

DELIVERABLE 4.1_CIRI_MAM

Rapporto con gli studi sviluppati circa estrazione features, apprendimento supervisionato e non, algoritmi on-line per la diagnostica, algoritmi per la stima della RUL

Code	O4.1
Date	30/04/2020
Type	Confidential
Participants	CIRI-MAM
Authors	Alberto Regattieri (CIRI-MAM), Francesca Calabrese (CIRI-MAM)
Corresponding Authors	Alberto Regattieri

Abstract

L'apprendimento automatico, o Machine Learning (ML), viene definito come lo studio di algoritmi che imparano in modo autonomo tramite l'esperienza. In campo industriale, il ML ha diverse applicazioni, tra le quali il controllo dei processi di produzione e la manutenzione predittiva degli impianti. Seppur con obiettivi diversi, entrambe le applicazioni si basano sul monitoraggio di grandezze raccolte da componenti o sistemi all'interno di un impianto industriale per mezzo di opportuni sensori. Pertanto entrambe richiedono un'accurata analisi di grandi moli di dati al fine di estrarre informazioni utili alla comprensione dello stato del processo/impianto. Dal punto di vista della manutenzione, la realizzazione di una politica di tipo predittivo può avvenire attraverso un approccio chiamato in letteratura Prognostic Health Management (PHM), che si pone come obiettivo quello di stimare la vita utile residua dei sistemi o componenti monitorati, in modo da minimizzare il verificarsi di guasti inaspettati e di conseguenza massimizzare la disponibilità degli impianti. Questo approccio prevede 4 attività: la raccolta dati, l'estrazione delle features, la diagnostica e la prognostica.

A partire dagli studi relativi all'approccio PHM, in questo documento, verranno approfonditi i principali metodi esistenti in letteratura per l'estrazione delle features dai segnali grezzi, in quanto attività fondamentale sia per la diagnostica che per la rilevazione delle anomalie di processo.

Indice

Il contesto	3
L'estrazione delle features.....	5
Il processamento dei segnali.....	6
L'analisi nel dominio del tempo.....	6
L'analisi nel dominio della frequenza.....	7
L'analisi nel dominio tempo-frequenza.....	8
La riduzione della dimensionalità.....	9
L'estrazione delle features nelle applicazioni industriali	11
Algoritmi di ML per l'ottimizzazione dei processi.....	12
Riferimenti	13

Il contesto

La manutenzione dei sistemi di produzione ha un ruolo cruciale nella minimizzazione del costo complessivo di realizzazione dei prodotti. Contrariamente al passato, la funzione manutenzione non ha il solo compito di riparazione degli asset; piuttosto, gli obiettivi fondamentali che essa si pone sono: il raggiungimento della massima disponibilità degli impianti; l'ottimizzazione delle condizioni di funzionamento; la massimizzazione dell'utilizzo delle risorse e della vita utile dei macchinari; la minimizzazione delle scorte di ricambi; l'abilità nel reagire velocemente (Mobley, 2002). Per i componenti più critici, le strategie preventive, come la manutenzione su condizione (CBM), possono essere rimpiazzate da strategie predittive. La manutenzione predittiva (PM) può essere vista come un'evoluzione della manutenzione su condizione. Come mostrato in Figura 1, entrambe le politiche sono basate sul monitoraggio di grandezze raccolte dai macchinari per mezzo di opportuni sensori. Secondo la strategia su condizione, quando tali grandezze raggiungono una soglia di guasto prestabilita, viene innescato un allarme. Invece, secondo la strategia predittiva, viene prima identificato il momento in cui sta iniziando una degradazione. Poi, viene calcolata la vita utile residua (RUL), come la differenza tra il momento in cui è previsto che la grandezza raggiunga la soglia di guasto e il momento attuale. Questo consente dunque di predire il verificarsi di guasti, così da avere tempo sufficiente per la pianificazione sia degli interventi manutentivi che dell'approvvigionamento delle parti di ricambio. Inoltre, tale strategia consente di integrare dati storici raccolti dal componente in differenti condizioni di salute, con i dati raccolti in tempo reale durante il funzionamento della macchina, i quali forniscono informazioni sul suo stato reale. Pertanto, la RUL viene calcolata sulla base di un modello di degradazione costruito durante l'analisi dei dati, ma anche sulla base del livello reale di degradazione, così da consentire lo sfruttamento di tutta la vita utile residua.

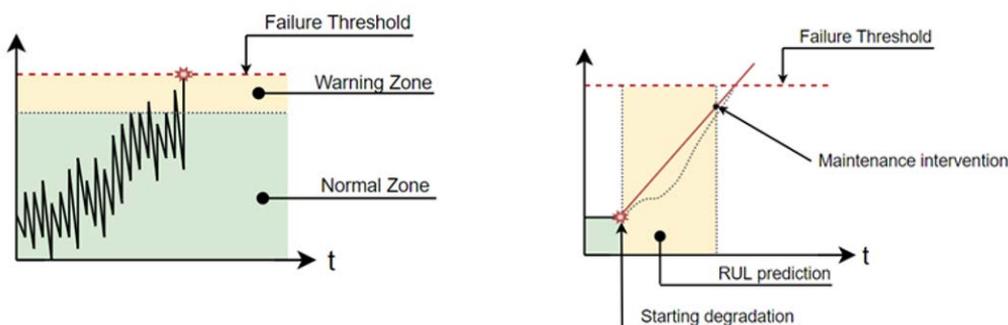


Figura 1 Manutenzione su condizione - Diagnostica (sinistra). Manutenzione predittiva - Prognostica (destra)

In letteratura, la manutenzione predittiva viene anche identificata con l'acronimo PHM – Prognostic Health Management. Sulla base di questo paradigma, per la predizione della RUL sono necessarie 4 attività principali:

1. Acquisizione dati. Le grandezze, come vibrazioni, emissioni acustiche o correnti, vengono raccolte dal sistema in differenti condizioni di salute.
2. Estrazione delle features. I segnali grezzi vengono processati in modo da estrarre caratteristiche rilevanti, chiamate *features*, che caratterizzano il comportamento del sistema in ogni condizione e consentono di distinguere le condizioni di salute da quelle di guasto. Inoltre, viene calcolato anche un indicatore di salute (HI), che rispecchia l'andamento della degradazione.
3. Diagnostica. L'identificazione delle relazioni tra le features estratte e le condizioni di salute è lo scopo della diagnostica. In questa fase, le regole per la separazione/classificazione delle diverse condizioni note vengono identificate per mezzo di modelli di Machine Learning (ML), in modo da associare le *features* alla condizione corrispondente (Jardine et al., 2006).

4. Prognostica. La modellazione dell'andamento dell'HI è prerogativa della prognostica. In questa fase, vengono identificate le relazioni tra l'HI e il corrispondente livello di degradazione, che consentono di predire in qualsiasi momento la vita utile residua (Yaguo Lei et al., 2018).

Il framework generale del PHM è mostrato in Figura 2. Si possono distinguere due approcci principali, che si differenziano sulla base del tipo di apprendimento che viene utilizzato, supervisionato o non supervisionato.

