3.117 – Finestra "Risultati" della schermata di previsione realtime della RUL

Questo semplice prototipo di applicativo può essere utilizzato per testare il modello prognostico basato sulla classificazione SVM e l'analisi delle serie temporali ARIMA sui dati del proprio impianto prima di

investire in una sua integrazione con il sistema software di gestione degli asset. Sto lavorando alla sua traduzione in Python che lo renderebbe un prodotto completamente Open Source.



## 4 - Discussione

I modelli diagnostici e prognostici teorizzati ed applicati al caso studio hanno mostrato in generale delle ottime performance soprattutto per quanto riguarda i modelli diagnostici e di state-detection. Il modello di stima della RUL presentato, che sfrutta il modello di analisi delle serie temporali ARIMA, si è invece rilevato non sempre adeguato alla previsione come dettagliato nel paragrafo 3.8.3 -. Questo potrebbe essere dovuto al fenomeno in analisi; il fenomeno del pompaggio infatti, come già discusso, non è un vero e proprio fenomeno di guasto ma un'instabilità che se prolungata può portare al guasto per rottura dei componenti meccanici. L'insorgenza del fenomeno deve essere gestita da modelli hard real-time che come dettagliato nel paragrafo 2.4 - non prevedono per loro natura la fase di previsione della vita utile residua. Essendo però il compressore centrifugo l'unico componente di impianto del quale disponevamo dei dati sia di funzionamento a regime che di degrado fino all'insorgenza del fenomeno di "guasto", è stato comunque utilizzato per la validazione anche del modello di stima della RUL con modello statistico ARIMA. Sarebbe quindi necessario, al fine di validare il modello proposto, l'applicazione ad altri casi studio che presentino una base dati eterogenea e consistente e testare anche altre tecniche di analisi di serie temporali come ad esempio modelli machine learning come le NAR (Nonlinear AutoRegressive neural network). Nel corso dei 3 anni di dottorato, nonostante la mia collaborazione anche in altri progetti cofinanziati dalla regione toscana, in particolare il progetto IoT4WATER e IoT4Yacht, che hanno avuto lo stesso obiettivo del progetto SmartGEO, ovvero quello di implementare un sistema di manutenzione predittiva, nel primo caso ad un impianto di trattamento di acque reflue e nel secondo caso ad una imbarcazione di tipo Yacht, non ho mai avuto a disposizione una base dati consistente che mi permettesse di validare il modello di stima della RUL su un modo di guasto che richieda modelli soft real-time.

Per quanto riguarda i modelli diagnostici proposti, questi presentano tutti un'elevata accuratezza nel segnalare condizioni di degrado del componente in analisi.

Dato il doppio compito che svolge il modello diagnostico di:

1. Diagnosticare in funzione delle condizioni di degrado lo specifico modo di guasto;
2. Distillare una variabile proxy delle performance del macchinario a partire dalle variabili spia del modo di guasto in analisi.

Abbiamo selezionato per l'applicazione al caso studio il modello di classificazione SVM che comprende entrambe le funzioni. La classe attuale prevista dal classificatore indica l'eventuale modo di guasto in atto

e la distanza dalla superficie di separazione tra le classi indica invece il livello di degrado delle prestazioni del componente. Il modello fino ad adesso validato è un modello addestrato su 2 classi. La classe di funzionamento a regime e la classe di funzionamento nel particolare modo di guasto di interesse. Ogni componente o macchinario di impianto può essere però soggetto a differenti modi di guasto ed un buon modello diagnostico deve essere in grado non solo di individuare l'inizio del fenomeno di degrado ma anche di metterlo in relazione al particolare modo di guasto. In questo senso il modello SVM risulta comunque essere indicato, infatti può essere addestrato anche su molteplici classi. In questo caso quindi il numero di classi su cui addestrare il modello saranno pari al numero dei modi di guasto individuati dall'analisi FMEA più una classe corrispondente al funzionamento a regime del macchinario.

Per quanto riguarda l'architettura informatica proposta, questa risulta essere efficiente ed efficace grazie anche alle caratteristiche del software PI System di OsiSoft che mostra elevate capacità di interconnessione con strumenti di analisi e visualizzazione avanzati. All'interno del caso studio, è mancata però una gestione più struttura degli interventi di guasto, applicabile adottando in impianto un software per la gestione degli interventi manutentivi (Computerized Maintenance Management System – CMMS), che permetta di collegare più agilmente un periodo di funzionamento anomalo con un particolare intervento di manutenzione. All'interno del progetto queste informazioni erano disponibili solo a livello cartaceo e sono state poi trascritte su un foglio Excel per una lettura migliore. Uno strumento software più avanzato e strutturato permette di ottenere tre vantaggi principali:

1. Aiuta lo svolgimento dell'analisi dei modi di guasto;
2. Permette di strutturare e codificare le informazioni in modo da collegare inequivocabilmente una anomalia di funzionamento del componente allo specifico modo di guasto;
3. Permette quindi di agevolare il processo di addestramento del modello diagnostico e di renderlo semi-automatico.

Infatti, utilizzando un CMMS di impianto integrato al software di gestione del DCS di impianto, ogniqualvolta si effettuano interventi manutentivi non diagnosticati dai modelli sviluppati si può prevedere un riaddestramento dei modelli stessi integrando nel training-set gli ulteriori valori anomali registrati a database. Questo processo che non può essere effettuato in maniera completamente automatica, in quanto l'operatore dovrà individuare il periodo antecedente all'intervento di guasto nel quale sono stati registrati valori anomali, può essere però guidato dal sistema di manutenzione predittiva sia nell'individuazione del guasto che nella necessità o meno di effettuare un nuovo addestramento dei modelli diagnostici.

---

## Conclusioni e sviluppi futuri

In riferimento alle domande di ricerca del mio progetto di dottorato, descritte nel paragrafo 1.4 -, tramite l'analisi globale della letteratura e l'analisi degli articoli specifici per i componenti del caso studio è stato possibile analizzare nel dettaglio i metodi diagnostici e prognostici in funzione della tipologia di componente e modo di guasto in analisi rispondendo di fatto alla prima domanda di ricerca (Quali sono i modelli diagnostici e prognostici adottati per componenti di impianti industriali? Quali metodologie adottare in funzione della tipologia di componente?).

Successivamente grazie anche all'applicazione al caso studio è stato possibile creare una metodologia generale in cui sono descritte le macro-fasi necessarie per implementare un sistema prognostico in impianto (paragrafo 2.1 -) rispondendo quindi anche alla seconda domanda di ricerca ("Quali passi devono essere seguiti per applicare prognostica in impianto? Su quali componenti dovrebbe essere focalizzata l'analisi?").

Infine, analizzando l'architettura informatica e le tecnologie utilizzate per implementare i modelli nel caso studio, ho potuto testare alcune delle più avanzate tecnologie presenti sul mercato per la loro implementazione sul campo, rispondendo così alla terza domanda di ricerca ("Quali sono le tecnologie chiave per l'implementazione di queste metodologie?").

Nonostante lo sforzo di utilizzare modelli diagnostici e prognostici più generali possibili, grazie all'uso di modelli statistici e machine learning che non sono legati alle peculiarità del componente in analisi o del particolare modo di guasto, l'applicazione di queste ad un singolo caso studio, non consente di generalizzare le metodologie proposte ed i modelli presentati. Di conseguenza in ottica di proseguimento del lavoro di tesi presentato, risulta di assoluto interesse l'applicazione dei modelli e delle tecnologie proposte anche in altri contesti industriali, sperando di avere a disposizione una base dati eterogenea e consistente per validare al meglio anche modelli di tipo soft-realtime.



# Bibliografia

- [1] \AAkerman, M., Lundgren, C., Barring, M., Folkesson, M., Berggren, V., Stahre, J., Engström, U., Friis, M., 2018. Challenges building a data value chain to enable data-driven decisions: a predictive maintenance case in 5G-Enabled manufacturing. *Procedia Manuf.* 17, 411–418.
- [2] Abramovici, M., Wolf, M., Adwernat, S., Neges, M., 2017. Context-aware maintenance support for Augmented Reality assistance and synchronous multi-user collaboration. *Procedia CIRP* 59, 18–22.
- [3] Adams, S., Malinowski, M., Heddy, G., Choo, B., Beling, P.A., 2017. The WEAR methodology for prognostics and health management implementation in manufacturing. *J. Manuf. Syst.* 45, 82–96.
- [4] Ahmad, W., Khan, S.A., Kim, J.-M., 2017. A hybrid prognostics technique for rolling element bearings using adaptive predictive models. *IEEE Trans. Ind. Electron.* 65, 1577–1584.
- [5] Ahuett-Garza, H., Kurfess, T., 2018. A brief discussion on the trends of habilitating technologies for Industry 4.0 and Smart manufacturing. *Manuf. Lett.* 15, 60–63.
- [6] Alcácer, V., Cruz-Machado, V., 2019. Scanning the industry 4.0: A literature review on technologies for manufacturing systems. *Eng. Sci. Technol. Int. J.* 22, 899–919.
- [7] Algoritmo Support Vector Machine, 2018. . Lorenzo Govoni. URL <https://lorenzogovoni.com/support-vector-machine/> (accessed 10.17.20).
- [8] Allmendinger, G., Lombreglia, R., 2005. Four strategies for the age of smart services. *Harv. Bus. Rev.* 83, 131.
- [9] Angelopoulos, A., Michailidis, E.T., Nomikos, N., Trakadas, P., Hatziefremidis, A., Voliotis, S., Zahariadis, T., 2020. Tackling Faults in the Industry 4.0 Era—A Survey of Machine-Learning Solutions and Key Aspects. *Sensors* 20, 109.
- [10] Awad, M., Khanna, R., 2015. Support vector regression, in: *Efficient Learning Machines*. Springer, pp. 67–80.
- [11] Baraldi, P., Di Maio, F., Al-Dahidi, S., Zio, E., Mangili, F., 2017. Prediction of industrial equipment remaining useful life by fuzzy similarity and belief function theory. *Expert Syst. Appl.* 83, 226–241.
- [12] Bechhoefer, E., Kingsley, M., 2009. A review of time synchronous average algorithms, in: *Annual Conference of the Prognostics and Health Management Society*. San Diego, pp. 1–10.
- [13] Bellotti, R., Trotta, E., n.d. *Macchine di Boltzmann per l'analisi di immagini mediche* 72.
- [14] Bolzern, P.G.E., Scattolini, R., Schiavoni, N.L., 2008. *Fondamenti di controlli automatici*. McGraw-Hill.
- [15] Boyes, H., Hallaq, B., Cunningham, J., Watson, T., 2018. The industrial internet of things (IIoT): An analysis framework. *Comput. Ind.* 101, 1–12.
- [16] Bromley, R.C., Bottomley, E., 1994. Failure modes, effects and criticality analysis (FMECA), in: *IEE Colloquium on Masterclass in Systems Engineering-Part Two*. IET, pp. 1–1.
- [17] Brown, R.N., 1997. *Compressors: Selection and sizing*. Gulf Professional Publishing.
- [18] Cai, J., Li, X., Chen, X., 2016. Optimization of Aeroengine Shop Visit Decisions Based on Remaining Useful Life and Stochastic Repair Time. *Math. Probl. Eng.* 2016.
- [19] Candy, J.V., 2016. *Bayesian signal processing: classical, modern, and particle filtering methods*. John Wiley & Sons.
- [20] Carlson, C.S., 2014. Understanding and applying the fundamentals of FMEAs, in: *Annual Reliability and Maintainability Symposium*. pp. 1–35.
- [21] Carnero, M.C., Gómez, A., 2017. A Multicriteria Model for Optimization of Maintenance in Thermal Energy Production Systems in Hospitals: A Case Study in a Spanish Hospital. *Sustainability* 9, 493.
- [22] Carvalho, T.P., Soares, F.A., Vita, R., Francisco, R. da P., Basto, J.P., Alcalá, S.G., 2019. A systematic literature review of machine learning methods applied to predictive maintenance. *Comput. Ind. Eng.* 137, 106024.

- [23] Castleberry, K.N., Nowlin, C.H., 1990. Machine monitoring via motor-current demodulation techniques, in: Proc of the 44th Meeting of the Mechanical Failures Prevention Group. pp. 87–96.
- [24] Chamberlain, G., 1982. Multivariate regression models for panel data. *J. Econom.* 18, 5–46.
- [25] Chen, Z., Li, Y., Xia, T., Pan, E., 2019. Hidden Markov model with auto-correlated observations for remaining useful life prediction and optimal maintenance policy. *Reliab. Eng. Syst. Saf.* 184, 123–136.
- [26] Cheng, F., Qu, L., Qiao, W., Hao, L., 2018. Enhanced particle filtering for bearing remaining useful life prediction of wind turbine drivetrain gearboxes. *IEEE Trans. Ind. Electron.* 66, 4738–4748.
- [27] Choi, W., Youn, B.D., Oh, H., Kim, N.H., 2019. A Bayesian approach for a damage growth model using sporadically measured and heterogeneous on-site data from a steam turbine. *Reliab. Eng. Syst. Saf.* 184, 137–150.
- [28] Civerchia, F., Bocchino, S., Salvadori, C., Rossi, E., Maggiani, L., Petracca, M., 2017. Industrial Internet of Things monitoring solution for advanced predictive maintenance applications. *J. Ind. Inf. Integr.* 7, 4–12.
- [29] Compare, M., Zio, E., 2014. Predictive maintenance by risk sensitive particle filtering. *IEEE Trans. Reliab.* 63, 134–143.
- [30] Corazza, M.V., Magnalardo, S., Musso, A., Petracci, E., Tozzi, M., Vasari, D., de Verdalle, E., 2017. Testing an innovative predictive management system for bus fleets: outcomes from the Ravenna case study. *IET Intell. Transp. Syst.* 12, 286–293.
- [31] Culot, G., Nassimbeni, G., Orzes, G., Sartor, M., 2020. Behind the definition of industry 4.0: Analysis and open questions. *Int. J. Prod. Econ.* 107617.
- [32] Dalenogare, L.S., Benitez, G.B., Ayala, N.F., Frank, A.G., 2018. The expected contribution of Industry 4.0 technologies for industrial performance. *Int. J. Prod. Econ.* 204, 383–394.
- [33] Daniotti, B., Gianinetto, M., Della Torre, S., 2020. Digital transformation of the design, construction and management processes of the built environment. Springer Nature.
- [34] Daponte, P., Lamonaca, F., Picariello, F., De Vito, L., Mazzilli, G., Tudosa, I., 2018. A survey of measurement applications based on IoT, in: 2018 Workshop on Metrology for Industry 4.0 and IoT. IEEE, pp. 1–6.
- [35] Day, I.J., 2016. Stall, surge, and 75 years of research. *J. Turbomach.* 138.
- [36] de Aguiar, D.C., Salomon, V.A.P., Mello, C.H.P., 2015. An ISO 9001 based approach for the implementation of process FMEA in the Brazilian automotive industry. *Int. J. Qual. Reliab. Manag.*
- [37] de Andrade Vieira, R.J., Sanz-Bobi, M.A., 2013. Failure risk indicators for a maintenance model based on observable life of industrial components with an application to wind turbines. *IEEE Trans. Reliab.* 62, 569–582.
- [38] De Jager, B., 1995. Rotating stall and surge control: A survey, in: Proceedings of 1995 34th IEEE Conference on Decision and Control. IEEE, pp. 1857–1862.
- [39] Demuth, H.B., Beale, M.H., De Jess, O., Hagan, M.T., 2014. Neural network design. Martin Hagan.
- [40] Dhillon, B.S., 2002. Engineering maintenance: a modern approach. cRc press.
- [41] Dickson, M.H., Fanelli, M., 2004. Cos' è l'energia geotermica. *Geotherm. Energy.*
- [42] Dilberoglu, U.M., Gharehpapagh, B., Yaman, U., Dolen, M., 2017. The role of additive manufacturing in the era of industry 4.0. *Procedia Manuf.* 11, 545–554.
- [43] Djuric, A.M., Urbanic, R.J., Rickli, J.L., 2016. A framework for collaborative robot (CoBot) integration in advanced manufacturing systems. *SAE Int. J. Mater. Manuf.* 9, 457–464.
- [44] Dong, G., Chen, Z., Wei, J., Ling, Q., 2018. Battery health prognosis using Brownian motion modeling and particle filtering. *IEEE Trans. Ind. Electron.* 65, 8646–8655.
- [45] Dorr, R., Kratz, F., Ragot, J., Loisy, F., Germain, J.-L., 1997. Detection, isolation, and identification of sensor faults in nuclear power plants. *IEEE Trans. Control Syst. Technol.* 5, 42–60.
- [46] Eshleman, R.L., 1990. Detection, diagnosis and prognosis: An evaluation of current technology. *Curr. Pract. Trends Mech. Fail. Prev.* 33–42.
- [47] Eysers, D.R., Potter, A.T., 2017. Industrial Additive Manufacturing: A manufacturing systems perspective. *Comput. Ind.* 92, 208–218.

