Università degli Studi di Padova

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE
Corso di Laurea Magistrale in Ingegneria dell'automazione

Tesi di Laurea Magistrale

Approcci di Machine Learning per il Riconoscimento di Fault e Anomalie in Macchine Utensili

Machine Learnig Approches for Fault and Anomaly Detection in Machine Tools

Candidato:
Marco Bessegato
Matricola 1156480

Relatore:
Prof. Gian Antonio Susto

Correlatori:
Diego Turesso
Federico Milan

Anno Accademico 2018-2019

A mia madre, primo consigliere e strenua sostenitrice

A mio padre, che mi ispira con la sua instancabile voglia di migliorarsi

A mia sorella, che mi sprona a essere sempre una persona migliore

A Claudia, compagna di viaggio insostituibile

Ai miei amici, per la loro costante presenza

Ai miei famigliari, per mostrarmi cosa significhi vivere una vita condivisa

Sommario

Questo progetto di tesi è partito con un obiettivo specifico: cercare di stabilire se la lavorazione svolta da una macchina utensile è avvenuta correttamente (ovvero rispetta le specifiche e segue le indicazioni) oppure è avvenuta in maniera impropria (oltre le specifiche consigliate, parametri non consoni). Comprendere e stabilire una distinzione netta tra questi due tipi di lavorazione grazie allo studio dei livelli di vibrazione risulterebbe sicuramente un punto di forza notevole per l'azienda, in quanto macchine con questa funzionalità aggiuntiva possono abilitare strategie di manutenzione predittiva e riscontro di anomalie ed essere quindi più performanti e appetibili sul mercato.

La parte iniziale della tesi è stata dedicata allo studio delle macchine utensili e delle vibrazioni e al confronto di vari accelerometri. Lo scopo era quello di trovare la posizione migliore per descrivere le vibrazioni presenti della macchina utensile, in particolare del mandrino e dell'utensile.

Una volta identificata una posizione ottimale per gli accelerometri e ottenuto un dataset che contenesse prove errate e prove corrette, la seconda parte del progetto è stata rivolta allo studio di tecniche di *Machine Learning* che identificassero l'errore in un approccio supervisionato (*Fault Detection*) o che stabilissero il grado di anomalia in un approccio non supervisionato (*Anomaly Detection*).

Indice

Capitolo 1

Introduzione

La rivoluzione del *Machine Learning* (apprendimento automatico) è ormai inarrestabile. La sua presenza è ovunque e trasversale: dai software degli smartphone alle automobili, dai programmi di riconoscimento vocale al settore finanziario, dai sistemi di raccomandazione alla medicina.

L'apprendimento automatico non solo consente di velocizzare processi lenti e sequenziali, ma permette inoltre in molti casi di evitare la modellizzazione fisica di un sistema la quale è un'operazione solitamente molto onerosa, perché richiede una conoscenza e un'esperienza notevole del dominio, che fornisce spesso una descrizione parziale di fenomeni complessi.

L'approccio viene totalmente modificato: non si cerca più di programmare la macchina passo dopo passo descrivendo minuziosamente la procedura che essa andrà a svolgere, piuttosto si cerca di fornire un set di dati di training inseriti in un generico algoritmo affinché il programma sviluppi una propria logica per risolvere l'attività o la funzione richiesti. Ne consegue che la possibilità di accedere a un numero sempre maggiore di dati di training diventi una condizione necessaria per migliorare le prestazioni che un algoritmo di Machine Learning potrà fornire.

Per un'azienda che cerchi uno sviluppo costante e un'innovazione continua, abbracciare questo cambiamento diventa dunque fondamentale, anche e soprattutto per restare competitivi sul mercato.

Da questa idea parte la mia collaborazione con Breton SpA, azienda leader nella produzione di centri di lavoro verticali ad alta precisione ed alta velocità per la lavorazione della pietra, di alluminio, di acciaio, di compositi e resina.

L'obiettivo generale di questo progetto è stato quello di convogliare i temi del Machine Learning verso un'attività che avesse anche una valenza pratica e potesse portare benefici all'azienda.