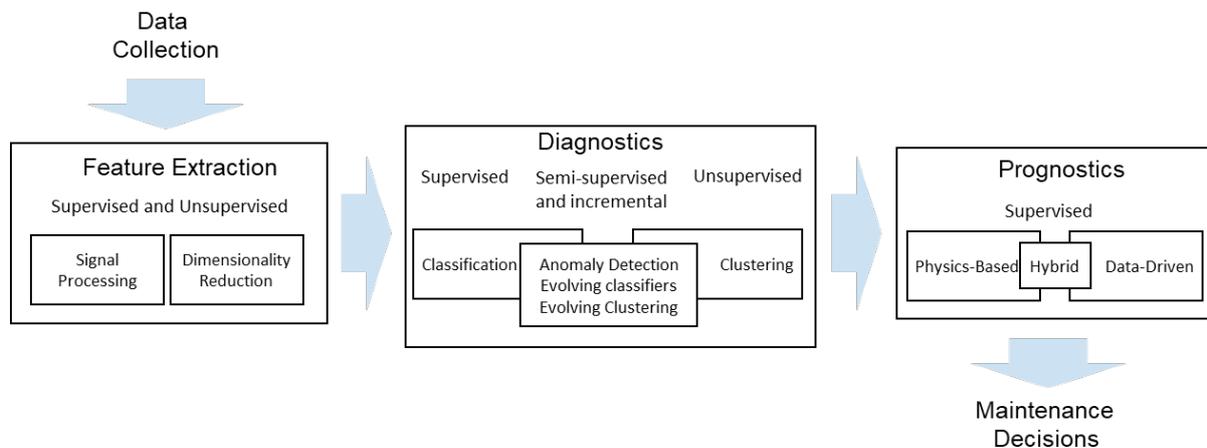


Figure 2. Framework del Prognostic Health Management

L'apprendimento supervisionato è uno degli approcci più comuni adottato in letteratura. Secondo questo tipo di apprendimento, un dataset deve contenere sia i predittori che le etichette, ovvero è necessario che ogni *feature* estratta (predittore) sia associata alla corrispondente condizione di salute o livello di degradazione (etichetta). In tal modo, i modelli di ML possono essere allenati su un sottoinsieme del dataset al fine di estrarre le regole di classificazione da applicare a nuove misurazioni, per le quali l'etichetta non è disponibile.

L'apprendimento non supervisionato viene invece utilizzato quando i segnali raccolti non sono associati ad alcuna condizione di salute. Questa situazione si verifica piuttosto spesso nelle applicazioni reali, in quanto non è possibile svolgere test in laboratorio o non è possibile portare al guasto alcuni componenti/sistemi per ragioni di sicurezza. Quindi, i segnali vengono raccolti direttamente dal campo, senza alcuna informazione sulle condizioni al contorno. In questi casi, gli algoritmi di classificazione vengono sostituiti da algoritmi di clustering, i quali separano i dati in gruppi, in modo da minimizzare (massimizzare) la distanza (similarità) intra-cluster e massimizzare (minimizzare) la distanza (similarità) tra cluster diversi.

La scelta dell'una o dell'altra tipologia di apprendimento, sostanzialmente dipende dalla tipologia di dati disponibili.

In figura 2, viene proposto anche un terzo approccio. Si tratta di un approccio misto, chiamato semi-supervisionato e incrementale. Sia l'apprendimento supervisionato che quello non supervisionato si basano infatti su dati storici per generare modelli statici, che devono essere allenati di nuovo se si presenta una nuova condizione. Dato che i contesti industriali sono per natura dinamici, questi due approcci falliscono nel riconoscere un comportamento della macchina che non era noto al momento del training. Quindi, l'apprendimento non supervisionato può essere applicato ai dati in streaming, così da riconoscere se qualcosa sta cambiando e se la condizione corrente è nota o meno. Nel primo caso, l'apprendimento supervisionato viene usato per classificare l'osservazione in una delle condizioni note. Nel secondo caso, gli algoritmi di cluster possono essere utilizzati per raggruppare dati "simili" che potenzialmente appartengono alla stessa condizione non nota. Questi due approcci possono poi essere integrati aggiungendo il paradigma di apprendimento incrementale, secondo cui i cluster identificati come corrispondenti a condizioni non note possono essere automaticamente incorporati all'interno dei modelli già allenati di classificazione.

L'analisi della letteratura sul ML ha fornito anche il focus sull'applicazione di tali modelli ai processi industriali e, in particolare, ai processi meccanici. Per tale applicazione, il focus è concentrato maggiormente su due classici problemi di apprendimento supervisionato: la classificazione e la regressione. È stato quindi condotto uno studio approfondito di vari casi di studio e tecniche di risoluzione che utilizzano funzioni di costo, discesa del gradiente, vettorizzazione, regressione lineare e polinomiale (da una variabile a più variabili), regressione logistica per problemi di classificazione a più variabili, overfitting, regolarizzazione, Reti neurali artificiali (ANN) and algoritmi a vettori di supporto (SVM). Per supportare la gestione dei problemi di apprendimento supervisionato, sono stati utilizzati i software Python™ and GNU OCTAVE, per i quali è stata prevista un'attività di apprendimento attraverso tutorial e corsi specifici on line. L'utilizzo di questi due software hanno inoltre reso possibile affrontare lo studio di tecniche di importazione, manipolazione e pulizia dei dati, come attività di pre-processamento.

In entrambe le applicazioni, manutenzione predittiva e ottimizzazione di processi, l'estrazione delle *features* dai segnali grezzi rappresenta un'attività fondamentale, in quanto da essa dipendono i risultati sia della diagnostica che dell'ottimizzazione dei parametri di processo (Escobar & Morales-Menendez, 2018). L'estrazione delle *features* può essere condotta in modo diverso a seconda che si utilizzi un approccio supervisionato o non supervisionato. Nelle sezioni successive, viene eseguita una breve revisione della letteratura sulle tecniche di estrazione delle *features* più comunemente usate nelle due applicazioni di interesse.

L'estrazione delle features

L'estrazione delle features ha l'obiettivo di mappare i segnali acquisiti in uno spazio m -dimensionale in uno spazio n -dimensionale, con $n < m$, in modo da fornire una rappresentazione sintetica dei segnali grezzi, preservandone la struttura originaria e il contenuto informativo. In questo processo, vengono dunque estratte caratteristiche rilevanti dei segnali che consentono di caratterizzare i diversi comportamenti della macchina in esame.

Relativamente alla scelta delle features, ci sono due problemi rilevanti. Il primo è connesso alla frequenza di acquisizione dei segnali, mentre il secondo al numero di segnali acquisiti. I segnali acquisiti ad elevate frequenze, nell'ordine ad esempio dei kilohertz, presentano andamenti oscillatori che possono compromettere notevolmente i risultati della classificazione. Per questa ragione, viene realizzata un'attività definita di campionamento (Lee et al., 2014), o di processamento dei segnali, in cui alcune statistiche nei domini del tempo, frequenza o tempo-frequenza vengono calcolate su un segmento di segnale di una certa lunghezza, in modo da sintetizzarlo con una o più *features*. Questa attività, tuttavia, può portare ad un aumento del numero di variabili, che possono o meno essere rilevanti ai fini della classificazione. Inoltre, spesso le machine sono provviste di un numero elevato di sensori e quindi il numero di segnali (predittori) è già elevato. Per ogni segnale in più, il numero delle *features* estratte come appena descritto raddoppia, generando un data set ad elevate dimensioni.