- [48] Fernandes, M., Canito, A., Bolón-Canedo, V., Conceição, L., Praça, I., Marreiros, G., 2019. Data analysis and feature selection for predictive maintenance: A case-study in the metallurgic industry. *Int. J. Inf. Manag.* 46, 252–262.
- [49] Ferreiro, S., Arnaiz, A., Sierra, B., Irigoien, I., 2011. A Bayesian network model integrated in a prognostics and health management system for aircraft line maintenance. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* 225, 886–901.
- [50] Frank, A.G., Dalenogare, L.S., Ayala, N.F., 2019. Industry 4.0 technologies: Implementation patterns in manufacturing companies. *Int. J. Prod. Econ.* 210, 15–26.
- [51] Fryman, J., Matthias, B., 2012. Safety of industrial robots: From conventional to collaborative applications, in: *ROBOTIK 2012; 7th German Conference on Robotics*. VDE, pp. 1–5.
- [52] Fumeo, E., Oneto, L., Anguita, D., 2015. Condition Based Maintenance in Railway Transportation Systems Based on Big Data Streaming Analysis., in: *INNS Conference on Big Data*. pp. 437–446.
- [53] Furlanetto, L., Garetti, M., Macchi, M., 2006. *Principi generali di gestione della manutenzione*. FrancoAngeli.
- [54] Gao, Y., Zhang, Q., Kong, X., 2003. Wavelet-based pressure analysis for hydraulic pump health diagnosis. *Trans. ASAE* 46, 969.
- [55] Gattullo, M., Scurati, G.W., Fiorentino, M., Uva, A.E., Ferrise, F., Bordegoni, M., 2019. Towards augmented reality manuals for industry 4.0: A methodology. *Robot. Comput.-Integr. Manuf.* 56, 276–286.
- [56] Ge, Z., 2017. Review on data-driven modeling and monitoring for plant-wide industrial processes. *Chemom. Intell. Lab. Syst.* 171, 16–25.
- [57] Geladi, P., Kowalski, B.R., 1986. Partial least-squares regression: a tutorial. *Anal. Chim. Acta* 185, 1–17.
- [58] Ghasemi, A., Hodkiewicz, M.R., 2012. Estimating mean residual life for a case study of rail wagon bearings. *IEEE Trans. Reliab.* 61, 719–730.
- [59] Gombé, B.O., Mérout, G.G., Breschi, K., Guyennet, H., Friedt, J.-M., Felea, V., Medjaher, K., 2019. A SAW wireless sensor network platform for industrial predictive maintenance. *J. Intell. Manuf.* 30, 1617–1628.
- [60] Gonçalves, L.F., Bosa, J.L., Balen, T.R., Lubaszewski, M.S., Schneider, E.L., Henriques, R.V., 2011. Fault detection, diagnosis and prediction in electrical valves using self-organizing maps. *J. Electron. Test.* 27, 551–564.
- [61] Goodall, P., Sharpe, R., West, A., 2019. A data-driven simulation to support remanufacturing operations. *Comput. Ind.* 105, 48–60.
- [62] Gouriveau, R., Zerhouni, N., 2012. Connexionist-systems-based long term prediction approaches for prognostics. *IEEE Trans. Reliab.* 61, 909–920.
- [63] Granata, F., DE MARINIS, G., 2018. *ALGORITMI DELL'INTELLIGENZA ARTIFICIALE APPLICATI A TEMATICHE IDROLOGICHE E IDRAULICHE*.
- [64] Gravdahl, J.T., Egeland, O., 1999. Centrifugal compressor surge and speed control. *IEEE Trans. Control Syst. Technol.* 7, 567–579.
- [65] Green, D.W., Perry, R.H., 2019. *Perry's Chemical Engineers' Handbook/edición Don W. Green y Robert H. Perry*.
- [66] Gullo, G., 2018. Deep Learning svelato: ecco come funzionano le Reti Neurali Artificiali [WWW Document]. Medium. URL <https://medium.com/professionai/deep-learning-svelato-ecco-come-funzionano-le-reti-neurali-artificiali-f7dced217135> (accessed 10.17.20).
- [67] Gunal, M.M., 2019. *Simulation for Industry 4.0: Past, Present, and Future*. Springer.
- [68] Hancock, K.M., Zhang, Q., 2006. A hybrid approach to hydraulic vane pump condition monitoring and fault detection. *Trans. ASABE* 49, 1203–1211.
- [69] Hao, Y., Helo, P., 2017. The role of wearable devices in meeting the needs of cloud manufacturing: A case study. *Robot. Comput.-Integr. Manuf.* 45, 168–179.
- [70] Hashemian, H.M., 2010. State-of-the-art predictive maintenance techniques. *IEEE Trans. Instrum. Meas.* 60, 226–236.

- [71] He, Y., Han, X., Gu, C., Chen, Z., 2018. Cost-oriented predictive maintenance based on mission reliability state for cyber manufacturing systems. *Adv. Mech. Eng.* 10, 1687814017751467.
- [72] Herzog, M.A., Marwala, T., Heyns, P.S., 2009. Machine and component residual life estimation through the application of neural networks. *Reliab. Eng. Syst. Saf.* 94, 479–489.
- [73] Heydarzadeh, M., Nourani, M., 2016. A two-stage fault detection and isolation platform for industrial systems using residual evaluation. *IEEE Trans. Instrum. Meas.* 65, 2424–2432.
- [74] Holguín, M., Orozco, Á.Á., Holguín, G.A., Álvarez, M., 2016. Optimal state selection and tuning parameters for a degradation model in bearings using Mel-Frequency Cepstral Coefficients and Hidden Markov Chains. *INGENIARE-Rev. Chil. Ing.* 24.
- [75] Hoppenstedt, B., Pryss, R., Stelzer, B., Meyer-Brötz, F., Kammerer, K., Treß, A., Reichert, M., 2018. Techniques and emerging trends for state of the art equipment maintenance systems—a bibliometric analysis. *Appl. Sci.* 8, 916.
- [76] Jennings, C., Wu, D., Terpenney, J., 2016. Forecasting obsolescence risk and product life cycle with machine learning. *IEEE Trans. Compon. Packag. Manuf. Technol.* 6, 1428–1439.
- [77] Jolliffe, I.T., 2003. Principal component analysis. *Technometrics* 45, 276.
- [78] Kalsi, M.S., Eldiwany, B., Sharma, V., Richie, A., 2004. Effect of butterfly valve disc shape variations on torque requirements for power plant applications, in: *Proc. 8th NRC/ASME Symposium on Valve and Pump Testing*, Washington, DC. pp. 1–25.
- [79] Karim, R., Westerberg, J., Galar, D., Kumar, U., 2016. Maintenance analytics—the new know in maintenance. *IFAC-Pap.* 49, 214–219.
- [80] Karkalos, N.E., Markopoulos, A.P., Davim, J.P., 2019. *Computational Methods for Application in Industry 4.0*. Springer.
- [81] Khelif, R., Chebel-Morello, B., Malinowski, S., Laajili, E., Fnaiech, F., Zerhouni, N., 2016. Direct remaining useful life estimation based on support vector regression. *IEEE Trans. Ind. Electron.* 64, 2276–2285.
- [82] Kipper, G., Rampolla, J., 2012. *Augmented Reality: an emerging technologies guide to AR*. Elsevier.
- [83] Kleinbaum, D.G., Kupper, L.L., Nizam, A., Rosenberg, E.S., 2013. *Applied regression analysis and other multivariable methods*. Nelson Education.
- [84] Koch, P.J., van Amstel, M.K., Dębska, P., Thormann, M.A., Tetzlaff, A.J., Bøgh, S., Chrysostomou, D., 2017. A skill-based robot co-worker for industrial maintenance tasks. *Procedia Manuf.* 11, 83–90.
- [85] Kohonen, T., 1990. The self-organizing map. *Proc. IEEE* 78, 1464–1480.
- [86] Kosasih, B.Y., Caesarendra, W., Tieu, K., Widodo, A., Moodie, C.A., Tieu, A.K., 2014. Degradation trend estimation and prognosis of large low speed slewing bearing lifetime, in: *Applied Mechanics and Materials. Trans Tech Publ*, pp. 343–348.
- [87] Krzanowski, W., 2000. *Principles of multivariate analysis*. OUP Oxford.
- [88] Kumar, A., Chinnam, R.B., Tseng, F., 2019. An HMM and polynomial regression based approach for remaining useful life and health state estimation of cutting tools. *Comput. Ind. Eng.* 128, 1008–1014.
- [89] Kumar, S., Purohit, B.S., Manjrekar, V., Singh, V., Lad, B.K., 2018. Investigating the value of integrated operations planning: A case-based approach from automotive industry. *Int. J. Prod. Res.* 56, 6971–6992.
- [90] Laudante, E., 2017. Industry 4.0, Innovation and Design. A new approach for ergonomic analysis in manufacturing system. *Des. J.* 20, S2724–S2734.
- [91] Le, T.T., Bérenguer, C., Chatelain, F., 2015. Multi-branch Hidden semi-Markov modeling for RUL prognosis, in: *2015 Annual Reliability and Maintainability Symposium (RAMS)*. IEEE, pp. 1–6.
- [92] Lee, C.-Y., Huang, T.-S., Liu, M.-K., Lan, C.-Y., 2019. Data science for vibration heteroscedasticity and predictive maintenance of rotary bearings. *Energies* 12, 801.
- [93] Lee, J., Kao, H.-A., Yang, S., 2014. Service innovation and smart analytics for industry 4.0 and big data environment. *Procedia Cirp* 16, 3–8.



- [94] Lee, J., Qiu, H., Ni, J., Djurdjanovic, D., 2004. Infotronics technologies and predictive tools for next-generation maintenance systems. *IFAC Proc.* Vol. 37, 291–302.
- [95] Lei, Y., Li, N., Gontarz, S., Lin, J., Radkowski, S., Dybala, J., 2016. A model-based method for remaining useful life prediction of machinery. *IEEE Trans. Reliab.* 65, 1314–1326.
- [96] Li, C., Liang, M., 2012. Time–frequency signal analysis for gearbox fault diagnosis using a generalized synchrosqueezing transform. *Mech. Syst. Signal Process.* 26, 205–217.
- [97] Li, Naipeng, Lei, Y., Yan, T., Li, Ningbo, Han, T., 2018. A Wiener-process-model-based method for remaining useful life prediction considering unit-to-unit variability. *IEEE Trans. Ind. Electron.* 66, 2092–2101.
- [98] Li, S., Da Xu, L., Zhao, S., 2015. The internet of things: a survey. *Inf. Syst. Front.* 17, 243–259.
- [99] Li, Z., He, Q., 2015. Prediction of railcar remaining useful life by multiple data source fusion. *IEEE Trans. Intell. Transp. Syst.* 16, 2226–2235.
- [100] Li, Z., Wu, D., Hu, C., Terpenney, J., 2019. An ensemble learning-based prognostic approach with degradation-dependent weights for remaining useful life prediction. *Reliab. Eng. Syst. Saf.* 184, 110–122.
- [101] Liao, L., Jin, W., Pavel, R., 2016. Enhanced restricted Boltzmann machine with prognosability regularization for prognostics and health assessment. *IEEE Trans. Ind. Electron.* 63, 7076–7083.
- [102] Liao, L., Köttig, F., 2014. Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction. *IEEE Trans. Reliab.* 63, 191–207.
- [103] Lin, J., Zuo, M.J., 2003. Gearbox fault diagnosis using adaptive wavelet filter. *Mech. Syst. Signal Process.* 17, 1259–1269.
- [104] Lu, Y., 2017. Industry 4.0: A survey on technologies, applications and open research issues. *J. Ind. Inf. Integr.* 6, 1–10.
- [105] Marmo, R., 2020. Algoritmi per l'intelligenza artificiale: Progettazione dell'algoritmo-Dati e Machine Learning-Neural Network-Deep Learning. HOEPLI EDITORE.
- [106] Masoni, R., Ferrise, F., Bordegoni, M., Gattullo, M., Uva, A.E., Fiorentino, M., Carrabba, E., Di Donato, M., 2017. Supporting remote maintenance in industry 4.0 through augmented reality. *Procedia Manuf.* 11, 1296–1302.
- [107] Mathew, V., Toby, T., Singh, V., Rao, B.M., Kumar, M.G., 2017. Prediction of Remaining Useful Lifetime (RUL) of turbofan engine using machine learning, in: 2017 IEEE International Conference on Circuits and Systems (ICCS). IEEE, pp. 306–311.
- [108] McGhee, J., Henderson, I.A., Baird, A., 1997. Neural networks applied for the identification and fault diagnosis of process valves and actuators. *Measurement* 20, 267–275.
- [109] McGhee, M.J., Galloway, G., Catterson, V., Brown, B., Harrison, E., 2014. Prognostic modelling of valve degradation within power stations, in: Annual Conference of the Prognostics and Health Management Society 2014 (PHM).
- [110] Mell, P., Grance, T., 2011. The NIST definition of cloud computing.
- [111] Mobley, R.K., 2002. An introduction to predictive maintenance. Elsevier.
- [112] Monostori, L., Kádár, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., Sauer, O., Schuh, G., Sihn, W., Ueda, K., 2016. Cyber-physical systems in manufacturing. *Cirp Ann.* 65, 621–641.
- [113] Mourtzis, D., Vlachou, E., 2018. A cloud-based cyber-physical system for adaptive shop-floor scheduling and condition-based maintenance. *J. Manuf. Syst.* 47, 179–198.
- [114] Nandi, S., Toliyat, H.A., 1999. Condition monitoring and fault diagnosis of electrical machines-a review, in: Conference Record of the 1999 IEEE Industry Applications Conference. Thirty-Forth IAS Annual Meeting (Cat. No. 99CH36370). IEEE, pp. 197–204.
- [115] Nguyen, K.T., Medjaher, K., 2019. A new dynamic predictive maintenance framework using deep learning for failure prognostics. *Reliab. Eng. Syst. Saf.* 188, 251–262.
- [116] Ogata, K., Yang, Y., 2002. Modern control engineering. Prentice hall India.
- [117] Oliveira, R., Farinha, J.T., Fonseca, I., Barbosa, F.M., 2016. Augmented reality system for maintenance of high-voltage systems, in: 2016 51st International Universities Power Engineering Conference (UPEC). IEEE, pp. 1–5.