Per le aziende produttrici, tra cui Breton SpA stessa, i costi derivanti direttamente e indirettamente da attività di supporto o assistenza (così pure dalle eventuali sostituzioni o riparazioni di equipment), necessarie a mantenere alta la soddisfazione dei clienti, incidono pesantemente sul costo totale anche dopo la vendita sia per l'azienda che per lo stesso cliente; per esempio, senza l'utilizzo di strumenti di analisi e predizione evoluti, non di rado è necessario inviare personale tecnico specializzato presso i clienti con relativi costi di viaggio, vitto ed alloggio. A seconda del tipo di anomalia riscontrata, i costi sostenuti possono essere a carico del cliente (costo percepito dal cliente) o dell'azienda produttrice, erodendo così il margine ed il guadagno. Riuscire a rilevare e stabilire la causa dell'anomalia fin dai primi istanti in cui essa si manifesta diventa fondamentale per fornire il giusto servizio al cliente e per valutare il tipo di assistenza.

Da questo punto di vista l'utilizzo di paradigmi di industria 4.0 e di analisi guidate dai dati consente di sviluppare percorsi alternativi per affrontare queste tematiche. In quest'ottica si manifesta uno degli aspetti chiave del Machine Learning, il quale consente di ridefinire modelli di business e di miglioramento dei servizi preposti alla soddisfazione del cliente, esplorando così nuove opportunità per le aziende produttrici e per i clienti.

L'utilizzo di metodi di Machine Learning, sfruttando le informazioni derivanti da sensori reali o virtuali, potrebbe permettere di realizzare previsioni sempre più accurate che si possono concretizzare sia in una riduzione dei tempi di intervento presso il cliente sia in una pianificazione anticipata del tipo di intervento e delle risorse necessarie in modo tale da permettere un reale risparmio per l'azienda produttrice e un tempo ridotto di non produttività per il cliente (soddisfazione e cura del cliente stesso).

Da queste considerazioni nasce l'obiettivo primario del progetto: cercare di stabilire se il lavoro svolto con una macchina utensile da parte di un operatore sia appropriato (ovvero rispetta le specifiche e segue le indicazioni) oppure inappropriato (oltre le specifiche e lavorazioni non consone) attraverso lo studio dei livelli di vibrazione con algoritmi di Machine Learning. Lo sviluppo di queste tecnologie renderebbe sicuramente questo tipo di macchine utensili più performanti e di conseguenza anche più appetibili sul mercato .

1.1 Industria 4.0

I temi del Machine Learning sono strettamente interconnessi con la nuova rivoluzione industriale che sta conducendo a un nuovo modello di industria denominato *Industria 4.0*.
La storia è stata segnata da quattro grandi rivoluzioni industriali, compresa quella attuale, ognuna delle quali ha generato dirompenti cambiamenti sulla produttività:

- *Prima rivoluzione industriale*: Grazie all'introduzione della macchina a vapore, la produzione si sgancia dalla sola forza fisica o animale utilizzata fin lì. Si introduce l'utilizzo di macchine azionate da energia meccanica.

- *Seconda rivoluzione industriale*: L'utilizzo dell'energia elettrica allarga le dimensioni dei mercati, si sviluppano le linee di assemblaggio. Inoltre vengono introdotti il petrolio e altri elementi chimici.

- *Terza rivoluzione industriale*: Viene automatizzato il processo grazie all'industria informatica e all'elettronica, vengono introdotto i robot e si riduce di molto il tempo di produzione.

- *Quarta rivoluzione industriale*: Prevede la digitalizzazione del sistema e lo sviluppo di macchine e sensori intelligenti interconnessi in grado di monitorare e correggere in tempo reale ogni singolo dettaglio del processo. Si cerca di sfruttare la mole enormi di dati e informazioni attraverso analisi e elaborazioni complesse.