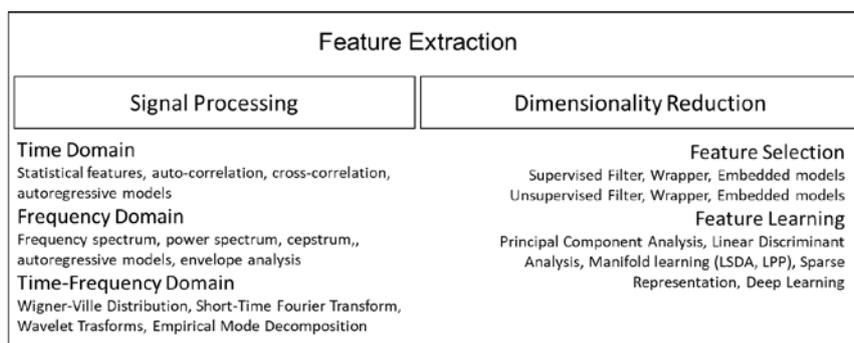


Figura 3 Tecniche di estrazione delle features

Pertanto, viene realizzata una seconda attività, chiamata riduzione della dimensionalità, al fine di ridurre lo spazio dimensionale del dataset, selezionando o estraendo le features rilevanti e non ridondanti (R. Hu et al., 2017), (Wang et al., 2017). Secondo l'approccio supervisionato, le features migliori possono essere selezionate o sulla base di criteri definiti dagli esperti, i quali hanno una profonda conoscenza del componente in analisi, oppure attraverso tecniche di selezione automatica. Secondo l'approccio non supervisionato, invece, le features migliori vengono estratte attraverso tecniche automatiche sulla base della sola struttura dei dati.

Il processamento dei segnali

Le tecniche di processamento dei segnali vengono suddivise in tre categorie: tecniche nel dominio del tempo, tecniche nel dominio della frequenza, tecniche nel dominio del tempo-frequenza. Le features così estratte hanno un preciso significato fisico.

L'analisi nel dominio del tempo

Il processamento dei segnali nel dominio del tempo consiste principalmente nell'estrazione di statistiche del segnale. L'analisi statistica mira a descrivere la natura dei segnali assumendo che loro presentano un certo grado di regolarità. L'idea è quindi che, se si verifica un cambio nella condizione di salute, la funzione di densità di probabilità (PDF) dei segnali, e quindi i parametri statistici, cambiano di conseguenza (Mehrjou et al., 2017). Le features statistiche più utilizzate sono riassunte in Tabella 1. La media, la varianza, la *skewness* e il *kurtosis* sono, rispettivamente, il primo, secondo, terzo e quarto momento statistico della PDF del segnale. In particolare, la *skewness* misura il grado di asimmetria della PDF intorno alla media, mentre il *kurtosis* fornisce informazioni sui picchi e le code della PDF (Gomes Teixeira de Almeida et al., 2002). Il fattore di cresta, la *clearance* e il fattore di impulso misurano gli impulsi che esistono nel segnale. In particolare, il fattore di cresta è una misura del numero e della nitidezza dei picchi in un segnale. Sia il *kurtosis* che il fattore di cresta sono robusti al variare della condizione operativa e pertanto sono buoni indicatori per la rilevazione di guasti allo stato iniziale. Tuttavia, il *kurtosis* decresce all'aumentare della severità del guasto e pertanto non è un buon indicatore per guasti agli stadi avanzati. Il *Root Mean Square* (RMS) è una misura dell'intensità energetica del segnale, mentre il *Root amplitude* e il valore di picco massimo riflettono l'ampiezza del segnale. Sebbene questi ultimi sono capaci di differenziare diverse severità di guasto, non sono idonei per la rilevazione di guasti allo stato iniziale, il che vale anche per l'RMS.

Tabella 1 Feature statistiche nel dominio del tempo

Picco massimo	$X_{peak} = \max(x(n))$	Root Amplitude	$X_{root} = \left(\frac{\sum_{n=1}^N \sqrt{ x(n) }}{N} \right)^2$
Varianza	$X_{var} = \sqrt{\frac{\sum_{n=1}^N (x(n) - x_m)^2}{N - 1}}$	Fattore di cresta	$X_{crest} = \frac{X_{peak}}{X_{rms}}$
Root Mean Square	$X_{rms} = \sqrt{\frac{\sum_{n=1}^N (x(n))^2}{N}}$	Clearance	$X_{clearance} = \frac{X_{peak}}{X_{root}}$
Skewness	$X_{skewness} = \frac{\sum_{n=1}^N (x(n) - x_m)^3}{(N - 1)X_{var}^3}$	Fattore di forma	$X_{shape} = \frac{X_{rms}}{\frac{1}{N} \sum_{n=1}^N x(n) }$

Kurtosis	$X_{kurt} = \frac{\sum_{n=1}^N (x(n) - x_m)^4}{(N-1)X_{var}^4}$	Fattore di impulso	$X_{impulse} = \frac{X_{peak}}{\frac{1}{N} \sum_{n=1}^N x(n) }$
----------	---	--------------------	--

dove $x_m = \frac{\sum_{i=1}^N x(n)}{N}$ è la media del segnale.

Altre tecniche di analisi del segnale nel dominio del tempo sono l'analisi di autocorrelazione, l'analisi di correlazione incrociata e i modelli di autoregressione. L'analisi di autocorrelazione è capace di captare le componenti periodiche da un segnale caratterizzato da un notevole rumore. La correlazione incrociata è utile per la stima del livello di similarità tra due segnali. Infine, tramite modelli di autoregressione è possibile trovare i coefficienti che minimizzano una certa funzione, i quali possono essere utilizzati direttamente come feature per la successiva classificazione.

L'analisi nel dominio della frequenza

Molti component critici mostrano frequenze caratteristiche per alcune tipologie di guasto. In questi casi, l'analisi nel dominio della frequenza viene preferita, in quanto permette di identificare e isolare facilmente le componenti di interesse. L'analisi in frequenza è spesso basata sullo spettro di frequenza, sullo spettro di potenza, o sul cepstrum, ovvero il logaritmo dello spettro del segnale. Questi ultimi due sono spesso adottati per analisi più complesse (Liang et al., 2013), in quanto più sensibili ai guasti nello stadio iniziale.

Il fondamento teorico dello spettro di frequenza risiede nella trasformata di Fourier, secondo la quale il segnale viene diviso in una somma di component sinusoidali utilizzando la serie di Fourier. Lo spettro di frequenza del segnale è quindi ottenuto proiettando tali componenti sull'asse della frequenza. La trasformata di Fourier di un segnale periodico $x(t)$, con periodo T e frequenza f può essere calcolato come segue:

$$X(f) = \int_{-\infty}^{+\infty} x(t) e^{-j2\pi f t} dt \quad (1)$$

Dato che la trasformata di Fourier può essere applicata solo a segnali continui, viene spesso utilizzata la trasformata di Fourier discreta (DFT) al fine ottenere lo spettro di frequenza di un segnale discreto, come segue:

$$X(f) = \sum_{n=-\infty}^{\infty} x(t_n) e^{-j2\pi f t_n} dt \quad (2)$$

dove f_s è la frequenza di campionamento, $t_n = n\Delta t = n/f_s$, e Δt è l'intervallo tra due misurazioni consecutive. Dal momento che la DFT richiede tempo per essere calcolata nel caso di segnali molto lunghi, vengono spesso utilizzati degli algoritmi per il calcolo della trasformata di Fourier veloce (FFT).