- [118] Omar, F.K., Gaouda, A.M., 2009. Gearbox Diagnostics Using Wavelet-Based Windowing, in: *Journal of Physics: Conference Series*. p. 012089.
- [119] Oztemel, E., Gursev, S., 2020. Literature review of Industry 4.0 and related technologies. *J. Intell. Manuf.* 31, 127–182.
- [120] Pang, C.K., Zhou, J.-H., Yan, H.-C., 2014. PDF and breakdown time prediction for unobservable wear using enhanced particle filters in precognitive maintenance. *IEEE Trans. Instrum. Meas.* 64, 649–659.
- [121] Paprocka, I., 2019. Evaluation of the Effects of a Machine Failure on the Robustness of a Job Shop System—Proactive Approaches. *Sustainability* 11, 65.
- [122] Parrondo, J.L., Velarde, S., Santolaria, C., 1998. Development of a predictive maintenance system for a centrifugal pump. *J. Qual. Maint. Eng.*
- [123] Peng, Y., Hou, Y., Song, Y., Pang, J., Liu, D., 2018. Lithium-ion battery prognostics with hybrid Gaussian process function regression. *Energies* 11, 1420.
- [124] Peres, R.S., Rocha, A.D., Leitao, P., Barata, J., 2018. IDARTS—Towards intelligent data analysis and real-time supervision for industry 4.0. *Comput. Ind.* 101, 138–146.
- [125] Popkova, E.G., Ragulina, Y.V., Bogoviz, A.V., 2019. Fundamental differences of transition to industry 4.0 from previous industrial revolutions, in: *Industry 4.0: Industrial Revolution of the 21st Century*. Springer, pp. 21–29.
- [126] Popovic, D., Bhatkar, V.P., 1990. *Distributed computer control systems in industrial automation*. CRC Press.
- [127] Pusey, H.C., 2007. Turbomachinery condition monitoring and failure prognosis. *Sound Vib.* 41, 10–15.
- [128] Ribeiro, D.A., Ferreira, D.D., Pereira, D.A., Junior, R.A.B., Nunes, R.D., 2019. Mechanical fault detection in electric motors measured by a digital signal processing device in an optical mouse. *Measurement* 138, 350–355.
- [129] Roblek, V., Meško, M., Krapež, A., 2016. A complex view of industry 4.0. *Sage Open* 6, 2158244016653987.
- [130] Rødseth, H., Schjøberg, P., Marhaug, A., 2017. Deep digital maintenance. *Adv. Manuf.* 5, 299–310.
- [131] Roy, R., Stark, R., Tracht, K., Takata, S., Mori, M., 2016. Continuous maintenance and the future—Foundations and technological challenges. *Cirp Ann.* 65, 667–688.
- [132] Sait, A.S., Sharaf-Eldeen, Y.I., 2011. A review of gearbox condition monitoring based on vibration analysis techniques diagnostics and prognostics, in: *Rotating Machinery, Structural Health Monitoring, Shock and Vibration, Volume 5*. Springer, pp. 307–324.
- [133] Sakthivel, N.R., Sugumaran, V., Babudevasenapati, Sjes., 2010. Vibration based fault diagnosis of monoblock centrifugal pump using decision tree. *Expert Syst. Appl.* 37, 4040–4049.
- [134] Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C., 2001. Estimating the support of a high-dimensional distribution. *Neural Comput.* 13, 1443–1471.
- [135] Seber, G.A., 2009. *Multivariate observations*. John Wiley & Sons.
- [136] Sezer, E., Romero, D., Guedea, F., Macchi, M., Emmanouilidis, C., 2018. An industry 4.0-enabled low cost predictive maintenance approach for smes, in: *2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*. IEEE, pp. 1–8.
- [137] Shafi, U., Safi, A., Shahid, A.R., Ziauddin, S., Saleem, M.Q., 2018. Vehicle remote health monitoring and prognostic maintenance system. *J. Adv. Transp.* 2018.
- [138] Sharma, A.B., Golubchik, L., Govindan, R., 2010. Sensor faults: Detection methods and prevalence in real-world datasets. *ACM Trans. Sens. Netw. TOSN* 6, 1–39.
- [139] Shi, Y.-T., Zhang, Y., Xiang, X., Wang, L., Lei, Z.-W., Sun, D.-H., 2018. Stochastic Hybrid Estimator Based Fault Detection and Isolation for Wind Energy Conversion Systems with Unknown Fault Inputs. *Energies* 11, 2227.
- [140] Shrouf, F., Miragliotta, G., 2015. Energy management based on Internet of Things: practices and framework for adoption in production management. *J. Clean. Prod.* 100, 235–246.

- 
- [141] Si, X.-S., Wang, W., Hu, C.-H., Zhou, D.-H., 2011. Remaining useful life estimation—a review on the statistical data driven approaches. *Eur. J. Oper. Res.* 213, 1–14.
- [142] Siddiqui, K.M., Sahay, K., Giri, V.K., 2014. Health monitoring and fault diagnosis in induction motor—a review. *Int. J. Adv. Res. Electr. Electron. Instrum. Eng.* 3, 6549–6565.
- [143] Susto, G.A., Beghi, A., De Luca, C., 2012. A predictive maintenance system for epitaxy processes based on filtering and prediction techniques. *IEEE Trans. Semicond. Manuf.* 25, 638–649.
- [144] Susto, G.A., Schirru, A., Pampuri, S., Beghi, A., De Nicolao, G., 2018. A hidden-Gamma model-based filtering and prediction approach for monotonic health factors in manufacturing. *Control Eng. Pract.* 74, 84–94.
- [145] Tiraboschi, M., Seghezzi, F., 2016. Il Piano nazionale Industria 4.0: una lettura lavoristica.
- [146] Vaidya, S., Ambad, P., Bhosle, S., 2018. Industry 4.0—a glimpse. *Procedia Manuf.* 20, 233–238.
- [147] Verrilli, F., Srinivasan, S., Gambino, G., Canelli, M., Himanka, M., Del Vecchio, C., Sasso, M., Glielmo, L., 2016. Model predictive control-based optimal operations of district heating system with thermal energy storage and flexible loads. *IEEE Trans. Autom. Sci. Eng.* 14, 547–557.
- [148] Vianna, W.O.L., Yoneyama, T., 2017. Predictive maintenance optimization for aircraft redundant systems subjected to multiple wear profiles. *IEEE Syst. J.* 12, 1170–1181.
- [149] Wagner, T., Herrmann, C., Thiede, S., 2017. Industry 4.0 impacts on lean production systems. *Procedia Cirp* 63, 125–131.
- [150] Wang, C., Xu, J., Wang, H., Zhang, Z., 2018. Condition-based predictive order model for a mechanical component following Inverse Gaussian degradation process. *Math. Probl. Eng.* 2018.
- [151] Wang, D., Tsui, K.-L., Miao, Q., 2017. Prognostics and health management: A review of vibration based bearing and gear health indicators. *Ieee Access* 6, 665–676.
- [152] Wang, J., Song, Y., Liu, F., Hou, R., 2016. Analysis and application of forecasting models in wind power integration: A review of multi-step-ahead wind speed forecasting models. *Renew. Sustain. Energy Rev.* 60, 960–981.
- [153] Wang, K., Guo, H., Xu, A., Jameson, N.J., Pecht, M., Yan, B., 2018. Creating self-aware low-voltage electromagnetic coils for incipient insulation degradation monitoring for smart manufacturing. *IEEE Access* 6, 69860–69868.
- [154] Wen, Y., Wu, J., Zhou, Q., Tseng, T.-L., 2018. Multiple-change-point modeling and exact Bayesian inference of degradation signal for prognostic improvement. *IEEE Trans. Autom. Sci. Eng.* 16, 613–628.
- [155] Wilson, D.L., Coyle, J.R., Thomas, E.A., 2017. Ensemble machine learning and forecasting can achieve 99% uptime for rural handpumps. *Plos One* 12, e0188808.
- [156] Wu, D., Jennings, C., Terpenney, J., Gao, R.X., Kumara, S., 2017. A comparative study on machine learning algorithms for smart manufacturing: tool wear prediction using random forests. *J. Manuf. Sci. Eng.* 139.
- [157] Wu, X., Li, Y., 2012. Self-learning based centrifugal compressor surge mapping with computationally efficient adaptive asymmetric support vector machine. *J. Dyn. Syst. Meas. Control* 134.
- [158] Wu, X., Zhu, X., Wu, G.-Q., Ding, W., 2013. Data mining with big data. *IEEE Trans. Knowl. Data Eng.* 26, 97–107.
- [159] Wu, Z., Luo, H., Yang, Y., Lv, P., Zhu, X., Ji, Y., Wu, B., 2018. K-PdM: KPI-Oriented Machinery Deterioration Estimation Framework for Predictive Maintenance Using Cluster-Based Hidden Markov Model. *IEEE Access* 6, 41676–41687.
- [160] Xiao, F., Zheng, C., Wang, S., 2011. A fault detection and diagnosis strategy with enhanced sensitivity for centrifugal chillers. *Appl. Therm. Eng.* 31, 3963–3970.
- [161] Xiao, W., 2016. A probabilistic machine learning approach to detect industrial plant faults. *ArXiv Prepr. ArXiv160305770*.
- [162] Xiaoli, X., Yunbo, Z., Guoxin, W., 2011. Design of intelligent internet of things for equipment maintenance, in: 2011 Fourth International Conference on Intelligent Computation Technology and Automation. *IEEE*, pp. 509–511.
-

- [163] Xu, Y., Zhang, Y., Zhang, S., 2015. Uncertain generalized remaining useful life prediction-driven predictive maintenance decision, in: 2015 Prognostics and System Health Management Conference (PHM). IEEE, pp. 1–6.
- [164] Yan, J., Meng, Y., Lu, L., Li, L., 2017. Industrial big data in an industry 4.0 environment: Challenges, schemes, and applications for predictive maintenance. *IEEE Access* 5, 23484–23491.
- [165] Yang, C., Lou, Q., Liu, J., Gou, H., Bai, Y., 2015. Particle filter-based approach to estimate remaining useful life for predictive maintenance, in: *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer, pp. 692–701.
- [166] Yildirim, M., Gebraeel, N.Z., Sun, X.A., 2017. Integrated predictive analytics and optimization for opportunistic maintenance and operations in wind farms. *IEEE Trans. Power Syst.* 32, 4319–4328.
- [167] Zhong, R.Y., Xu, X., Klotz, E., Newman, S.T., 2017. Intelligent manufacturing in the context of industry 4.0: a review. *Engineering* 3, 616–630.
- [168] Zhou, K., Liu, T., Zhou, L., 2015. Industry 4.0: Towards future industrial opportunities and challenges, in: 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD). IEEE, pp. 2147–2152.
- [169] Zhou, L., Pan, S., Wang, J., Vasilakos, A.V., 2017. Machine learning on big data: Opportunities and challenges. *Neurocomputing* 237, 350–361.

# Appendice 1 - Algoritmo ARIMA

## ARIMA training

```

function
[meango,stdevgo]=arima_training(GUIID,tag,path,datago1,datago2,threshold,
nndata,estpoint,fail_duration,sampling_time)
%
%   value of passed variable for testing (to delete)
%
%   tag='SG_RA2_AQ_LT_501C'
%   path='C:\Users\navicelli\Documents\MATLAB\Smartgeo\data\archivi
annuali\rep';
%   save_path='C:\Users\navicelli\Documents\MATLAB\Smartgeo\Prove
Arima';
%   datago1 = '01/01/2017 00:00:00';
%   datago2 = '15/02/2017 00:00:00';
%
%   GUID := unique identifier of diagnostic element instance in AF
%   tag := tag name to which the diagnostic is applied
%   path := path for saved historical data of tag
%   datago1 := starting timestamp for training data interval
format:'dd/mm/yyyy HH:MM:SS'
%   datago2 := ending timestamp for training data interval
format:'dd/mm/yyyy HH:MM:SS'
%   threshold := threshold/s value/s for tag value
%   nndata := data number for arima training
%   estpoint := data number for arima prediction (used to evaluate the
%   necessity of re-sampling the data vector)
%   fail_duration := time duration of the fail event [s] (used to
evaluate the
%   necessity of re-sampling the data vector)
%   sampling_time := sampling time interval -> nel caso di varAM=1
verrà modificato
%           e assumerà il significato di bucket temporale per il
calcolo della MA
%
current_dir=pwd;
%
%   home_dir := root directory for AF element data
%
home_dir = 'C:\Users\navicelli\Documents\MATLAB\Smartgeo\ARIMA rev';

% % Purging data with timestamp earlier than 2013
Lvl_va=Sensor(tag,path);
[times,values]=load(Lvl_va);
datestart=datenum(datago1,'dd/mm/yyyy HH:MM:SS');
dateend=datenum(datago2,'dd/mm/yyyy HH:MM:SS');
godata1=find(times>=datestart);
godata2=find(times<=dateend);
govalues=values(godata1(1):godata2(end));

%