Le direttrici principali di questo nuova rivoluzione riguardano soprattutto: la digitalizzazione, per misurare e ottenere valori da un fenomeno analogico; l'interconnessione, per scambiare informazioni e dati in modo veloce e rendere i sistemi intelligenti e comunicanti; l'elaborazione real time, per ottenere sistemi in grado di essere modificati in maniera istantanea anche da remoto.
Tuttavia per comprendere appieno questo nuovo modello produttivo bisogna capire chi è il vero attore principale ovverosia il *dato*.
Come recita uno slogan nel sito di Breton, infatti: *"Il vero protagonista di Industry 4.0 è il dato, tutto il contenuto tecnologico deve essere digitalizzato e condiviso in tempo reale tra tutti gli utenti interessati."*
Il dato è il responsabile del cambiamento; l'opportunità di misurare, caratterizzare, virtualizzare un processo rende infinite le possibilità di miglioramento e ottimizzazione del prodotto aziendale. É

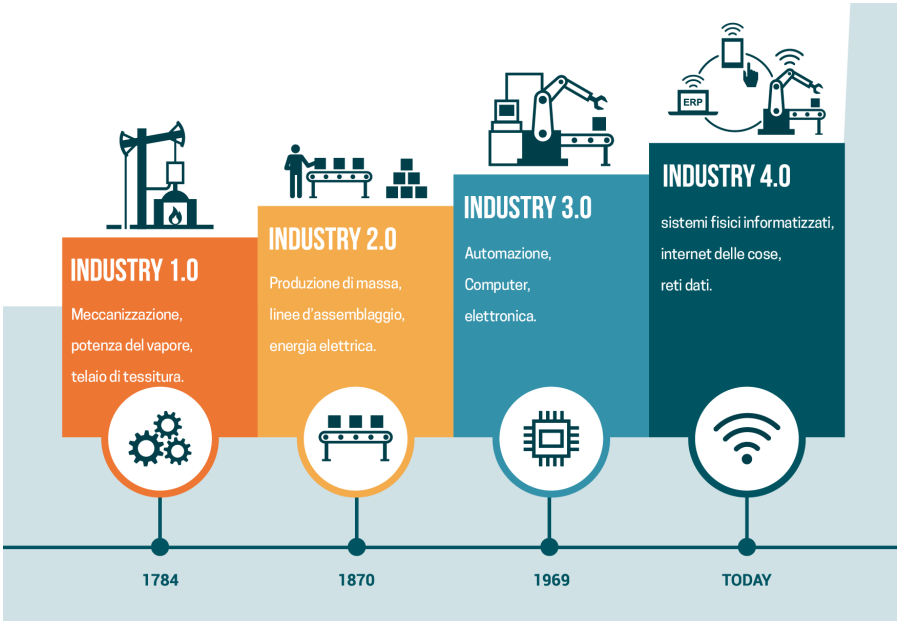


Figura 1.1: Rivoluzioni industriali

grazie all'enorme quantità di dati che si è in grado di raccogliere dalla rete di macchine e sensori tra loro interconnessi che è possibile migliorare l'intera catena produttiva, analizzare nel dettaglio le debolezze del sistema e sfruttare la tecnologia per innovare e rendere il prodotto moderno e non convenzionale.

In questa situazione l'approccio di Breton è stato quello di sviluppare autonomamente il software *Sentinel*, composto da un pacchetto di applicazioni modulari al quale possono essere aggiunte nuove funzionalità e da un nucleo hardware facilmente implementabile. Questo software non solo riesce a fornire le funzionalità più avanzate come il monitoraggio in tempo reale dei parametri e delle performance della macchina e l'assistenza alla manutenzione preventiva, ma anche gestire i dati sia in locale sia in rete appoggiandosi a server sicuri. Altre funzionalità implementate riguardano il calcolo degli indici di efficienza, il monitoraggio dello stato degli utensili, la lavorazione adattiva e la guida alla risoluzione guasti.

La soluzione sviluppata "in casa" nonostante richieda un investimento maggiore in termini di tempo, risorse e rischio è stata preferita perché la padronanza dell'intero ciclo di gestione del dato e la conoscenza approfondita del processo di creazione dei macchinari stessi, consente nel lungo periodo di aumentare al massimo la produttività dell'impianto.