Sulla base dello spettro di frequenza, le *features* possono essere estratte calcolando alcune statistiche, le più importanti delle quali sono riassunte in Tabella 2.

Table 2 Statistical Features in the frequency-domain

$p_1 = \frac{\sum_{k=1}^K (s(k) - p_1)^2}{K-1}$	$p_6 = \sqrt{\frac{\sum_{k=1}^K f_k^2 s(k)}{\sum_{k=1}^K s(k)}}$	$p_{11} = \frac{\sum_{k=1}^K (f_k - p_5)^3 s(k)}{K p_6^3}$
$p_2 = \frac{\sum_{k=1}^K (s(k) - p_1)^3}{K(\sqrt{p_2})^2}$	$p_7 = \sqrt{\frac{\sum_{k=1}^K (f_k - p_5)^2 s(k)}{\sum_{k=1}^K s(k)}}$	$p_{12} = \frac{\sum_{k=1}^K (f_k - p_5)^4 s(k)}{K p_6^4}$

$p_3 = \frac{\sum_{k=1}^K (s(k) - p_1)^4}{K p_2^2}$	$p_8 = \sqrt{\frac{\sum_{k=1}^K f_k^4 s(k)}{\sum_{k=1}^K f_k^2 s(k)}}$	$p_{13} = \frac{\sum_{k=1}^K (f_k - p_5)^{1/2} s(k)}{K \sqrt{p_6}}$
$p_4 = \frac{\sum_{k=1}^K f_k s(k)}{\sum_{k=1}^K s(k)}$	$p_9 = \frac{\sum_{k=1}^K f_k^2 s(k)}{\sqrt{\sum_{k=1}^K s(k) \sum_{k=1}^K f_k^4 s(k)}}$	$p_{14} = \frac{\sum_{k=1}^K s(k)}{K}$
$p_5 = \sqrt{\frac{\sum_{k=1}^K (f_k - p_5)^2 s(k)}{K}}$	$p_{10} = \frac{p_6}{p_5}$	

dove $s(k)$ è lo spettro per $k = 1, 2, \dots, K$, K è il numero delle linee dello spettro e f_k è il valore di frequenza in corrispondenza della k -esima linea dello spettro. La feature p_{14} è la frequenza media: essa indica l'energia del segnale nel dominio della frequenza e rappresenta la media dell'ampiezza di tutte le frequenze. Più è elevato il grado di guasto, tanto più è grande la frequenza media. Le features $p_1, p_2, p_3, p_5, p_{10}, p_{11}, p_{12}$ descrivono la convergenza dello spettro di potenza, riflettendo l'energia dello spettro di frequenza. Le features p_4, p_6, p_7, p_8 mostrano il cambiamento della posizione delle frequenza principali che sono dominanti nello spettro di frequenza.

Anche l'analisi in frequenza può essere condotta utilizzando modelli di autoregressione, quali lo spettro di autoregressione o lo spettro di autoregressione a media mobile.

Infine, in tale dominio può essere realizzata anche l'analisi dell'involuppo, la quale si basa sulla trasformata di Hilbert e rappresenta un'affidabile tecnica di demodulazione del segnale per rilevare e isolare guasti nei cuscinetti, ingranaggi, motori e altri componenti (Geropp, 1997).

L'analisi nel dominio tempo-frequenza

Sebbene sia l'analisi nel dominio del tempo che quella nel dominio della frequenza forniscano ottimi risultati, in molti casi i segnali non sono stazionari e la trasformata di Fourier non riflette il cambiamento della frequenza nel tempo (Y Lei, 2016). Le componenti non stazionarie dei segnali contengono informazioni rilevanti rispetto ai guasti e l'analisi nel dominio del tempo-frequenza in questi casi non solo è obbligatoria, ma fornisce anche una rappresentazione del segnale più facilmente interpretabile. Le tecniche principali in questo dominio sono la distribuzione di Wigner-Ville Distribution (WVD), la trasformata di Fourier a breve termine (STFT) e le trasformate Wavelet (WT). La WVD e la STFT sono distribuzioni in tempo-frequenza che rappresentano l'energia o la potenza dei segnali in una funzione bidimensionale del tempo e della frequenza.

Secondo la STFT, il segnale $x(t)$ viene suddiviso in segmenti a durata temporale breve, in cui può essere considerato stazionario e successivamente ad ogni segmento viene applicata la trasformata di Fourier.

La trasformata Wavelet può essere utilizzata come un'analisi multi-scalata del segnale, in cui la risoluzione nel dominio tempo-frequenza dipende dalla frequenza del segnale (Peng & Chu, 2004). Dunque tale trasformata esprime il segnale in una serie di funzioni ondulate (figlie) con differenti frequenze in istanti temporali diversi attraverso la dilatazione e traslazione dalla forma d'onda (wavelet) madre. La WT è particolarmente efficace quando un segnale è composto da componenti ad alta frequenza di breve durata e da componenti a bassa frequenza di lunga durata. Generalmente, tutti i segnali possono essere scomposti in una somma di wavelet con differenti coefficienti, che possono essere ottimizzati attraverso opportuni algoritmi. Grazie alle proprietà di buona concentrazione di energia, la WT può rappresentare il segnale con un numero limitato di coefficienti, che misurano la similarità tra il segnale e ognuna delle sue wavelet figlie. Questi coefficienti possono essere direttamente utilizzati come features. La trasformata wavelet continua (CWT) viene così definita:

$$W(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} x(t) \psi^* \left(\frac{t-b}{a} \right) dt \quad (3)$$

dove $x(t)$ è il segnale, a è il parametro di scala, b è il parametro tempo, ψ è una funzione wavelet, che è una funzione oscillatoria centrata nello zero e a energia finita, e ψ^* è il suo complesso coniugato. La serie di wavelet con differenti frequenze a istanti di tempo diversi può essere ottenuta attraverso la dilatazione del parametro di scala e la traslazione del parametro tempo. Al fine di interpretare il segnale, vengono spesso utilizzati lo scalogramma, definito come $|W(a, b)|^2$, e lo spettro di fase della wavelet. La CWT può essere realizzata con differenti funzioni wavelet di base, che portano a risultati diversi. Una delle più utilizzate è la Wavelet di Haar, che ha buone caratteristiche di filtro passa-basso (Li et al., 2007b). La funzione di Haar può essere espressa come segue:

$$\psi(t) = \begin{cases} 1 & 0 \leq t \leq \frac{1}{2} \\ -1 & \frac{1}{2} \leq t \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Talvolta viene utilizzata la funzione di Morlet, la quale ha caratteristiche di passa-basso ed è un buon rilevatore di alte frequenze. La funzione di Morlet è data da:

$$\psi(t) = e^{-\beta(t^2/2)} e^{j\omega_0 t} \quad (5)$$

Dove i parametri $\beta = \omega_0^2$ e ω_0 vengono definiti sulla base della particolare applicazione. .