```

```
%Computing average and standard deviation for good operating
conditions
%
meango = mean(govalues);
stdevgo = std(govalues);

%
%   evaluation of the necessity of re-sampling the data vector
%

if fail_duration / (estpoint + ndata) > 2 * sampling_time
    varMA = 1;
    ma_time = fail_duration / (estpoint + ndata);
    MAlag = round(ma_time / sampling_time);
    sampling_time = MAlag * sampling_time;
else
    varMA = 0;
    MAlag = 1;
end
p_estpoint = estpoint;
p_ndata = ndata;
p_threshold = threshold;
%
%   Check the presence of the GUID folder in the save path,
%   if not available create the folder and save in it the
training_file
%
if exist(strcat(home_dir, '\', GUID), 'dir')
    cd(strcat(home_dir, '\', GUID));
else
    cd(home_dir);
    mkdir(GUID);
    cd(strcat(home_dir, '\', GUID));
end
save('training_file', 'p_estpoint', 'p_ndata', 'p_threshold', 'sampling_time',
'MAlag', 'varMA');
if isfile('history_file.mat') == 1           % if exists delete it
    delete history_file.mat
end
cd(current_dir);
```

## ARIMA init

```
function [] = arima_init(GUID,method,limit)
%
%   GUID := GUID of the AF element instance
%   method := selection of the method for the evaluation of the ARIMA
%   parameters
%   limit := upper limit of p and q ARIMA parameter
%
current_dir=pwd;
%
%   home_dir := root directory for AF element data
%
```

---

```

home_dir = 'C:\Users\navicelli\Documents\MATLAB\Smartgeo\ARIMA_rev';
%
cd(strcat(home_dir,'\',GUID));
load('training_file.mat');
load('history_file.mat');

if length(val)>= ndata
%Re-sampling data according to training evaluation (varMA=1)
    if varMA == 0
        val_ma = val;
    elseif varMA == 1
        valma = tsmovavg(val(2,:), 's',MAlag);
        val_ma = [val(1,MAlag:MAlag:end);
valma(MAlag:MAlag:length(valma))];
    end
end
if length(val_ma) >= ndata

    xfit = 1:length(val_ma);
    fitvars = polyfit(xfit,val_ma(2,:),1);

    if fitvars(2) >= 0
% Parametrization of ARIMA model (p,D,q)
D = 1; % parameter D is a boolean assuming value 1 if the process is
not stationary

        if method==1

            [acf,alags,abounds] =
autocorr(val_ma(2,:), 'NumLags',limit);
            [pacf,plags,pbounds] =
parcorr(val_ma(2,:), 'NumLags',limit);
            sa = 0;
            i = 1;

            while acf(i) <= bounds(1) && acf(i) >= bounds(2) && i <=
length(acf)
                p = alags(i);
                sa = sa+1;
                i=i+1;
            end

            % assign the maximum value to p if no significant lag
values exist
            if sa == 0
                p = alags(end);
            end

            sp = 0;
            i = 1;

            while pacf(i) <= pbounds(1) && pacf(i) >= pbounds(2) && i
<= length(pacf)
                q = plags(i);
                sp = sp+1;
                i=i+1;
            end
end

```

---

```

        % assign the maximum value to q if no significant lag
values exist
    if sp == 0
        q = plags(end);
    end

    elseif method==2
        minAIC=10^12;
        minBIC=10^12;
        for i = 1:limit
            for j =1:limit
                Mdl = arima(i,D,j);
                try
                    EstMdl =
estimate(Mdl, val_ma(2, :)', 'Display', 'off');
                    summ = summarize(EstMdl);
                    AIC=summ.AIC;
                    BIC=summ.BIC;
                    if AIC<minAIC
                        minAIC=AIC;
                        opt_aic_p=i;
                        opt_aic_q=j;
                    end
                    if BIC<minBIC
                        minBIC=BIC;
                        opt_bic_p=i;
                        opt_bic_q=j;
                    end

                    catch ME
                        fprintf('init ARIMA without success: %s\n',
ME.message);
                        continue; % Jump to next iteration of: for i
                    end
                end
            end
        end

        if minAIC < minBIC
            p=opt_aic_p;
            q=opt_aic_q;
        else
            p=opt_bic_p;
            q=opt_bic_q;
        end
    end
else
    p = 0;
    D = 0;
    q = 0;
end
else
    p = 0;
    D = 0;
    q = 0;
end
end

```



```

cd(strcat(home_dir, '\', GUID));
save('history_file', 'ndata', 'val', 'val_ma', 'p', 'D', 'q');
cd(current_dir);

```

## ARIMA realtime

```

function
[earlyT, meanT, lateT, varMAT]=arima_realtime(GUID, newval, estpoint, thresh
old, ndata, nsigma)
%
%   varMAT is the returned variable assuming 3 possible values
%   varMAT=1 -> possible diagnosys, values valid ;
%   varMAT=2 -> possible diagnosys, values not valid (forecast
unfeasible);
%   varMAT=3 -> initialization needed (too short value history);
%   varMAT=4 -> offline training needed -> raising alarm
%
%   meanT := expected value for forecasted threshold trespassing time
%   earlyT := 5? percentile value for forecasted threshold trespassing
time
%   (based on nsigma value)
%   lateT := 95? percentile value for forecasted threshold trespassing
time
%   (based on nsigma value)
%
%   GUID := GUID of the AF element instance
%   newval := present value for tag to be diasgnosed
%   estpoint := data number for arima prediction
%   threshold := trespassing this value (upper, lower or upper and
lower limit) is considered a failure
%   ndata := data number for arima training
%   nsigma := number of st.dev for forcasted threshold
%
current_dir=pwd;
%
%   home_dir := root directory for AF element data
%
home_dir = 'C:\Users\navicelli\Documents\MATLAB\Smartgeo\ARIMA rev';
%
%   all the data needed for an instance of a diagnostic elements are
in the
%   subdirectory named with the GUID of the element instance in the
root
%   directory for AF element data
%
%   First of all, check the existance of the GUID subdirectory
%
if exist(strcat(home_dir, '\', GUID), 'dir')      % if exists change
directory to it
    cd(strcat(home_dir, '\', GUID));
else                                             % else change to the home
directory
    cd(home_dir);
    mkdir(GUID);                               % create GUID subdirectory

```

---

```

        cd(strcat(home_dir, '\', GUID));           % change to the created
subdicrectory
end
%
% Then check if the default file containing the trained output
parameter exists
%
if isfile('training_file.mat') == 1           % if exists load it
    load('training_file.mat');
    %
    % Check if the input parameters are still the same of those used
for
    % training the diagnostic algorithm
    %
    if p_ndata==ndata && p_estpoint==estpoint &&
p_threshold==threshold
        %
        % If the input parameters are the same then it is possible
to use
        % the trained output parameters
        %
        % Check if history of data exists
        %
        if isfile('history_file.mat') == 1
            load('history_file.mat');
            %
            % Check if the history of data available is up to date
and
            % sufficient
            %
            if newval(1) - val(1,end) < 10 * sampling_time
                %10*sampling_time tolerated period for valid init

                if p~=0 && D~=0 && q~=0

                    val(1,end+1) = newval(1);
                    val(2,end)= newval(2);
                    val = [val(1,2:end) ; val(2,2:end)];

                    if varMA == 0
                        val_ma = val;
                    elseif varMA == 1
                        valma = tsmovavg(val(2,:), 's', MAlag);
                        val_ma = [val(1, MAlag: MAlag:end);
valma(MAlag: MAlag:length(valma))];
                    end

                    xfit = 1:length(val_ma);
                    fitvars = polyfit(xfit, val_ma(2,:), 1);

                    if length(threshold) == 1
                        if threshold >= val_ma(2,end) && fitvars(1) >=
0
                            valfit = 11;
                        elseif threshold <= val_ma(2,end) &&
fitvars(1) <= 0
                            valfit = 12;

```

---

```

        else
            valfit = 0;
        end
    elseif length(threshold) == 2
        minth = min(threshold);
        maxth = max(threshold);
        if val_ma(2,end) <= maxth && val_ma(2,end) >=
minth
            if fitvars(1) >= 0
                valfit = 21;
            elseif fitvars(1) <=0
                valfit = 22;
            end
        else
            valfit = 0;
        end
    end
end

if valfit ~= 0

    try
        Mdl = arima(p,D,q);
        time = val_ma(1,:)' ;
        y = val_ma(2,:)' ;
        T = length(y);
        EstMdl = estimate(Mdl,y,'Display','off');
        [yF,yMSE] =
forecast(EstMdl,estpoint,'Y0',y);
        upper = yF + nsigma * sqrt(yMSE);
        lower = yF - nsigma * sqrt(yMSE);
        endestT = ((time(T) +
sampling_time):sampling_time:(time(T) + estpoint * sampling_time))';

        if valfit == 11 || valfit == 21
            earlyT =
interp1(upper,endestT,max(threshold));
            lateT =
interp1(lower,endestT,max(threshold));
            meanT =
interp1(yF,endestT,max(threshold));
        elseif valfit == 12 || valfit == 22
            earlyT =
interp1(lower,endestT,min(threshold));
            lateT =
interp1(upper,endestT,min(threshold));
            meanT =
interp1(yF,endestT,min(threshold));
        end

    catch ME
        fprintf('ARIMA without success: %s\n',
ME.message);

        earlyT = NaN;
        lateT = NaN;
        meanT = NaN;
    end
end

```

```
isnan(lateT)                if isnan(earlyT) && isnan(meanT) &&
                            varMAT = 2;
                            else
                                varMAT = 1;
                            end

                            else
                                varMAT = 3;
                                earlyT = NaN;
                                lateT = NaN;
                                meanT = NaN;
                            end

save('history_file','ndata','val','val_ma','p','D','q');
    else
        varMAT = 3;
        val(1,end+1) = newval(1);
        val(2,end) = newval(2);
        earlyT=NaN;
        lateT=NaN;
        meanT=NaN;

save('history_file','ndata','val','val_ma','p','D','q');
    end

    else
        varMAT = 3;
        val = [newval(1);newval(2)];
        val_ma = val;
        earlyT=NaN;
        lateT=NaN;
        meanT=NaN;

save('history_file','ndata','val','val_ma','p','D','q');
    end

    else
        varMAT = 3;
        val = [newval(1);newval(2)];
        val_ma = val;
        p = 0;
        D = 0;
        q = 0;
        earlyT=NaN;
        lateT=NaN;
        meanT=NaN;
        save('history_file','ndata','val','val_ma','p','D','q');
    end

    else
        varMAT = 4;
        earlyT=NaN;
        lateT=NaN;
        meanT=NaN;

end
```

```
else
  varMAT = 4;
  earlyT=NaN;
  lateT=NaN;
  meanT=NaN;

end
cd(current_dir)
```

# Appendice 2 – Prototipo di applicativo prognostico

## Training del modello SVM

```

classdef APP_SmartGEOInfluxdb_1_2tables_update < matlab.apps.AppBase

% Properties that correspond to app components
properties (Access = public)
    UIFigure matlab.ui.Figure
    EsisteunmodelloSVMaddestratoButtonGroup matlab.ui.container.ButtonGroup
    SIButton matlab.ui.control.RadioButton
    NOButton matlab.ui.control.RadioButton
    InviaButton matlab.ui.control.Button
    AddestraButton matlab.ui.control.Button
    FineAddestramentoButton matlab.ui.control.Button
    AccuratezzaLabel matlab.ui.control.Label
    EditField matlab.ui.control.NumericEditField
    CommentiTextArea matlab.ui.control.TextArea
    ProvaConnessioneButton matlab.ui.control.Button
    ConnessionealdbInfluxLabel matlab.ui.control.Label
    KernelListBoxLabel matlab.ui.control.Label
    KernelListBox matlab.ui.control.ListBox
    save_dirEditFieldLabel matlab.ui.control.Label
    save_dirEditField matlab.ui.control.EditField
    home_dirEditFieldLabel matlab.ui.control.Label
    home_dirEditField matlab.ui.control.EditField
    URLEditFieldLabel matlab.ui.control.Label
    URLEditField matlab.ui.control.EditField
    USEREditFieldLabel matlab.ui.control.Label
    USEREditField matlab.ui.control.EditField
    PASSWORDEditFieldLabel matlab.ui.control.Label
    PASSWORDEditField matlab.ui.control.EditField
    DATABASEEditFieldLabel matlab.ui.control.Label
    DATABASEEditField matlab.ui.control.EditField
    UITable matlab.ui.control.Table
    Button matlab.ui.control.Button
    Button_2 matlab.ui.control.Button
    FormatodataggmmaaaahhmssRegimeLabel0AnomaliaLabel1 matlab.ui.control.Label
    FormatodataggmmaaaahhmssLabel_2 matlab.ui.control.Label
    UITable2 matlab.ui.control.Table
    Button_3 matlab.ui.control.Button
    Button_4 matlab.ui.control.Button
    FormatodataggmmaaaahhmssRegimeLabel0AnomaliaLabel1_2 matlab.ui.control.Label
    CreaTraining_setButton matlab.ui.control.Button
    BilanciaTraining_setButton matlab.ui.control.Button
    Sbilanciamentototraining_setEditFieldLabel matlab.ui.control.Label