Si vedrà in seguito come *Sentinel* sia risultato utile a catturare caratteristiche che insieme ad altri parametri hanno formato il dataset da cui si è partiti per riconoscere le anomalie.

Lo sviluppo di questo progetto, e allo stesso modo della tesi, si è svolto essenzialmente in tre fasi. La prima fase, corrispondente al Capitolo 2, ha riguardato lo studio della strumentazione; si è cercato dunque di comprendere non solo il funzionamento di una macchina utensile e delle sue parti più critiche ma anche di analizzare e studiare il fenomeno delle vibrazioni le quali incidono maggiormente sul mandrino e sull'utensile. In particolare poi si è cercato di analizzare gli accelerometri, ovvero gli strumenti selezionati per catturare le vibrazioni. Dopo aver valutato le specifiche e i diversi metodi di montaggio, è stata eseguita una breve panoramica di confronto degli accelerometri a disposizione.

Una volta descritto il settaggio, nella fase successiva si è passati a svolgere le prove necessarie per la creazione del dataset (Capitolo 3). Le prove sono state diverse e sono servite, innanzitutto, per stabilire la posizione che meglio descrivesse il fenomeno e in seguito per analizzare le due diverse tipologie di lavorazione: lavorazione corretta e lavorazione non corretta. Dai segnali di vibrazione registrati si sono estratte della features che descrivono in maniera accurata il fenomeno, creando alla fine un dataset contenente una collezione di dati informativi sul segnale di vibrazione.

Dopo aver affrontato uno studio generale sulle peculiari caratteristiche del machine learning (Capitolo 4), nella fase finale di analisi dei dati sono stati affrontati due tipi di approcci diversi: l'approccio supervisionato (Capitolo 5) e l'approccio non supervisionato (Capitolo 6). La differenza è sostanziale: mentre nel primo caso si ha a disposizione un dataset che presenta un dato che esplicitamente conferma l'appartenenza alla classe di lavorazione corretta o non, nel secondo caso questo tipo di informazione è mancante e si cerca di riconoscere l'anomalia dallo studio del resto dei dati. Di conseguenza nell'approccio supervisionato si è cercato un algoritmo di machine learning che tentasse di discernere in maniera corretta l'appartenenza a una delle due classi valutandone l'accuratezza e nell'approccio non supervisionato si è cercato un algoritmo di anomaly detection che tentasse di identificare correttamente il dato che fosse anomalo rispetto al resto delle lavorazioni.

Infine nel Capitolo 7 si sono riportate le conclusioni dell'analisi con i migliori algoritmi individuati e si sono proposti possibili sviluppi futuri di continuazione di questo progetto.

Capitolo 2

Set-Up

2.1 Macchine Utensili

Essenzialmente il *core business* di Breton S.P.A. si sviluppa attorno alle macchine utensili a 5 o più assi per la fresatura e la tornitura. Sono entrambe tipi di lavorazione meccanica per asportazione di truciolo (ovvero da un materiale grezzo viene asportato materiale in eccesso) ma mentre la tornitura prevede un moto rotatorio del pezzo e un moto traslatorio dell'utensile tagliente, la fresatura al contrario richiede la rotazione dell'utensile e la traslazione del pezzo che viene ancorato al banco. La gamma dei centri di lavori prodotti si focalizza soprattutto sui settori dell'aerospace, dell'automotive, della meccanica generale e degli stampi.
Il settore sul quale si è deciso di concentrare gli sforzi sono le macchine utensili di medie dimensioni, che richiedono una precisione anche superiore al centesimo di millimetro e determinate performance di velocità, possedendo anche un sistema di controllo molto complesso. In particolare la macchina utensile su cui si è fatto riferimento è la Ultrix, centro di lavoro verticali multifunzione, adibito alla lavorazione dell'acciaio, alluminio e superlega.