Recentemente, è stata sviluppata una nuova tecnica di analisi nel tempo-frequenza, chiamata Empirical Mode Decomposition (EMD). Secondo questa tecnica, il segnale viene scomposto in un insieme di component quasi ortogonali, chiamati Intrinsic Mode Function (IMF). Da questi, è possibile estrarre l'energia dei segnali a differenti bande di frequenza, che cambiano al verificarsi di un guasto o di un'anomalia (Bin et al., 2012). La decomposizione parte con la ricerca dei minimi e dei massimi lungo il segnale $x(t)$. Successivamente, i minimi e i massimi vengono interpolati attraverso due curve, $s_{max}(t)$ e $s_{min}(t)$, che rappresentano i confini del segnale. Infine, dal segnale originale viene estratta la funzione media $m(t)$, ottenendo il nuovo segnale $x_1(t) = x(t) - m(t)$. Questo rappresenta un IMF se soddisfa le seguenti condizioni (Rai & Upadhyay, 2017): (1) deve sussistere un'uguaglianza o una differenza di al massimo uno tra il numero di estremi e il numero di attraversamenti dello 0; (2) l'involuppo definito dai massimi locali e quello definito dai minimi locali devono avere una media uguale a 0. Quindi la procedura viene ripetuta fin quando tali condizioni non vengono soddisfatte, momento in cui si ottiene il primo IMF, $C_1(t)$. Questo poi viene sottratto al segnale originario per ottenere il segnale residuo $r_1(t) = x(t) - C_1(t)$. Questo segnale rappresenta l'input per il calcolo del secondo IMF. Questo processo continua fin quando il segnale residuo è costante o monotono (Ricci & Pennacchi, 2011). Dagli IMF così ottenuti è possibile calcolare una *feature* che rappresenta l'energia del segnale come segue

$$E_j = \sum_{k=1}^n C_i |k|^2 \quad (8)$$

dove E_j è l'energia della j -esima banda del IMF C_i e k rappresenta il corrispondente istante temporale.

La riduzione della dimensionalità

Insieme di dati ad elevate dimensioni spesso contengono *features* ridondanti. Infatti, la dimensionalità intrinseca di questi dataset è generalmente piccola. Pertanto, possono essere utilizzate tecniche di riduzione della dimensionalità per trasformare dataset m -dimensionali in dataset n -dimensionali, con $n \ll m$. Le tecniche di riduzione della dimensionalità possono essere classificate in tecniche di apprendimento e tecniche di selezione, a seconda che cambi o meno lo spazio originario (Tang et al., 2014). L'apprendimento delle *features* corrisponde al processo di proiezione delle *features* in uno spazio con dimensionalità inferiore, mentre i metodi di selezione

delle *features* selezionato un sottoinsieme di *features* rilevanti e non ridondanti dallo spazio originario, senza alcuna trasformazione.

L'idea alla base dell'apprendimento è quella di imparare automaticamente le *features* da una grande mole di dati. Queste *features* non hanno un significato fisico come quelle estratte nella fase di processamento dei segnali, ma sono capaci di catturare informazioni nascoste nei dati grezzi, di amplificare le informazioni rilevanti e di eliminare variazioni irrilevanti (Yaguo Lei et al., 2016). Metodi tradizionali di apprendimento delle *features* sono la *Principal Component Analysis* (PCA) (F. Wang et al., 2015) e la *Linear Discriminant Analysis* (C. Zhang et al., 2018), le quali però hanno limitazioni nelle trasformazioni non lineari. Recentemente, stanno emergendo metodi di apprendimento *manifold*, come la *Locality Sensitive Discriminant Analysis* (LSDA) e *Locality Preserving Projection* (LPP) (X. Zhang et al., 2017), che identificano la struttura di spazi con bassa dimensionalità (*manifold*) incorporata nello spazio ad alta dimensionalità, intorno alla quale è più probabile che si concentrino i dati (Gan et al., 2015). L'apprendimento delle *features* può essere visto anche come uno strumento di rappresentazione, che impara quindi rappresentazioni e trasformazioni dei dati, rendendo più semplice l'estrazione di informazioni rilevanti (Bengio et al., 2013). In quanto tale, la teoria della *sparse representation* e il *deep learning* stanno ricevendo sempre maggiore attenzione. La *sparse representation* è un metodo di ricostruzione del segnale che rappresenta il segnale di input come una combinazione lineare sparsa di atomi scelti da una collezione chiamata *dictionary*, in modo che soltanto una piccola porzione di atomi contribuisca all'approssimazione del segnale (Han et al., 2017). Il *deep learning* rappresenta un *breakthrough* nell'apprendimento delle *features*. Le architetture *deep* realizzano una serie di trasformazioni non lineari al fine di imparare in modo automatico una gerarchia di *features*, che rappresenta il segnale in modo sempre più astratto. Pertanto queste architetture possono essere viste come un modo di decomporre il problema dell'estrazione delle *features* in sotto-problemi con differenti livelli di astrazione (Lillicrap et al., 2009).

I metodi di selezione delle *features*, invece, vengono applicati ad un insieme di *features* precedentemente estratte al fine di ridurre il numero e di conseguenza aumentare l'accuratezza e la capacità di generalizzazione dei modelli di classificazione, di ridurre l'overfitting, i tempi di computazione e la memoria richiesta.

Questi metodi possono essere divisi in tre categorie: *filter*, *wrapper* ed *embedded*. I primi valutano le *features* assegnandogli un punteggio sulla base delle caratteristiche generali dei dati e selezionano le *features* con punteggio più elevato. Il costo computazionale di tali metodi è relativamente basso e sono piuttosto robusti rispetto al problema dell'overfitting. Tuttavia, dato che non coinvolgono alcun algoritmo di classificazione, ignorano l'effetto delle *features* selezionate sulle performance della classificazione. I metodi *wrapper*, invece, utilizzano metodi di classificazione per valutare la qualità di una *feature*. Quindi prima estraggono un sottoinsieme di *features* e lo valutano sulla base dell'accuratezza di un classificatore; dopo ripetono il processo con differenti sottoinsiemi e selezionano quello che fornisce l'accuratezza migliore. Sebbene le *features* così selezionate abbiano buone performance, i metodi *wrapper* sono più complessi da un punto di vista computazionale e propensi all'overfitting. Infine, i metodi *embedded* realizzano la selezione delle *features* e la classificazione in un unico problema di ottimizzazione, considerando quindi la selezione delle *features* come parte del *training*. Pertanto questi metodi hanno i vantaggi di entrambi i metodi precedenti, in quanto sono meno complessi dal punto di vista computazionale, ma incorporano le performance della classificazione.

Di recente, sono state sviluppate anche versioni non supervisionate di questi metodi (Xie et al., 2018). Ad esempio, i metodi *filter* non supervisionati, come il *Laplacian Score*, assegnano ad ogni *feature* un punteggio che indica la sua capacità di preservare la struttura dei dati. La versione non supervisionata dei metodi *wrapper* utilizza algoritmi di clustering al posto di quelli di classificazione (D. Wang et al., n.d.). Infine, nei metodi *embedded*, la selezione delle *features* è spesso realizzata in combinazione con i modelli di apprendimento delle *features*, come la PCA (Y. Zhu et al., 2018), lo *sparse learning* (Hu et al., 2017) o il *manifold learning* (X. Zhu et al., 2017).