```

```
Sbilanciamentotrainig_setEditField matlab.ui.control.NumericEditField
Ncampionitrainig_setEditFieldLabel matlab.ui.control.Label
Ncampionitrainig_setEditField matlab.ui.control.NumericEditField
end
```

```
% Callbacks that handle component events
```

```
methods (Access = private)
```

```
% Button pushed function: ProvaConnessioneButton
```

```
function ProvaConnessioneButtonPushed(app, event)
```

```
URL = app.URLEditField.Value;
```

```
USER = app.USEREditField.Value;
```

```
PASS = app.PASSWORDEditField.Value;
```

```
DATABASE = app.DATABASEEditField.Value;
```

```
influxdb = InfluxDB(URL, USER, PASS, DATABASE);
```

```
% Controlla lo stato dell'istanza InfluxDB
```

```
[ok, ping] = influxdb.ping();
```

```
if ok==1
```

```
pingstr=num2str(ping);
```

```
pingstr=strcat('ping =',pingstr);
```

```
Txt={'connessione effettuata';pingstr};
```

```
app.CommentiTextArea.Value=Txt;
```

```
else
```

```
app.CommentiTextArea.Value='impossibile connettersi';
```

```
return
```

```
end
```

```
% Cambia il database corrente
```

```
try
```

```
influxdb.use(app.DATABASEEditField.Value);
```

```
catch ME
```

```
app.CommentiTextArea.Value{end+1,1}=ME.message;
```

```
return
```

```
end
```

```
app.SIButton.Enable = 'on';
```

```
app.NOButton.Enable = 'on';
```

```
app.InviaButton.Enable = 'on';
```

```
GUID='DATA_FOLDER';
```

```
funName = mfilename;
```

```
p = mfilename('fullpath');
```

```
home_dir=strrep(p,funName,'');
```

```
app.home_dirEditField.Value = home_dir;
```

```
cd(home_dir)
```

```
if exist(strcat(home_dir,GUID),'dir') == 0
```

```
cd(home_dir);
```

```
mkdir(GUID);
```

```
end
```

```
% save_dir = strcat(home_dir,GUID);
```

```
% app.save_dirEditField.Value = save_dir;
```

```
% cd(save_dir)
```

```
% save('influx_connection','URL','USER','PASS','DATABASE')
```

```
% cd(home_dir)
```

```
series = Series('InfluxConn') ...
```

```
.fields('URLurl', app.URLEditField.Value, 'USERuser', app.USEREditField.Value,
'PASSpass', app.PASSWORDEditField.Value, 'DBdb', app.PASSWORDEditField.Value) ...
.time(datetime('01/01/2020 00:00:00','InputFormat', 'dd,MM,yyyy hh:mm:ss',
'TimeZone', 'local'));
try
influxdb.writer() ...
.append(series) ...
.execute();
catch ME
app.CommentiTextArea.Value{end+1,1}=ME.message;
return
end
end
```

```
% Button pushed function: InviaButton
function InviaButtonPushed(app, event)
% Si fa sempre in modo di salvare tutti i dati nella cartella DataFolder
(save_dir), mentre si svolgono i calcoli nella directory principale (home_dir)
% Se viene premuto il pulsante SI, allora si addestra automaticamente SVM
partendo dal Training Set già a disposizione
if app.SIButton.Value == 1
try
[filenameS1,pathS1] = uigetfile();
cd(pathS1)
load(filenameS1)
cd(home_dir)
catch ME
app.CommentiTextArea.Value{end+1,1}=ME.message;
return
end
```

```
app.UIFigure.Visible = 'off';
app.CommentiTextArea.Value = 'Training completato';
APP_SmartGEOInfluxdb_2
app.SIButton.Enable = 'off';
app.NOButton.Enable = 'off';
app.InviaButton.Enable = 'off';
else
% Se, invece, viene premuto il pulsante NO allora l'utente deve compilare i campi
che appaiono
```

```
app.UITable.Enable = 'on';
app.UITable2.Enable = 'on';
app.Button.Enable = 'on';
app.Button_3.Enable = 'on';
app.CreaTraining_setButton.Enable = 'on';
app.UITable.Data = cell(1,length(app.UITable.ColumnNames));
app.UITable2.Data = cell(2,length(app.UITable2.ColumnNames));
end
end
```



```

% Button pushed function: AddestraButton
function AddestraButtonPushed(app, event)
% L'utente deve specificare la directory del file Excel contenente il Training
Set
% Se la directory che inserisce è sbagliata, compare a video la scritta che
indica l'errore
%Scelta del kernel e dell'eventuale ordine del polinomio di addestramento
%in base alla listBox
cd(app.save_dirEditField.Value)
load('training_file_SVM','training_set','variabili_utili','A','B')
cd(app.home_dirEditField.Value)
Items = app.KernellistBox.Items;
idx = find(strcmp([Items], app.KernellistBox.Value));
if idx == 1
kernel_function = 'gaussian';
polyord = [];
elseif idx == 2
kernel_function = 'linear';
polyord = [];
elseif idx == 3
kernel_function = 'polynomial';
polyord =2;
else
kernel_function = 'polynomial';
polyord =3;
end
[trainedClassifier, validationAccuracy] =
trainClassifier(training_set,kernel_function,polyord);
app.EditField.Value = validationAccuracy;
app.CommentiTextArea.Value{end+1,1} = 'Training completato';
[~,score] = predict(trainedClassifier.ClassificationSVM,training_set(:,2:end));
max_int = max(score(:,1));
min_int = min(score(:,1));
% A questo punto si salvano in Data_Folder il file di Training Set, la funzione
di allenamento e le colonne degli attributi.
cd(app.save_dirEditField.Value)
save('training_file_SVM','trainedClassifier','training_set','variabili_utili','A'
,'B','max_int','min_int')
cd(app.home_dirEditField.Value)
app.FineAddestramentoButton.Enable = 'on';
clear variables
% Quando l'addestramento è finito, si deve aprire la seconda APP
%%% attenzione, la seconda APP si deve aprire solo se davvero i label sono 0 o 1
% A quel punto si deve disabilitare la prima APP
end

% Button pushed function: FineAddestramentoButton
function FineAddestramentoButtonPushed(app, event)
APP_SmartGEOInfluxdb_2
app.FineAddestramentoButton.Enable = 'off';
app.AddestraButton.Enable = 'off';
app.CommentiTextArea.Enable = 'off';
app.KernellistBox.Enable = 'off';

```

```
app.InviaButton.Enable = 'off';
app.SIButton.Enable = 'off';
app.NOButton.Enable = 'off';
end

% Button pushed function: Button
function ButtonPushed(app, event)
app.UITable.Data{end+1,1}='';
app.Button_2.Enable = 'on';
end

% Button pushed function: Button_2
function Button_2Pushed(app, event)
app.UITable.Data(end,:)=[];
if size(app.UITable.Data,1)==1
app.Button_2.Enable = 'off';
end
end

% Button pushed function: Button_3
function Button_3Pushed(app, event)
app.UITable2.Data{end+1,1}='';
app.Button_4.Enable = 'on';
end

% Button pushed function: Button_4
function Button_4Pushed(app, event)
app.UITable2.Data(end,:)=[];
if size(app.UITable2.Data,1)==2
app.Button_4.Enable = 'off';
end
end

% Button pushed function: CreaTraining_setButton
function CreaTraining_setButtonPushed(app, event)
URL = app.URLEditField.Value;
USER = app.USEREditField.Value;
PASS = app.PASSWORDEditField.Value;
DATABASE = app.DATABASEEditField.Value;
influxdb = InfluxDB(URL, USER, PASS, DATABASE);
A = app.UITable.Data;
B = app.UITable2.Data;
dataset_name{1,1} = 'Label1';
variabili_utili = A(:,3)';
dataset_name=[dataset_name, variabili_utili];
%fill the dataset with the series
Lperiod = size(B,1);
Ltable = size(A,1);
for i = 1:Lperiod
```

```

LABEL = B{i,3};
dataset{i,1} = LABEL;
dataset_time{i,1} = LABEL;
for j=1:Ltable
DB = A{j,1};
SERIES = A{j,2};
FIELDS = A{j,3};
time_in = B{i,1};
time_fn = B{i,2};
try
influxdb.use(DB);
result = influxdb.query(SERIES)...
.fields(FIELDS)...
.after(datetime(time_in, 'InputFormat', 'dd/MM/yyyy HH:mm:ss'))...
.before(datetime(time_fn, 'InputFormat', 'dd/MM/yyyy HH:mm:ss'))...
.execute();
data_var = result.series(SERIES);
data_field = data_var.field(FIELDS);
time = result.series.time();
catch ME
app.CommentiTextArea.Value{end+1,1} = strcat('Controllare i dati inseriti nelle
tabelle!:',ME.message);
return
end
idx_u = strcmp(dataset_name,FIELDS);
idx = find(idx_u == 1);
dataset{i,idx} = data_field;
dataset_time{i,idx} = time;
end
end
%Sincronizzazione degli eventuali tags non sincronizzati
r_ds = size(dataset,1);
c_ds = size(dataset,2);
dataset_sync = dataset(:,[1,2]);
if c_ds >= 3
for i = 1:r_ds
for j = 3:c_ds
dataset_sync{i,j} = interp1(dataset_time{i,j},dataset{i,j},dataset_time{i,2});
end
end
end
%Creazione del dataset nel formato di addestramento
if c_ds>=3
for i=1:r_ds
lbl_set = zeros(size(dataset_sync{i,2}));
lbl_set(:,1) = str2double(dataset_sync{i,1});
tr_set = [lbl_set, dataset_sync{i,2}];
for j=3:c_ds
tr_set = [tr_set, dataset_sync{i,j}];
end
dtst_tr{i,1} = tr_set;
end
training_set = dtst_tr{1,1};
for i=2:size(dtst_tr,1)

```

```

training_set = [training_set; dtst_tr{i,1}];
end
else
lbl_set = zeros(size(dataset_sync{1,2}));
lbl_set(:,1) = str2double(dataset_sync{1,1});
training_set = [lbl_set, dataset_sync{1,2}];
end
cd(app.save_dirEditField.Value)
save('training_file_SVM', 'training_set', 'variabili_utili', 'A', 'B')
cd(app.home_dirEditField.Value)
n_anomali = length(find(training_set(:,1)==1));
n_trset = size(training_set,1);
sбилan = n_anomali./n_trset*100;
app.Sbilanciamentotrainig_setEditField.Value = sbilan;
app.Ncampionitrainig_setEditField.Value = n_trset;
if n_trset <= 1000
app.Ncampionitrainig_setEditField.BackgroundColor = [1.00 1.00 0.07];
end
if sbilan >=60 || sbilan<=40
app.BilanciaTraining_setButton.Enable = 'on';
app.Sbilanciamentotrainig_setEditField.BackgroundColor = [1.00 1.00 0.07];
if sbilan >=80 || sbilan <= 20
app.Sbilanciamentotrainig_setEditField.BackgroundColor = [1.00 0.00 0.00];
end
else
app.BilanciaTraining_setButton.Enable = 'off';
app.Sbilanciamentotrainig_setEditField.BackgroundColor = [1.00 1.00 1.00];
end
app.KernellistBox.Enable = 'on';
app.KernellistBoxLabel.Enable = 'on';
app.AddestraButton.Enable = 'on';
app.AccuratezzaLabel.Enable = 'on';
app.EditField.Enable = 'on';
end

```

% Button pushed function: BilanciaTraining\_setButton

```

function BilanciaTraining_setButtonPushed(app, event)
cd(app.save_dirEditField.Value)
load('training_file_SVM', 'training_set', 'variabili_utili', 'A', 'B')
cd(app.home_dirEditField.Value)
anomali = find(training_set(:,1)==1);
n_anomali = length(anomali);
regime = find(training_set(:,1)==0);
n_regime = length(regime);
if n_anomali > n_regime
y = datasample(anomali,n_regime,1, 'Replace', false);
training_set = [training_set(y,:); training_set(regime,:)];
sбилan = 100*length(y)/size(training_set,1);
else
y = datasample(regime,n_anomali,1, 'Replace', false);
training_set = [training_set(y,:); training_set(anomali,:)];
sбилan = 100*n_anomali/size(training_set,1);
end

```

```

cd(app.save_dirEditField.Value)
save('training_file_SVM','training_set','variabili_utili','A','B')
cd(app.home_dirEditField.Value)
app.Ncampionitraining_setEditField.Value = size(training_set,1);
if size(training_set,1) <= 1000
app.Ncampionitraining_setEditField.BackgroundColor = [1.00 1.00 0.07];
end
app.Sbilanciamentotraining_setEditField.Value = sbilan;
app.Sbilanciamentotraining_setEditField.BackgroundColor = [1.00 1.00 1.00];
app.BilanciaTraining_setButton.Enable = 'off';
end
end

```

```

% Component initialization
methods (Access = private)

```

```

% Create UIFigure and components
function createComponents(app)

```

```

% Create UIFigure and hide until all components are created
app.UIFigure = uifigure('Visible', 'off');
app.UIFigure.Color = [1 1 1];
app.UIFigure.Position = [100 100 763 708];
app.UIFigure.Name = 'UI Figure';

```

```

% Create EsisteunmodelloSVMaddestratoButtonGroup
app.EsisteunmodelloSVMaddestratoButtonGroup = uibuttongroup(app.UIFigure);
app.EsisteunmodelloSVMaddestratoButtonGroup.TitlePosition = 'centertop';
app.EsisteunmodelloSVMaddestratoButtonGroup.Title = 'Esiste un modello SVM
addestrato?';
app.EsisteunmodelloSVMaddestratoButtonGroup.BackgroundColor = [0.5882 0.8353 1];
app.EsisteunmodelloSVMaddestratoButtonGroup.FontName = 'Calibri Light';
app.EsisteunmodelloSVMaddestratoButtonGroup.FontSize = 14;
app.EsisteunmodelloSVMaddestratoButtonGroup.Position = [11 449 743 67];

```

```

% Create SIButton
app.SIButton = uiradiobutton(app.EsisteunmodelloSVMaddestratoButtonGroup);
app.SIButton.Enable = 'off';
app.SIButton.Text = 'SI';
app.SIButton.FontName = 'Calibri Light';
app.SIButton.FontSize = 14;
app.SIButton.Position = [83 22 58 22];
app.SIButton.Value = true;

```

```

% Create NOButton
app.NOButton = uiradiobutton(app.EsisteunmodelloSVMaddestratoButtonGroup);
app.NOButton.Enable = 'off';
app.NOButton.Text = 'NO';