L'estrazione delle features nelle applicazioni industriali

In questa sezione, verrà descritto come, in pratica, le *features* rilevanti vengono estratte dai segnali grezzi, nei casi di apprendimento supervisionato, non supervisionato e incrementale.

Apprendimento supervisionato. I metodi di estrazione delle *features* supervisionati vengono utilizzati quando sono disponibili un gran numero di dati associati a ciascuna delle condizioni di salute o di guasto note. Quando si ha a che fare con segnali di vibrazione o di corrente, solitamente raccolti a frequenza elevate, viene consigliata l'estrazione delle *features* in uno dei tre domini per ogni condizione. Se il numero di *features* estratte è elevato, possono poi essere applicati i metodi di selezione delle *features*. In figura 1, viene mostrato un esempio di estrazione delle *features* nel paradigma dell'apprendimento supervisionato. I segnali grezzi che corrispondono alla condizione di salute (blu) e a una condizione di guasto (arancione) vengono prima divisi in piccoli segmenti di una certa lunghezza. Poi, per ogni segmento vengono estratte delle *features* in uno o più domini di analisi dei segnali, al fine di ridurre la frequenza di campionamento mantenendo il contenuto informativo. Infine, le *features* più rilevanti, ovvero quelle che distinguono meglio le due condizioni disponibili, vengono selezionate per ridurre il numero di variabili del dataset.

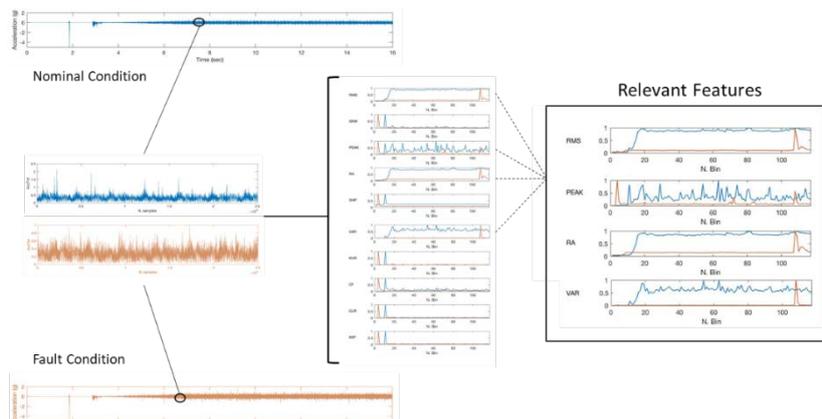


Figura 4 L'estrazione delle features con l'approccio supervisionato

Per altre tipologie di segnali, come ad esempio temperature e pressioni, che solitamente oscillano meno e vengono raccolti a più bassa frequenza, possono prima essere applicate tecniche di processing dei segnali semplici per il ridurre il campionamento. In alternativa, le tecniche di selezione delle *features* possono essere direttamente applicate ai segnali grezzi.

Apprendimento non supervisionato. Quando non sono disponibili dati etichettati, riferiti cioè a una specifica condizione, si rendono necessarie le tecniche di riduzione della dimensionalità non supervisionate. In tal caso, le *features* possono essere sia estratte da un insieme precedentemente creato attraverso il processing dei

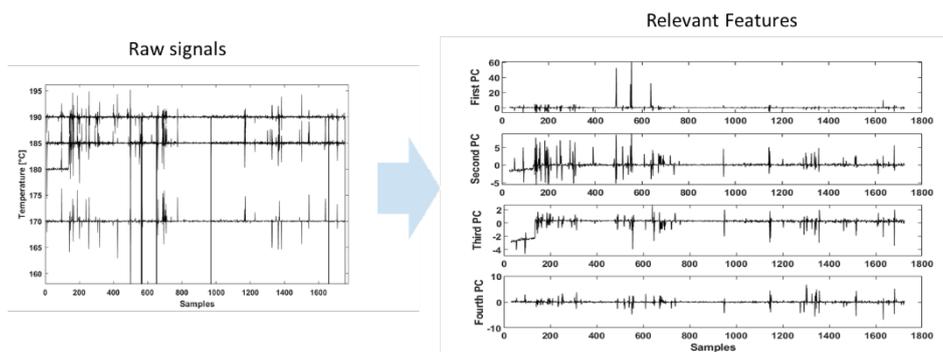


Figura 5 Estrazione delle features con l'approccio non supervisionato

segnali, oppure direttamente dai segnali grezzi. In figura 5, viene mostrato un esempio in cui le *features* (a destra) sono estratte automaticamente tramite la PCA. È evidente come siano profondamente diverse dal segnale originario, ma al tempo stesso sono capaci di mantenere una certa variabilità dei dati originari.

Apprendimento non supervisionato e incrementale. Nelle analisi *streaming*, i segnali non sono tutti disponibili all'inizio, ma si rendono disponibili man mano che la macchina in esame lavora. In questo caso, se un'analisi supervisionata e in *batch* era stata precedentemente effettuata, allora è possibile semplicemente estrarre le *features* identificate nella precedente analisi in *streaming*, non appena il segmento di segnale nella finestra temporale stabilita si rende disponibile. Tuttavia, se si sono utilizzate tecniche di riduzione della dimensionalità non supervisionate, allora non è possibile utilizzare la metodologia appena descritta. Inoltre, se l'obiettivo è quello di riconoscere nuove condizioni, allora anche l'apprendimento delle *features* dovrebbe essere realizzato in modo incrementale. In questo caso, tali tecniche possono essere modificate per essere implementate in *streaming*. Un esempio di PCA incrementale è fornito da Lippi & Ceccarelli (2019), dove i risultati della metodologia sono proposti e sono del tutto simili al caso in cui la stessa tecnica fosse implementata ai dati in *batch*.

Algoritmi di ML per l'ottimizzazione dei processi

L'ottimizzazione dei processi attraverso il Machine Learning (ML) può essere strutturato in due diversi problemi (Weichert et al., 2019). Nel caso di ottimizzazione con cambio dei parametri di processo, i parametri ottimi di processo vengono determinati utilizzando i dati di un processo già in atto. Aggiustando tali parametri sulla base delle caratteristiche del prodotto e dello specifico obiettivo di ottimizzazione, è possibile raggiungere una qualità migliore. L'obiettivo dell'ottimizzazione può essere specifico del prodotto o del processo. Quantità specifiche di prodotto sono ad esempio la rugosità della superficie o il restringimento, mentre il consumo di energia o la degradazione di un utensile sono esempi di qualità specifiche di processo. L'ottimizzazione di entrambi gli obiettivi migliora la qualità di prodotto in termini di costi, tempo e consumo di risorse. Al fine di ottimizzare i parametri di processo, vengono usualmente condotte le seguenti attività (Köksal et al., 2011):

1. Generazione di un database di esperimenti o simulazioni
2. Modellazione delle correlazioni fisiche tra i parametri di processo e i criteri di qualità attraverso tecniche di ML
3. Ottimizzazione dei parametri di processo utilizzando i modelli creati al punto precedente
4. Aggiustamento manuale o automatico dei parametri di processo.

Nei processi meccanici intelligenti, che adottano quindi le tecniche di ML, i parametri di processo possono essere ottimizzati in tempo reale e in modo automatico, ottenendo ottime performance di processo e ottime caratteristiche di qualità del prodotto. Una revisione completa dei diversi modelli di ML utilizzati in questo campo è fornita da Kim et al. (2018).

Molte industrie stanno adottando il ML per migliorare i loro processi. Sono nate pertanto collaborazioni con diverse industrie, con lo scopo di introdurre l'apprendimento automatico nei processi. Una tra queste è la collaborazione con GEA, un'impresa operante nel settore di progettazione e costruzione di macchinari per l'imbottigliamento di bevande in contenitori di plastica, soprattutto nel settore del riempimento in asettico e prodotti sensibili. Questa azienda ha sviluppato un sistema di soffiaggio e riempimento in asettico. In questa tipologia di impianto la preforma di PET viene sterilizzata con vapori di acqua ossigenata e soffiata con una soffiatrice asettica a cui seguono il riempimento e tappatura in sterilità.

Il progetto è diviso in due parti, una dedicata al miglioramento della parte di soffiaggio della bottiglia e una parte al riempimento. In questi mesi, l'attività di collaborazione con GEA ha previsto una presa visione degli impianti

presso le sedi di Roncade (TV) e Sala Baganza (PR) con definizione delle due problematiche relative alla parte dell'area *blower*:

- ottimizzazione della pressione di compensazione dello stampo
- ottimizzazione del riscaldamento delle preforme.

Individuati i due casi da ottimizzare, sono stati definiti gli input e gli output di processo caratteristici del processo produttivo di GEA. Inoltre, è stato avviato il coordinamento con il gruppo di UNIMORE supervisionato dal Prof. Guerra per le attività di implementazione delle tecniche di machine learning ai fini dell'ottimizzazione dei due problemi rilevati.

Riferimenti

- Alsina, Emanuel F., Chica, M., Trawiński, K., & Regattieri, A. (2018). On the use of machine learning methods to predict component reliability from data-driven industrial case studies. *International Journal of Advanced Manufacturing Technology*, 94(5–8), 2419–2433. <https://doi.org/10.1007/s00170-017-1039-x>
- Alsina, Emanuel Federico, Cabri, G., & Regattieri, A. (2016). A Neural Network Approach to Find The Cumulative Failure Distribution: Modeling and Experimental Evidence. *Quality and Reliability Engineering International*, 32(2), 567–579. <https://doi.org/10.1002/qre.1773>
- Azadeh, A., Saberi, M., Kazem, A., Ebrahimipour, V., Nourmohammadzadeh, A., & Saberi, Z. (2013). A flexible algorithm for fault diagnosis in a centrifugal pump with corrupted data and noise based on ANN and support vector machine with hyper-parameters optimization. *Applied Soft Computing Journal*, 13(3), 1478–1485. <https://doi.org/10.1016/j.asoc.2012.06.020>
- Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 1798–1828. <https://doi.org/10.1109/TPAMI.2013.50>
- Bin, G. F., Gao, J. J., Li, X. J., & Dhillon, B. S. (2012). Early fault diagnosis of rotating machinery based on wavelet packets - Empirical mode decomposition feature extraction and neural network. *Mechanical Systems and Signal Processing*, 27(1), 696–711. <https://doi.org/10.1016/j.ymssp.2011.08.002>
- Calabrese, F., Casto, A., Regattieri, A., & Piana, F. (2018). Components monitoring and intelligent diagnosis tools for Prognostic Health Management approach. *Proceedings of the Summer School Francesco Turco, 2018-Sette*, 142–148.
- Cerrada, M., Sánchez, R. V., Cabrera, D., Zurita, G., & Li, C. (2015). Multi-stage feature selection by using genetic algorithms for fault diagnosis in gearboxes based on vibration signal. *Sensors (Switzerland)*, 15(9), 23903–23926. <https://doi.org/10.3390/s150923903>
- Escobar, C. A., & Morales-Menendez, R. (2018). Machine learning techniques for quality control in high conformance manufacturing environment. *Advances in Mechanical Engineering*, 10(2), 1–16. <https://doi.org/10.1177/1687814018755519>
- Gan, M., Wang, C., & Zhu, C. (2015). Multiple-domain manifold for feature extraction in machinery fault diagnosis. *Measurement: Journal of the International Measurement Confederation*, 75, 76–91. <https://doi.org/10.1016/j.measurement.2015.07.042>
- Geropp, B. (1997). Envelope Analysis - A Signal Analysis Technique for Early Detection and Isolation of Machine Faults. *IFAC Proceedings Volumes*, 30(18), 977–981. [https://doi.org/https://doi.org/10.1016/S1474-6670\(17\)42527-4](https://doi.org/https://doi.org/10.1016/S1474-6670(17)42527-4)
- Gokhale, M. Y., & Khanduja, D. K. (2010). Time Domain Signal Analysis Using Wavelet Packet Decomposition Approach. *International Journal of Communications, Network and System Sciences*, 03(03), 321–329. <https://doi.org/10.4236/ijcns.2010.33041>
- Gomes Teixeira de Almeida, R., Alexandra da Silva Vicente, S., & Rodrigues Padovese, L. (2002). New technique for evaluation of global vibration levels in rolling bearings. In *Shock and Vibration* (Vol. 9). IOS Press.
- Han, T., Jiang, D., Zhang, X., & Sun, Y. (2017). Intelligent diagnosis method for rotating machinery using dictionary learning and singular value decomposition. *Sensors (Switzerland)*, 17(4), 1–17. <https://doi.org/10.3390/s17040689>
- Hu, R., Zhu, X., Cheng, D., He, W., Yan, Y., Song, J., & Zhang, S. (2017). Graph self-representation method for unsupervised feature selection. *Neurocomputing*, 220, 130–137. <https://doi.org/10.1016/j.neucom.2016.05.081>
- Jardine, A. K. S., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 20(7), 1483–1510. <https://doi.org/10.1016/j.ymssp.2005.09.012>
- Kim, D. H., Kim, T. J. Y., Wang, X., Kim, M., Quan, Y. J., Oh, J. W., Min, S. H., Kim, H., Bhandari, B., Yang, I., & Ahn, S. H. (2018). Smart Machining Process Using Machine Learning: A Review and Perspective on Machining Industry. *International Journal of Precision Engineering and Manufacturing - Green Technology*, 5(4), 555–568.