```

```
app.NOButton.FontName = 'Calibri Light';
app.NOButton.FontSize = 14;
app.NOButton.Position = [83 1 65 22];

% Create InviaButton
app.InviaButton = uibutton(app.EsisteunmodelloSVMaddestratoButtonGroup, 'push');
app.InviaButton.ButtonPushedFcn = createCallbackFcn(app, @InviaButtonPushed,
true);
app.InviaButton.BackgroundColor = [1 1 1];
app.InviaButton.FontName = 'Calibri Light';
app.InviaButton.FontSize = 14;
app.InviaButton.FontWeight = 'bold';
app.InviaButton.Enable = 'off';
app.InviaButton.Position = [298 10 100 26];
app.InviaButton.Text = 'Invia';

% Create AddestraButton
app.AddestraButton = uibutton(app.UIFigure, 'push');
app.AddestraButton.ButtonPushedFcn = createCallbackFcn(app,
@AddestraButtonPushed, true);
app.AddestraButton.FontName = 'Calibri Light';
app.AddestraButton.FontSize = 14;
app.AddestraButton.Enable = 'off';
app.AddestraButton.Position = [209 75 137 26];
app.AddestraButton.Text = 'Addestra';

% Create FineAddestramentoButton
app.FineAddestramentoButton = uibutton(app.UIFigure, 'push');
app.FineAddestramentoButton.ButtonPushedFcn = createCallbackFcn(app,
@FineAddestramentoButtonPushed, true);
app.FineAddestramentoButton.FontName = 'Calibri Light';
app.FineAddestramentoButton.FontSize = 14;
app.FineAddestramentoButton.Enable = 'off';
app.FineAddestramentoButton.Position = [474 13 247 34];
app.FineAddestramentoButton.Text = 'Fine Addestramento';

% Create AccuratezzaLabel
app.AccuratezzaLabel = uilabel(app.UIFigure);
app.AccuratezzaLabel.BackgroundColor = [0.5882 0.8392 1];
app.AccuratezzaLabel.FontName = 'Calibri Light';
app.AccuratezzaLabel.FontSize = 24;
app.AccuratezzaLabel.FontWeight = 'bold';
app.AccuratezzaLabel.Enable = 'off';
app.AccuratezzaLabel.Position = [474 59 247 42];
app.AccuratezzaLabel.Text = ' Accuratezza';

% Create EditField
app.EditField = uieditfield(app.UIFigure, 'numeric');
app.EditField.Editable = 'off';
```

```

app.EditField.FontName = 'Calibri Light';
app.EditField.FontSize = 24;
app.EditField.Enable = 'off';
app.EditField.Position = [612 64 100 31.6000003814697];

% Create CommentiTextArea
app.CommentiTextArea = uitextarea(app.UIFigure);
app.CommentiTextArea.Editable = 'off';
app.CommentiTextArea.Position = [350 533 404 162];

% Create ProvaConnessioneButton
app.ProvaConnessioneButton = uibutton(app.UIFigure, 'push');
app.ProvaConnessioneButton.ButtonPushedFcn = createCallbackFcn(app,
@ProvaConnessioneButtonPushed, true);
app.ProvaConnessioneButton.BackgroundColor = [1 1 1];
app.ProvaConnessioneButton.FontName = 'Calibri Light';
app.ProvaConnessioneButton.FontSize = 14;
app.ProvaConnessioneButton.Position = [11 533 299 26];
app.ProvaConnessioneButton.Text = 'Prova Connessione';

% Create ConnessionealdbInfluxLabel
app.ConnessionealdbInfluxLabel = uilabel(app.UIFigure);
app.ConnessionealdbInfluxLabel.FontName = 'Calibri Light';
app.ConnessionealdbInfluxLabel.FontSize = 16;
app.ConnessionealdbInfluxLabel.FontWeight = 'bold';
app.ConnessionealdbInfluxLabel.Position = [83 673 161 22];
app.ConnessionealdbInfluxLabel.Text = 'Connessione al db Influx';

% Create KernellistBoxLabel
app.KernellistBoxLabel = uilabel(app.UIFigure);
app.KernellistBoxLabel.HorizontalAlignment = 'right';
app.KernellistBoxLabel.FontName = 'Calibri Light';
app.KernellistBoxLabel.FontSize = 14;
app.KernellistBoxLabel.Position = [14 77 41 22];
app.KernellistBoxLabel.Text = 'Kernel';

% Create KernellistBox
app.KernellistBox = uilistbox(app.UIFigure);
app.KernellistBox.Items = {'Gaussiano', 'Lineare', 'Polinomiale 2', 'Polinomiale
3'};
app.KernellistBox.Enable = 'off';
app.KernellistBox.FontName = 'Calibri Light';
app.KernellistBox.FontSize = 14;
app.KernellistBox.Position = [70 11 126 90];
app.KernellistBox.Value = 'Gaussiano';

% Create save_dirEditFieldLabel
app.save_dirEditFieldLabel = uilabel(app.UIFigure);

```

```
app.save_dirEditFieldLabel.HorizontalAlignment = 'right';
app.save_dirEditFieldLabel.Visible = 'off';
app.save_dirEditFieldLabel.Position = [53 494 51 22];
app.save_dirEditFieldLabel.Text = 'save_dir';

% Create save_dirEditField
app.save_dirEditField = uieditfield(app.UIFigure, 'text');
app.save_dirEditField.Editable = 'off';
app.save_dirEditField.Enable = 'off';
app.save_dirEditField.Visible = 'off';
app.save_dirEditField.Position = [119 494 100 22];

% Create home_dirEditFieldLabel
app.home_dirEditFieldLabel = uilabel(app.UIFigure);
app.home_dirEditFieldLabel.HorizontalAlignment = 'right';
app.home_dirEditFieldLabel.Enable = 'off';
app.home_dirEditFieldLabel.Visible = 'off';
app.home_dirEditFieldLabel.Position = [255 494 55 22];
app.home_dirEditFieldLabel.Text = 'home_dir';

% Create home_dirEditField
app.home_dirEditField = uieditfield(app.UIFigure, 'text');
app.home_dirEditField.Enable = 'off';
app.home_dirEditField.Visible = 'off';
app.home_dirEditField.Position = [325 494 100 22];

% Create URLEditFieldLabel
app.URLEditFieldLabel = uilabel(app.UIFigure);
app.URLEditFieldLabel.HorizontalAlignment = 'right';
app.URLEditFieldLabel.FontName = 'Calibri Light';
app.URLEditFieldLabel.FontSize = 14;
app.URLEditFieldLabel.Position = [40 644 27 22];
app.URLEditFieldLabel.Text = 'URL';

% Create URLEditField
app.URLEditField = uieditfield(app.UIFigure, 'text');
app.URLEditField.FontName = 'Calibri Light';
app.URLEditField.FontSize = 14;
app.URLEditField.Position = [110 643 200 23];

% Create USEREditFieldLabel
app.USEREditFieldLabel = uilabel(app.UIFigure);
app.USEREditFieldLabel.HorizontalAlignment = 'right';
app.USEREditFieldLabel.FontName = 'Calibri Light';
app.USEREditFieldLabel.FontSize = 14;
app.USEREditFieldLabel.Position = [39 618 35 22];
app.USEREditFieldLabel.Text = 'USER';
```



```
% Create USEREditField
app.USEREditField = uieditfield(app.UIFigure, 'text');
app.USEREditField.FontName = 'Calibri Light';
app.USEREditField.FontSize = 14;
app.USEREditField.Position = [110 617 200 23];

% Create PASSWORDEditFieldLabel
app.PASSWORDEditFieldLabel = uilabel(app.UIFigure);
app.PASSWORDEditFieldLabel.HorizontalAlignment = 'right';
app.PASSWORDEditFieldLabel.FontName = 'Calibri Light';
app.PASSWORDEditFieldLabel.FontSize = 14;
app.PASSWORDEditFieldLabel.Position = [17 592 69 22];
app.PASSWORDEditFieldLabel.Text = 'PASSWORD';

% Create PASSWORDEditField
app.PASSWORDEditField = uieditfield(app.UIFigure, 'text');
app.PASSWORDEditField.FontName = 'Calibri Light';
app.PASSWORDEditField.FontSize = 14;
app.PASSWORDEditField.Position = [110 591 200 23];

% Create DATABASEEditFieldLabel
app.DATABASEEditFieldLabel = uilabel(app.UIFigure);
app.DATABASEEditFieldLabel.HorizontalAlignment = 'right';
app.DATABASEEditFieldLabel.FontName = 'Calibri Light';
app.DATABASEEditFieldLabel.FontSize = 14;
app.DATABASEEditFieldLabel.Position = [22 567 63 22];
app.DATABASEEditFieldLabel.Text = 'DATABASE';

% Create DATABASEEditField
app.DATABASEEditField = uieditfield(app.UIFigure, 'text');
app.DATABASEEditField.FontName = 'Calibri Light';
app.DATABASEEditField.FontSize = 14;
app.DATABASEEditField.Position = [110 566 200 23];

% Create UITable
app.UITable = uitable(app.UIFigure);
app.UITable.ColumnName = {'Database'; 'Serie'; 'Field'};
app.UITable.RowName = {};
app.UITable.ColumnEditable = true;
app.UITable.Enable = 'off';
app.UITable.FontName = 'Calibri Light';
app.UITable.FontSize = 14;
app.UITable.Position = [15 242 321 140];

% Create Button
app.Button = uibutton(app.UIFigure, 'push');
app.Button.ButtonPushedFcn = createCallbackFcn(app, @ButtonPushed, true);
app.Button.BackgroundColor = [1 1 1];
```

```
app.Button.FontWeight = 'bold';
app.Button.Enable = 'off';
app.Button.Position = [345 357 26 22];
app.Button.Text = '+';

% Create Button_2
app.Button_2 = uibutton(app.UIFigure, 'push');
app.Button_2.ButtonPushedFcn = createCallbackFcn(app, @Button_2Pushed, true);
app.Button_2.BackgroundColor = [1 1 1];
app.Button_2.FontWeight = 'bold';
app.Button_2.Enable = 'off';
app.Button_2.Position = [345 333 26 22];
app.Button_2.Text = '-';

% Create FormatodataggmmaaahhmmssRegimeLabel0AnomaliaLabel1
app.FormatodataggmmaaahhmmssRegimeLabel0AnomaliaLabel1 = uilabel(app.UIFigure);
app.FormatodataggmmaaahhmmssRegimeLabel0AnomaliaLabel1.FontName = 'Calibri
Light';
app.FormatodataggmmaaahhmmssRegimeLabel0AnomaliaLabel1.FontWeight = 'bold';
app.FormatodataggmmaaahhmmssRegimeLabel0AnomaliaLabel1.Position = [396 381 318
29];
app.FormatodataggmmaaahhmmssRegimeLabel0AnomaliaLabel1.Text = {'Formato data:
gg/mm/aaaa hh:mm:ss Regime: Label = 0'; ' Anomalia: Label = 1'};

% Create FormatodataggmmaaahhmmssLabel_2
app.FormatodataggmmaaahhmmssLabel_2 = uilabel(app.UIFigure);
app.FormatodataggmmaaahhmmssLabel_2.FontName = 'Calibri Light';
app.FormatodataggmmaaahhmmssLabel_2.FontSize = 14;
app.FormatodataggmmaaahhmmssLabel_2.FontWeight = 'bold';
app.FormatodataggmmaaahhmmssLabel_2.Position = [94 409 153 22];
app.FormatodataggmmaaahhmmssLabel_2.Text = 'Variabili di addestramento';

% Create UITable2
app.UITable2 = uitable(app.UIFigure);
app.UITable2.ColumnName = {'Inizio (Time)'; 'Fine (Time)'; 'Label'};
app.UITable2.RowName = {};
app.UITable2.ColumnEditable = true;
app.UITable2.Enable = 'off';
app.UITable2.FontName = 'Calibri Light';
app.UITable2.FontSize = 14;
app.UITable2.Position = [389 242 332 140];

% Create Button_3
app.Button_3 = uibutton(app.UIFigure, 'push');
app.Button_3.ButtonPushedFcn = createCallbackFcn(app, @Button_3Pushed, true);
app.Button_3.BackgroundColor = [1 1 1];
app.Button_3.FontWeight = 'bold';
app.Button_3.Enable = 'off';
app.Button_3.Position = [728 357 26 22];
```

```

app.Button_3.Text = '+';

% Create Button_4
app.Button_4 = uibutton(app.UIFigure, 'push');
app.Button_4.ButtonPushedFcn = createCallbackFcn(app, @Button_4Pushed, true);
app.Button_4.BackgroundColor = [1 1 1];
app.Button_4.FontWeight = 'bold';
app.Button_4.Enable = 'off';
app.Button_4.Position = [728 333 26 22];
app.Button_4.Text = '-';

% Create FormatodataggmmaaaahhmmssRegimeLabel0AnomaliaLabel1_2
app.FormatodataggmmaaaahhmmssRegimeLabel0AnomaliaLabel1_2 =
uilabel(app.UIFigure);
app.FormatodataggmmaaaahhmmssRegimeLabel0AnomaliaLabel1_2.FontName = 'Calibri
Light';
app.FormatodataggmmaaaahhmmssRegimeLabel0AnomaliaLabel1_2.FontSize = 14;
app.FormatodataggmmaaaahhmmssRegimeLabel0AnomaliaLabel1_2.FontWeight = 'bold';
app.FormatodataggmmaaaahhmmssRegimeLabel0AnomaliaLabel1_2.Position = [514 409 147
22];
app.FormatodataggmmaaaahhmmssRegimeLabel0AnomaliaLabel1_2.Text = 'Periodi di
addestramento';

% Create CreaTraining_setButton
app.CreaTraining_setButton = uibutton(app.UIFigure, 'push');
app.CreaTraining_setButton.ButtonPushedFcn = createCallbackFcn(app,
@CreaTraining_setButtonPushed, true);
app.CreaTraining_setButton.FontName = 'Calibri Light';
app.CreaTraining_setButton.FontSize = 14;
app.CreaTraining_setButton.Enable = 'off';
app.CreaTraining_setButton.Position = [255 204 220 26];
app.CreaTraining_setButton.Text = 'Crea Training_set';

% Create BilanciaTraining_setButton
app.BilanciaTraining_setButton = uibutton(app.UIFigure, 'push');
app.BilanciaTraining_setButton.ButtonPushedFcn = createCallbackFcn(app,
@BilanciaTraining_setButtonPushed, true);
    app.BilanciaTraining_setButton.FontName = 'Calibri Light';
    app.BilanciaTraining_setButton.FontSize = 14;
    app.BilanciaTraining_setButton.Enable = 'off';
    app.BilanciaTraining_setButton.Position = [473 126 140 52];
    app.BilanciaTraining_setButton.Text = 'Bilancia Training_set';

% Create Sbilanciamentottraining_setEditFieldLabel
app.Sbilanciamentottraining_setEditFieldLabel = uilabel(app.UIFigure);
app.Sbilanciamentottraining_setEditFieldLabel.HorizontalAlignment =
'right';
app.Sbilanciamentottraining_setEditFieldLabel.FontName = 'Calibri
Light';

```

```

    app.Sbilanciamentottraining_setEditFieldLabel.FontSize = 16;
    app.Sbilanciamentottraining_setEditFieldLabel.FontWeight = 'bold';
    app.Sbilanciamentottraining_setEditFieldLabel.Position = [90 129 184
22];
    app.Sbilanciamentottraining_setEditFieldLabel.Text = 'Sbilanciamento
training_set';

    % Create Sbilanciamentottraining_setEditField
    app.Sbilanciamentottraining_setEditField = uieditfield(app.UIFigure,
'numeric');
    app.Sbilanciamentottraining_setEditField.Editable = 'off';
    app.Sbilanciamentottraining_setEditField.FontName = 'Calibri Light';
    app.Sbilanciamentottraining_setEditField.FontSize = 16;
    app.Sbilanciamentottraining_setEditField.Position = [288 126 100
23.6000003814697];

    % Create Ncampionittraining_setEditFieldLabel
    app.Ncampionittraining_setEditFieldLabel = uilabel(app.UIFigure);
    app.Ncampionittraining_setEditFieldLabel.HorizontalAlignment =
'right';
    app.Ncampionittraining_setEditFieldLabel.FontName = 'Calibri Light';
    app.Ncampionittraining_setEditFieldLabel.FontSize = 16;
    app.Ncampionittraining_setEditFieldLabel.FontWeight = 'bold';
    app.Ncampionittraining_setEditFieldLabel.Position = [110 158 163 22];
    app.Ncampionittraining_setEditFieldLabel.Text = 'N° campioni
training_set';

    % Create Ncampionittraining_setEditField
    app.Ncampionittraining_setEditField = uieditfield(app.UIFigure,
'numeric');
    app.Ncampionittraining_setEditField.FontName = 'Calibri Light';
    app.Ncampionittraining_setEditField.FontSize = 16;
    app.Ncampionittraining_setEditField.Position = [288 156 100
23.6000003814697];

    % Show the figure after all components are created
    app.UIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

    % Construct app
    function app = APP_SmartGEOInfluxdb_1_2tables_update

        % Create UIFigure and components