- <https://doi.org/10.1007/s40684-018-0057-y>
- Köksal, G., Batmaz, I., & Testik, M. C. (2011). A review of data mining applications for quality improvement in manufacturing industry. *Expert Systems with Applications*, *38*(10), 13448–13467. <https://doi.org/10.1016/j.eswa.2011.04.063>
- Lee, J., Wu, F., Zhao, W., Ghaffari, M., Liao, L., & Siegel, D. (2014). Prognostics and health management design for rotary machinery systems - Reviews, methodology and applications. *Mechanical Systems and Signal Processing*, *42*(1–2), 314–334. <https://doi.org/10.1016/j.ymssp.2013.06.004>
- Lei, Y. (2016). *Intelligent Fault Diagnosis and Remaining Useful Life Prediction of Rotating Machinery* (Butterworth-Heinemann (ed.)). Elsevier Science. <https://books.google.it/books?id=fTJQDAAQBAJ>
- Lei, Yaguo, Jia, F., Lin, J., Xing, S., & Ding, S. X. (2016). An Intelligent Fault Diagnosis Method Using Unsupervised Feature Learning Towards Mechanical Big Data. *63*(5), 3137–3147. <https://doi.org/10.1109/TIE.2016.2519325>
- Lei, Yaguo, Li, N., Guo, L., Li, N., Yan, T., & Lin, J. (2018). Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing*, *104*, 799–834. <https://doi.org/10.1016/j.ymssp.2017.11.016>
- Li, L., Qu, L., & Liao, X. (2007a). Haar wavelet for machine fault diagnosis. *Mechanical Systems and Signal Processing*, *21*(4), 1773–1786. <https://doi.org/10.1016/j.ymssp.2006.07.006>
- Li, L., Qu, L., & Liao, X. (2007b). Haar wavelet for machine fault diagnosis. *Mechanical Systems and Signal Processing*, *21*(4), 1773–1786. <https://doi.org/10.1016/j.ymssp.2006.07.006>
- Liang, B., Iwnicki, S. D., & Zhao, Y. (2013). Application of power spectrum, cepstrum, higher order spectrum and neural network analyses for induction motor fault diagnosis. *Mechanical Systems and Signal Processing*, *39*(1–2), 342–360. <https://doi.org/10.1016/j.ymssp.2013.02.016>
- Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2009). Continuous control with deep reinforcement learning. *Foundations and Trends in Machine Learning*, *2*(1), 1–127. <https://doi.org/10.1561/22000000006>
- Lin, J., & Qu, L. (2000). Feature extraction based on morlet wavelet and its application for mechanical fault diagnosis. *Journal of Sound and Vibration*, *234*(1), 135–148. <https://doi.org/10.1006/jsvi.2000.2864>
- Lippi, V., & Ceccarelli, G. (2019). Incremental principal component analysis: Exact implementation and continuity corrections. *ICINCO 2019 - Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics*, *1*, 473–480. <https://doi.org/10.5220/0007743604730480>
- Liu, H., Liu, C., & Huang, Y. (2011). Adaptive feature extraction using sparse coding for machinery fault diagnosis. *Mechanical Systems and Signal Processing*, *25*(2), 558–574. <https://doi.org/10.1016/j.ymssp.2010.07.019>
- Liu, R., Yang, B., Zio, E., & Chen, X. (2018). Artificial intelligence for fault diagnosis of rotating machinery: A review. *Mechanical Systems and Signal Processing*, *108*, 33–47. <https://doi.org/10.1016/j.ymssp.2018.02.016>
- Lolli, F., Gamberini, R., Regattieri, A., Balugani, E., Gatos, T., & Gucci, S. (2017). Single-hidden layer neural networks for forecasting intermittent demand. *International Journal of Production Economics*, *183*, 116–128. <https://doi.org/10.1016/j.ijpe.2016.10.021>
- Mehrjou, M. R., Mariun, N., Misron, N., & Radzi, M. A. M. (2017). Analysis of statistical features based on start-up current envelope for broken rotor bar fault detection in line start permanent magnet synchronous motor. *Electrical Engineering*, *99*(1), 187–201. <https://doi.org/10.1007/s00202-016-0404-3>
- Mobley, R. K. (2002). Role of Maintenance Organization. In *An Introduction to Predictive Maintenance* (pp. 43–59). <https://doi.org/10.1016/b978-075067531-4/50003-8>
- Peng, Z. K., & Chu, F. L. (2004). Application of the wavelet transform in machine condition monitoring and fault diagnostics: A review with bibliography. *Mechanical Systems and Signal Processing*, *18*(2), 199–221. [https://doi.org/10.1016/S0888-3270\(03\)00075-X](https://doi.org/10.1016/S0888-3270(03)00075-X)
- Rai, A., & Upadhyay, S. H. (2017). Bearing performance degradation assessment based on a combination of empirical mode decomposition and k-medoids clustering. *Mechanical Systems and Signal Processing*, *93*, 16–29. <https://doi.org/10.1016/j.ymssp.2017.02.003>
- Ricci, R., & Pennacchi, P. (2011). Diagnostics of gear faults based on EMD and automatic selection of intrinsic mode functions. *Mechanical Systems and Signal Processing*, *25*(3), 821–838. <https://doi.org/10.1016/j.ymssp.2010.10.002>
- Smola, A. J., & Olkoph, B. S. C. H. (2004). *A tutorial on support vector regression* *". 199–222.
- Tang, J., Alelyani, S., & Liu, H. (2014). Feature Selection for Classification: A Review. *Data Classification: Algorithms and Applications*, 37–64. <https://doi.org/10.1.1.409.5195>
- Wang, D., Nie, F., & Huang, H. (n.d.). *Unsupervised Feature Selection via Unified Trace Ratio Formulation and K-means Clustering (TRACK)*.
- Wang, F., Sun, J., Yan, D., Zhang, S., Cui, L., & Xu, Y. (2015). A feature extraction method for fault classification of rolling bearing based on PCA. *Journal of Physics: Conference Series*, *628*(1). <https://doi.org/10.1088/1742-6596/628/1/012079>
- Weichert, D., Link, P., Stoll, A., Rüping, S., Ihlenfeldt, S., & Wrobel, S. (2019). A review of machine learning for the optimization of production processes. *International Journal of Advanced Manufacturing Technology*, *104*(5–8),

1889–1902. <https://doi.org/10.1007/s00170-019-03988-5>

- Widodo, A., & Yang, B. S. (2007). Support vector machine in machine condition monitoring and fault diagnosis. In *Mechanical Systems and Signal Processing* (Vol. 21, Issue 6, pp. 2560–2574). <https://doi.org/10.1016/j.ymsp.2006.12.007>
- Xie, T., Ren, P., Zhang, T., & Tang, Y. Y. (2018). Distribution preserving learning for unsupervised feature selection. *Neurocomputing*, 289, 231–240. <https://doi.org/10.1016/j.neucom.2018.02.032>
- Zhang, C., Li, E., Liu, L., & Zhou, Y. (2018). Dimension Reduction Method For Data-Driven PHM Sysyem Based on ULDA Algorithm. *ICIC Express Letters*, 9(8), 765–771. <https://doi.org/10.24507/icicelb.09.08.765>
- Zhang, X., Zhang, Q., Li, H., Sun, Y., & Qin, X. (2017). Fault Diagnosis Using Locality Sensitive Discriminant Analysis for Feature Extraction. *2017 Prognostics and System Health Management Conference (PHM-Harbin)*, 1–6. <https://doi.org/10.1109/PHM.2017.8079198>
- Zhu, X., Li, X., Zhang, S., Ju, C., & Wu, X. (2017). Robust Joint Graph Sparse Coding for Unsupervised Spectral Feature Selection. *IEEE Transactions on Neural Networks and Learning Systems*, 28(6), 1263–1275. <https://doi.org/10.1109/TNNLS.2016.2521602>
- Zhu, Y., Zhang, X., Wang, R., Zheng, W., & Zhu, Y. (2018). Self-representation and PCA embedding for unsupervised feature selection. *World Wide Web*, 21(6), 1675–1688. <https://doi.org/10.1007/s11280-017-0497-2>