```

```

        createComponents(app)

        % Register the app with App Designer
        registerApp(app, app.UIFigure)

        if nargin == 0
            clear app
        end
    end

    % Code that executes before app deletion
    function delete(app)

        % Delete UIFigure when app is deleted
        delete(app.UIFigure)
    end
end
end
end
end

```

## Analisi prognostica real-time

```
classdef APP_SmartGEOInfluxdb_2_update < matlab.apps.AppBase
```

```

% Properties that correspond to app components
properties (Access = public)
    UIFigure matlab.ui.Figure
    StartButton matlab.ui.control.StateButton
    EditField3Label matlab.ui.control.Label
    EditField3 matlab.ui.control.NumericEditField
    StopButton matlab.ui.control.Button
    TextArea matlab.ui.control.TextArea
    UIAxes matlab.ui.control.UIAxes
    TempodicampionamentoSpinnerLabel matlab.ui.control.Label
    TempodicampionamentoSpinner matlab.ui.control.Spinner
    TalpiprestoDatePickerLabel matlab.ui.control.Label
    TalpiprestoDatePicker matlab.ui.control.DatePicker
    TmedioDatePickerLabel matlab.ui.control.Label
    TmedioDatePicker matlab.ui.control.DatePicker
    TalpitardiDatePickerLabel matlab.ui.control.Label
    TalpitardiDatePicker matlab.ui.control.DatePicker
    OutputdelmodelloLabel matlab.ui.control.Label
    InputdelmodelloLabel matlab.ui.control.Label
    npuntiproiettatiEditFieldLabel matlab.ui.control.Label
    npuntiproiettatiEditField matlab.ui.control.NumericEditField
    sogliaEditFieldLabel matlab.ui.control.Label
    sogliaEditField matlab.ui.control.NumericEditField
    ndatiEditFieldLabel matlab.ui.control.Label

```

```

ndatiEditField matlab.ui.control.NumericEditField
nsigmaLabel matlab.ui.control.Label
nsigmaEditField matlab.ui.control.NumericEditField
metodoSpinnerLabel matlab.ui.control.Label
metodoSpinner matlab.ui.control.Spinner
limiteEditFieldLabel matlab.ui.control.Label
limiteEditField matlab.ui.control.NumericEditField
EtichettaEditFieldLabel matlab.ui.control.Label
EtichettaEditField matlab.ui.control.NumericEditField
DegradoprestazioniEditFieldLabel matlab.ui.control.Label
DegradoprestazioniEditField matlab.ui.control.NumericEditField
savedirEditFieldLabel matlab.ui.control.Label
savedirEditField matlab.ui.control.EditField
homedirEditFieldLabel matlab.ui.control.Label
homedirEditField matlab.ui.control.EditField
SaveDb matlab.ui.control.EditField
SaveSeries matlab.ui.control.EditField
DatabaseLabel matlab.ui.control.Label
SeriesLabel matlab.ui.control.Label
OutputdellaprevisionesuInfluxLabel matlab.ui.control.Label
end

```

```

% Callbacks that handle component events
methods (Access = private)

```

```

% Value changed function: StartButton
function StartButtonValueChanged(app, event)
GUID = 'DATA_FOLDER';
funName = mfilename;
p = mfilename('fullpath');
home_dir=strrep(p,funName,'');
app.homedirEditField.Value = home_dir;
save_dir = strcat(home_dir,GUID);
% app.savedirEditField.Value = save_dir;
% cd(save_dir)
% load('training_file_SVM.mat','A','B','max_int','min_int','trainedClassifier')
app.TextArea.Value='';
scatter(app.UIAxes,datetime('now'),1,'white')
result = influxdb.query('InfluxConn') ...
.fields('URLLur1', 'USERuser', 'PASSpass', 'DBdb') ...
.execute();
InfluxConn = result.series('InfluxConn');
URL = InfluxConn.field('URLLur1');
USER = InfluxConn.field('USERuser');
PASS = InfluxConn.field('PASSpass');
DATABASE = InfluxConn.field('DBdb');
% Directory di salvataggio dei dati e di lavoro
% save_dir = app.savedirEditField.Value;
% home_dir = app.homedirEditField.Value;
% cd(save_dir)
% load('influx_connection','URL', 'USER', 'PASS', 'DATABASE')
% cd(home_dir)

```

```

influxdb = InfluxDB(URL, USER, PASS, DATABASE);
[ok, ~] = influxdb.ping();
if ok==0
app.TextArea.Value{end+1,1}='Connessione ad Influxdb persa!';
return
end
varMA = 0;
MAlag = 1;
sampling_time=app.TempodicampionamentoSpinner.Value/24/3600;
p_estpoint = app.npuntiproiettatiEditField.Value;
p_ndata = app.ndatiEditField.Value;
p_threshold = app.sogliaEditField.Value;
series = Series('Training_file_ARIMA') ...
.tags('city', 'barcelona', 'country', 'catalonia') ...
.fields('p_estpoint', p_estpoint, 'p_ndata', p_ndata, 'p_threshold', p_threshold)
...
.time(datetime('01/01/2020 00:00:00','InputFormat', 'dd,MM,yyyy hh:mm:ss',
'TimeZone', 'local')));
try
influxdb.writer() ...
.append(series) ...
.execute();
catch ME
app.CommentiTextArea.Value{end+1,1}=ME.message;
return
end
% cd(save_dir)
%
save('training_file_ARIMA','p_estpoint','p_ndata','p_threshold','sampling_time','
MAlag','varMA');
% if isfile('history_file.mat') == 1
% delete history_file.mat
% end
% %tutte le restanti operazioni vengono svolte fuori dalla cartella GUID
% cd(home_dir)
stop_app=1;
app.EditField3.Value=stop_app;
estpoint=app.npuntiproiettatiEditField.Value;
threshold=app.sogliaEditField.Value; %threshold è il valore di score t.c. se le
previsioni di ARIMA superano quel valore allora il componente si guasta
ndata=app.ndatiEditField.Value;
nsigma=app.nsigmaEditField.Value;
method=app.metodoSpinner.Value;
limit=app.limiteEditField.Value;
cd(save_dir)
load('training_file_SVM.mat');
cd(home_dir)
while stop_app==1
p1=now;
v_tot = zeros(1,length(variabili_utili)); % questa è un'inizializzazione
for i=1:length(variabili_utili)
DB = A{i,1};
SERIES = A{i,2};
FIELDS = A{i,3};

```

```

try
influxdb.use(DB);
result = influxdb.query(SERIES)...
.fields(FIELDS)...
.after(datetime(datetime(p1-
2*sampling_time, 'ConvertFrom', 'datetime', 'TimeZone', 'local')))...
.execute();
data_var = result.series(SERIES);
data_field = data_var.field(FIELDS);
time = result.series.time();
catch ME
app.TextArea.Value{end+1,1}=strcat('Il sampling_time scelto potrebbe essere
troppo piccolo rispetto a quello di scrittura di nuovi valori sul DB Influx:
',ME.message);
return
end
v=data_field(end);
t_dt = time(end);
t=datetime(t_dt);
v_tot(1,i)=v;
end
% Si carica SVM allenato e si prevede un nuovo valore (label). Label dice se il
componente funziona bene (0) o è in anomalia (1)
% Si calcola anche la distanza del valore dall'ipersuperficie di separazione
(score)
% Score è un vettore di due numeri uguali in modulo, ma opposti in segno. Noi
prendiamo solo il primo numero in valore assoluto e lo imponiamo come valore di
distanza
% La distanza viene poi convertita in un valore numerico compreso tra 0 e 1
(score_prob)
% Tale valore non è una probabilità, ma dà un'idea di quanto il punto rilevato
sia vicino alla superficie di separazione tra classe 0( buon funzionamento) e
classe 1
[label,score]=predict(trainedClassifier.ClassificationSVM,v_tot);
if score(1)>=max_int
score_prob = 1;
elseif score(1)<=min_int
score_prob = -1;
else
if score(1) >= 0
score_prob = 1/max_int*score(1);
else
score_prob = -1/min_int*score(1);
end
end
if label==0
score_prob_disp = 0;
else
score_prob_disp = abs(score_prob);
end
% Ora si scrive newval che è il nuovo valore acquisito real time e viene usato
per fare la previsione ARIMA
app.EtichettaEditField.Value = label;
app.DegradoprestazioniEditField.Value = round(score_prob_disp,2);

```



```

newval=[t,score_prob_disp];
influxdb.use(app.SaveDb.Value);
Series1 = Series(app.SaveSeries.Value) ...
.fields('degrado', score_prob_disp, 'Label', label) ...
.time(t_dt);
influxdb.writer() ...
.append(Series1) ...
.execute();
% Da qui in poi parte ARIMA, che viene applicata solo quando siamo in stato di
anomalia (label=1)
if label==1
app.TextArea.Value{end+1,1}='Individuato valore che suggerisce anomalia';
cd(save_dir)
% Si controlla se il file che di default contiene il modello di ARIMA allenato a
prevedere gli output esiste
% Se tale modello esiste, allora si carica
[earlyT,meanT,lateT,varMAT,message_err1]=arima_realtime(save_dir,newval,estpoint,
threshold,ndata,nsigma);
app.TextArea.Value{end+1,1} = message_err1;
if varMAT == 1
app.TextArea.Value{end+1,1}='Modello ARIMA: previsione completata. RUL calcolata
correttamente';
EarlyRUL = earlyT - t;
MeanRUL = meanT - t;
LateRUL = lateT - t;
Series1 = Series(app.SaveSeries.Value) ...
.fields('EarlyRUL', EarlyRUL, 'MeanRUL', MeanRUL, 'LateRUL', LateRUL) ...
.time(t_dt);
influxdb.writer() ...
.append(Series1) ...
.execute();
elseif varMAT == 2
app.TextArea.Value{end+1,1}='Modello ARIMA: previsione completata. Impossibile
calcolare RUL (provare ad aumentare il n° punti proiettati)';
elseif varMAT == 3
app.TextArea.Value{end+1,1}='Modello ARIMA: necessaria inizializzazione';
end
%Grafico nell'applicativo della previsione ARIMA
dateET = datetime(earlyT,'ConvertFrom','datetime','TimeZone','local');
dateMT = datetime(meanT,'ConvertFrom','datetime','TimeZone','local');
dateLT = datetime(lateT,'ConvertFrom','datetime','TimeZone','local');
app.TalpiprestoDatePicker.Value = dateET;
app.TmedioDatePicker.Value = dateMT;
app.TalpitardiDatePicker.Value = dateLT;
if isnat(dateET)==0
scatter(app.UIAxes,dateET,1,100,'filled','vb')
hold(app.UIAxes, 'on')
end
if isnat(dateMT)==0
scatter(app.UIAxes,dateMT,1,100,'filled','vr')
hold(app.UIAxes, 'on')
end
if isnat(dateLT)==0
scatter(app.UIAxes,dateLT,1,100,'filled','vb')

```

```

end
hold(app.UIAxes, 'off')
if varMAT==3
app.TextArea.Value{end+1,1}='Modello ARIMA model: inizializzazione avviata';
[message_err] = arima_init(save_dir,method,limit);
app.TextArea.Value{end+1,1} = message_err;
elseif varMAT==4
app.TextArea.Value{end+1,1}='Il modello ARIMA deve essere allenato!!!';
end
end
%controllo il tempo necessario all'algorithmo a leggere le righe precedenti;
%se troppo veloce lo ritardo per arrivare al sampling_time, altrimenti il ritardo
è nullo
p2=now;
p3=p2-p1;
if p3<=sampling_time
pause((sampling_time-p3)*24*3600)
end
stop_app=app.EditField3.Value;
end
cd(save_dir);
delete history_file.mat
delete training_file_ARIMA.mat
cd(home_dir);
scatter(app.UIAxes,datetime('now'),1,'white')
app.TalpiprestoDatePicker.Value = NaT;
app.TmedioDatePicker.Value = NaT;
app.TalpitardiDatePicker.Value = NaT;
end

% Button pushed function: StopButton
function StopButtonPushed(app, event)
app.EditField3.Value=0;
end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

% Create UIFigure and hide until all components are created
app.UIFigure = uifigure('Visible', 'off');
app.UIFigure.Color = [1 1 1];
app.UIFigure.Position = [100 100 744 682];
app.UIFigure.Name = 'UI Figure';

% Create StartButton

```

```
app.StartButton = uibutton(app.UIFigure, 'state');
app.StartButton.ValueChangedFcn = createCallbackFcn(app,
@StartButtonValueChanged, true);
app.StartButton.Text = 'Start';
app.StartButton.BackgroundColor = [0.5608 0.7882 0.3961];
app.StartButton.FontName = 'Calibri Light';
app.StartButton.FontSize = 14;
app.StartButton.FontWeight = 'bold';
app.StartButton.Position = [44 358 126 32];

% Create EditField3Label
app.EditField3Label = uilabel(app.UIFigure);
app.EditField3Label.HorizontalAlignment = 'right';
app.EditField3Label.Visible = 'off';
app.EditField3Label.Position = [323 648 62 22];
app.EditField3Label.Text = 'Edit Field3';

% Create EditField3
app.EditField3 = uieditfield(app.UIFigure, 'numeric');
app.EditField3.Editable = 'off';
app.EditField3.Visible = 'off';
app.EditField3.Position = [296 648 10 22];

% Create StopButton
app.StopButton = uibutton(app.UIFigure, 'push');
app.StopButton.ButtonPushedFcn = createCallbackFcn(app, @StopButtonPushed, true);
app.StopButton.BackgroundColor = [0.9216 0.5608 0.5608];
app.StopButton.FontName = 'Calibri Light';
app.StopButton.FontSize = 14;
app.StopButton.FontWeight = 'bold';
app.StopButton.Position = [245 358 126 32];
app.StopButton.Text = 'Stop';

% Create TextArea
app.TextArea = uitextarea(app.UIFigure);
app.TextArea.FontName = 'Calibri Light';
app.TextArea.Position = [419 358 304 309];

% Create UIAxes
app.UIAxes = uiaxes(app.UIFigure);
title(app.UIAxes, 'Data di guasto attesa')
ylabel(app.UIAxes, '')
app.UIAxes.PlotBoxAspectRatio = [5.99082568807339 1 1];
app.UIAxes.FontName = 'Calibri Light';
app.UIAxes.FontSize = 14;
app.UIAxes.MinorGridLineStyle = '--';
app.UIAxes.Box = 'on';
app.UIAxes.YTick = [];
app.UIAxes.XGrid = 'on';
```

```
app.UIAxes.BackgroundColor = [1 1 1];
app.UIAxes.Position = [1 17 722 214];
```

#### % Create TempodicampionamentoSpinnerLabel

```
app.TempodicampionamentoSpinnerLabel = uilabel(app.UIFigure);
app.TempodicampionamentoSpinnerLabel.HorizontalAlignment = 'right';
app.TempodicampionamentoSpinnerLabel.FontName = 'Calibri Light';
app.TempodicampionamentoSpinnerLabel.FontSize = 14;
app.TempodicampionamentoSpinnerLabel.Position = [32 502 152 22];
app.TempodicampionamentoSpinnerLabel.Text = 'Tempo di campionamento';
```

#### % Create TempodicampionamentoSpinner

```
app.TempodicampionamentoSpinner = uispinner(app.UIFigure);
app.TempodicampionamentoSpinner.Limits = [0 Inf];
app.TempodicampionamentoSpinner.FontName = 'Calibri Light';
app.TempodicampionamentoSpinner.FontSize = 14;
app.TempodicampionamentoSpinner.Position = [208 502 168 22];
app.TempodicampionamentoSpinner.Value = 1;
```

#### % Create TalpiprestoDatePickerLabel

```
app.TalpiprestoDatePickerLabel = uilabel(app.UIFigure);
app.TalpiprestoDatePickerLabel.HorizontalAlignment = 'right';
app.TalpiprestoDatePickerLabel.FontName = 'Calibri Light';
app.TalpiprestoDatePickerLabel.FontSize = 14;
app.TalpiprestoDatePickerLabel.Position = [52 287 119 26];
app.TalpiprestoDatePickerLabel.Text = 'T al più presto';
```

#### % Create TalpiprestoDatePicker

```
app.TalpiprestoDatePicker = uideatepicker(app.UIFigure);
app.TalpiprestoDatePicker.Editable = 'off';
app.TalpiprestoDatePicker.FontName = 'Calibri Light';
app.TalpiprestoDatePicker.FontSize = 14;
app.TalpiprestoDatePicker.BackgroundColor = [0.96 0.96 0.96];
app.TalpiprestoDatePicker.Position = [198 291 145 22];
```

#### % Create TmedioDatePickerLabel

```
app.TmedioDatePickerLabel = uilabel(app.UIFigure);
app.TmedioDatePickerLabel.HorizontalAlignment = 'right';
app.TmedioDatePickerLabel.FontName = 'Calibri Light';
app.TmedioDatePickerLabel.FontSize = 14;
app.TmedioDatePickerLabel.Position = [106 258 71 26];
app.TmedioDatePickerLabel.Text = 'T medio';
```

#### % Create TmedioDatePicker

```
app.TmedioDatePicker = uideatepicker(app.UIFigure);
app.TmedioDatePicker.Editable = 'off';
app.TmedioDatePicker.FontName = 'Calibri Light';
app.TmedioDatePicker.FontSize = 14;
```

```
app.TmedioDatePicker.BackgroundColor = [0.96 0.96 0.96];  
app.TmedioDatePicker.Position = [198 262 145 22];
```

#### % Create TalpitardiDatePickerLabel

```
app.TalpitardiDatePickerLabel = uilabel(app.UIFigure);  
app.TalpitardiDatePickerLabel.HorizontalAlignment = 'right';  
app.TalpitardiDatePickerLabel.FontName = 'Calibri Light';  
app.TalpitardiDatePickerLabel.FontSize = 14;  
app.TalpitardiDatePickerLabel.Position = [63 230 105 26];  
app.TalpitardiDatePickerLabel.Text = 'T al più tardi';
```

#### % Create TalpitardiDatePicker

```
app.TalpitardiDatePicker = uideatepicker(app.UIFigure);  
app.TalpitardiDatePicker.Editable = 'off';  
app.TalpitardiDatePicker.FontName = 'Calibri Light';  
app.TalpitardiDatePicker.FontSize = 14;  
app.TalpitardiDatePicker.BackgroundColor = [0.96 0.96 0.96];  
app.TalpitardiDatePicker.Position = [198 234 145 22];
```

#### % Create OutputdelmodelloLabel

```
app.OutputdelmodelloLabel = uilabel(app.UIFigure);  
app.OutputdelmodelloLabel.HorizontalAlignment = 'center';  
app.OutputdelmodelloLabel.VerticalAlignment = 'top';  
app.OutputdelmodelloLabel.FontName = 'Calibri Light';  
app.OutputdelmodelloLabel.FontSize = 18;  
app.OutputdelmodelloLabel.FontWeight = 'bold';  
app.OutputdelmodelloLabel.Position = [108 312 196 26];  
app.OutputdelmodelloLabel.Text = 'Output del modello';
```

#### % Create InputdelmodelloLabel

```
app.InputdelmodelloLabel = uilabel(app.UIFigure);  
app.InputdelmodelloLabel.FontName = 'Calibri Light';  
app.InputdelmodelloLabel.FontSize = 18;  
app.InputdelmodelloLabel.FontWeight = 'bold';  
app.InputdelmodelloLabel.Position = [128 640 155 27];  
app.InputdelmodelloLabel.Text = 'Input del modello';
```

#### % Create npuntiproiettatiEditFieldLabel

```
app.npuntiproiettatiEditFieldLabel = uilabel(app.UIFigure);  
app.npuntiproiettatiEditFieldLabel.HorizontalAlignment = 'right';  
app.npuntiproiettatiEditFieldLabel.FontName = 'Calibri Light';  
app.npuntiproiettatiEditFieldLabel.FontSize = 14;  
app.npuntiproiettatiEditFieldLabel.Position = [227 593 105 22];  
app.npuntiproiettatiEditFieldLabel.Text = 'n° punti proiettati';
```

#### % Create npuntiproiettatiEditField

```
app.npuntiproiettatiEditField = uieditfield(app.UIFigure, 'numeric');  
app.npuntiproiettatiEditField.Limits = [0 Inf];
```

```
app.npuntiproiettatiEditField.FontName = 'Calibri Light';
app.npuntiproiettatiEditField.FontSize = 14;
app.npuntiproiettatiEditField.Position = [347 593 38 22];
app.npuntiproiettatiEditField.Value = 50;

% Create sogliaEditFieldLabel
app.sogliaEditFieldLabel = uilabel(app.UIFigure);
app.sogliaEditFieldLabel.HorizontalAlignment = 'right';
app.sogliaEditFieldLabel.FontName = 'Calibri Light';
app.sogliaEditFieldLabel.FontSize = 14;
app.sogliaEditFieldLabel.Position = [294 561 38 22];
app.sogliaEditFieldLabel.Text = 'soglia';

% Create sogliaEditField
app.sogliaEditField = uieditfield(app.UIFigure, 'numeric');
app.sogliaEditField.Limits = [0 Inf];
app.sogliaEditField.FontName = 'Calibri Light';
app.sogliaEditField.FontSize = 14;
app.sogliaEditField.Position = [347 561 38 22];
app.sogliaEditField.Value = 0.8;

% Create ndatiEditFieldLabel
app.ndatiEditFieldLabel = uilabel(app.UIFigure);
app.ndatiEditFieldLabel.HorizontalAlignment = 'right';
app.ndatiEditFieldLabel.FontName = 'Calibri Light';
app.ndatiEditFieldLabel.FontSize = 14;
app.ndatiEditFieldLabel.Position = [32 593 42 22];
app.ndatiEditFieldLabel.Text = 'n° dati';

% Create ndatiEditField
app.ndatiEditField = uieditfield(app.UIFigure, 'numeric');
app.ndatiEditField.Limits = [5 Inf];
app.ndatiEditField.FontName = 'Calibri Light';
app.ndatiEditField.FontSize = 14;
app.ndatiEditField.Position = [89 593 38 22];
app.ndatiEditField.Value = 5;

% Create nsigmaLabel
app.nsigmaLabel = uilabel(app.UIFigure);
app.nsigmaLabel.HorizontalAlignment = 'right';
app.nsigmaLabel.FontName = 'Calibri Light';
app.nsigmaLabel.FontSize = 14;
app.nsigmaLabel.Position = [279 530 53 22];
app.nsigmaLabel.Text = 'n° sigma';

% Create nsigmaEditField
app.nsigmaEditField = uieditfield(app.UIFigure, 'numeric');
app.nsigmaEditField.Limits = [0 Inf];
```

```
app.nsigmaEditField.FontName = 'Calibri Light';
app.nsigmaEditField.FontSize = 14;
app.nsigmaEditField.Position = [347 530 38 22];
app.nsigmaEditField.Value = 1.96;

% Create metodoSpinnerLabel
app.metodoSpinnerLabel = uilabel(app.UIFigure);
app.metodoSpinnerLabel.HorizontalAlignment = 'right';
app.metodoSpinnerLabel.FontName = 'Calibri Light';
app.metodoSpinnerLabel.FontSize = 14;
app.metodoSpinnerLabel.Position = [26 563 50 22];
app.metodoSpinnerLabel.Text = 'metodo';

% Create metodoSpinner
app.metodoSpinner = uispinner(app.UIFigure);
app.metodoSpinner.Limits = [1 2];
app.metodoSpinner.FontName = 'Calibri Light';
app.metodoSpinner.FontSize = 14;
app.metodoSpinner.Position = [90 563 38 22];
app.metodoSpinner.Value = 1;

% Create limiteEditFieldLabel
app.limiteEditFieldLabel = uilabel(app.UIFigure);
app.limiteEditFieldLabel.HorizontalAlignment = 'right';
app.limiteEditFieldLabel.FontName = 'Calibri Light';
app.limiteEditFieldLabel.FontSize = 14;
app.limiteEditFieldLabel.Position = [40 532 37 22];
app.limiteEditFieldLabel.Text = 'limite';

% Create limiteEditField
app.limiteEditField = uieditfield(app.UIFigure, 'numeric');
app.limiteEditField.Limits = [1 Inf];
app.limiteEditField.FontName = 'Calibri Light';
app.limiteEditField.FontSize = 14;
app.limiteEditField.Position = [90 532 38 22];
app.limiteEditField.Value = 1;

% Create EtichettaEditFieldLabel
app.EtichettaEditFieldLabel = uilabel(app.UIFigure);
app.EtichettaEditFieldLabel.HorizontalAlignment = 'right';
app.EtichettaEditFieldLabel.FontName = 'Calibri Light';
app.EtichettaEditFieldLabel.FontSize = 14;
app.EtichettaEditFieldLabel.Position = [485 291 55 22];
app.EtichettaEditFieldLabel.Text = 'Etichetta';

% Create EtichettaEditField
app.EtichettaEditField = uieditfield(app.UIFigure, 'numeric');
app.EtichettaEditField.FontName = 'Calibri Light';
```

```
app.EtichettaEditField.FontSize = 14;
app.EtichettaEditField.Position = [555 291 100 22];
```

```
% Create DegradoprestazioniEditFieldLabel
```

```
app.DegradoprestazioniEditFieldLabel = uilabel(app.UIFigure);
app.DegradoprestazioniEditFieldLabel.HorizontalAlignment = 'right';
app.DegradoprestazioniEditFieldLabel.FontName = 'Calibri Light';
app.DegradoprestazioniEditFieldLabel.FontSize = 14;
app.DegradoprestazioniEditFieldLabel.Position = [422 254 118 22];
app.DegradoprestazioniEditFieldLabel.Text = 'Degrado prestazioni';
```

```
% Create DegradoprestazioniEditField
```

```
app.DegradoprestazioniEditField = uieditfield(app.UIFigure, 'numeric');
app.DegradoprestazioniEditField.FontName = 'Calibri Light';
app.DegradoprestazioniEditField.FontSize = 14;
app.DegradoprestazioniEditField.Position = [555 254 100 22];
```

```
% Create savedirEditFieldLabel
```

```
app.savedirEditFieldLabel = uilabel(app.UIFigure);
app.savedirEditFieldLabel.HorizontalAlignment = 'right';
app.savedirEditFieldLabel.Enable = 'off';
app.savedirEditFieldLabel.Visible = 'off';
app.savedirEditFieldLabel.Position = [453 619 44 22];
app.savedirEditFieldLabel.Text = 'savedir';
```

```
% Create savedirEditField
```

```
app.savedirEditField = uieditfield(app.UIFigure, 'text');
app.savedirEditField.Enable = 'off';
app.savedirEditField.Visible = 'off';
app.savedirEditField.Position = [512 619 100 22];
```

```
% Create homedirEditFieldLabel
```

```
app.homedirEditFieldLabel = uilabel(app.UIFigure);
app.homedirEditFieldLabel.HorizontalAlignment = 'right';
app.homedirEditFieldLabel.Enable = 'off';
app.homedirEditFieldLabel.Visible = 'off';
app.homedirEditFieldLabel.Position = [448 583 49 22];
app.homedirEditFieldLabel.Text = 'homedir';
```

```
    % Create homedirEditField
```

```
    app.homedirEditField = uieditfield(app.UIFigure, 'text');
    app.homedirEditField.Enable = 'off';
    app.homedirEditField.Visible = 'off';
    app.homedirEditField.Position = [512 583 100 22];
```

```
    % Create SaveDb
```

```
    app.SaveDb = uieditfield(app.UIFigure, 'text');
```



```
app.SaveDb.Position = [44 406 165 22];

% Create SaveSeries
app.SaveSeries = uicontrol(app.UIFigure, 'text');
app.SaveSeries.Position = [208 406 163 22];

% Create DatabaseLabel
app.DatabaseLabel = uicontrol(app.UIFigure);
app.DatabaseLabel.FontName = 'Calibri Light';
app.DatabaseLabel.FontSize = 14;
app.DatabaseLabel.Position = [98 427 58 22];
app.DatabaseLabel.Text = 'Database';

% Create SeriesLabel
app.SeriesLabel = uicontrol(app.UIFigure);
app.SeriesLabel.FontName = 'Calibri Light';
app.SeriesLabel.FontSize = 14;
app.SeriesLabel.Position = [279 427 39 22];
app.SeriesLabel.Text = 'Series';

% Create OutputdellaprevisionesuInfluxLabel
app.OutputdellaprevisionesuInfluxLabel = uicontrol(app.UIFigure);
app.OutputdellaprevisionesuInfluxLabel.FontName = 'Calibri Light';
app.OutputdellaprevisionesuInfluxLabel.FontSize = 18;
app.OutputdellaprevisionesuInfluxLabel.FontWeight = 'bold';
app.OutputdellaprevisionesuInfluxLabel.Position = [89 450 234 27];
app.OutputdellaprevisionesuInfluxLabel.Text = 'Output della
previsione su Influx';

% Show the figure after all components are created
app.UIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

% Construct app
function app = APP_SmartGEOInfluxdb_2_update

% Create UIFigure and components
createComponents(app)

% Register the app with App Designer
registerApp(app, app.UIFigure)
```

```
        if nargin == 0
            clear app
        end
    end

    % Code that executes before app deletion
    function delete(app)

        % Delete UIFigure when app is deleted
        delete(app.UIFigure)
    end
end
end
```