Innanzitutto è una macchina a controllo numerico a 5 assi: unisce alla lavorazione in larghezza, altezza e profondità la rotazione della testa porta utensile e del pezzo in lavorazione. Il movimento degli assi X ed Y avviene su guide lineari con pattini controllato da servomotori brushless digitali. Il movimento dell'asse Z avviene sempre grazie a un motore brushless digitale.
Come si nota dalla Figura 2.1 è presente una tavola basculante che garantisce precisione, durata e affidabilità sia per la tornitura che per la fresatura di pezzi sia piccoli che grandi. É una tavola

Figura 2.1: Ultrix 800

rototiltante e può sorreggere fino a due tonnellate; possiede inoltre un autocentrante e il bloccaggio idraulico del pezzo il quale permette di svolgere l'operazione di bloccaggio e sbloccaggio velocemente. Le ampie porte con grandi finestre agevolano la visibilità e la completa supervisione del campo operativo.

Il componente principale di una macchina utensile è sicuramente l'elettromandrino, che consente di generare il moto rotatorio di lavoro dell'utensile.
L'elettromandrino è composto da: un motore elettrico, un albero rotante e un sistema di controllo della velocità e della posizione dell'albero, dei cuscinetti e da altri parti che consentono la movimentazione, l'attacco e lo sgancio dell'utensile. Il motore elettrico è una macchina elettrica che trasforma una potenza di ingresso di tipo elettrico in un movimento di tipo meccanico che agisce nell'ambiente. A seconda della scelta si può avere un motore di tipo sincrono in cui la frequenza di alimentazione è multiplo di quella di rotazione ,o asincrono in cui le due frequenze sono diverse. Negli elettromandrini il più usato è il motore induttivo a corrente alternata in cui il rotore è attaccato direttamente all'albero motore.
Il motore elettrico e i cuscinetti sono fonti di riscaldamento non irrilevanti, risulta perciò necessaria

la presenza di un sistema di raffreddamento, in particolare attorno al motore elettrico è presente una protezione in cui scorre il liquido refrigerante.

L'albero motore invece, svolge due compiti principali: trasferire la potenza dal motore all'utensile, e sorreggere e posizionare i cuscinetti. Questi ultimi sono tra gli elementi più delicati e soggetti a rotture e usura nel tempo di tutto il sistema. Sono posizionati tra l'albero (parte rotante) e il supporto esterno (parte fissa). Sono composti da due anelli di diverso diametro, uno più interno a contatto con l'albero, uno più esterno a contatto con il supporto, entro i quali sono inserite le sfere (si parla infatti di *"cuscinetti a sfera"*).

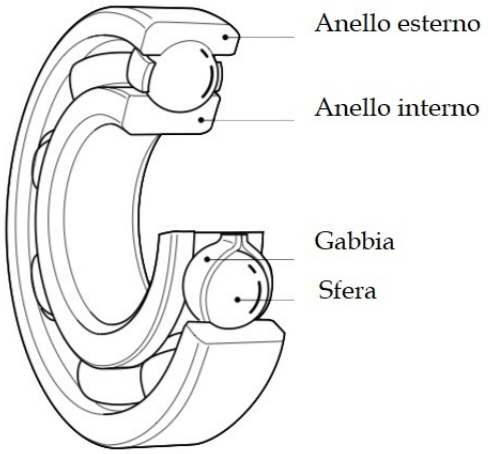


Figura 2.2: Cuscinetti a sfera

Le sfere sono di materiale ceramico; con questo materiale si riesce a ottenere performance fino al 30% superiori a quelle composte da acciaio, grazie anche al fatto che si riduce l'espansione termica e la suscettibilità alle vibrazioni.

Per proteggere la superficie dalle collisioni e evitare un degrado precoce, i cuscinetti devono essere costantemente lubrificati. Il sistema di lubrificazione inietta il liquido lubrificante nelle cavità dei cuscinetti fornendo una microscopica pellicola tra gli elementi rotanti,la quale diminuisce notevolmente l'attrito volvente. La lubrificazione può essere di due tipi: la prima, a grasso, che prevede un sistema semplificato a basso costo, utilizzato per soluzioni permanenti a basse velocità e basso numero di giri (inferiori ai $14000\ rpm$); la seconda ad aria e olio, che consente una lubrificazione a cicli che può essere regolata, un'efficienza maggiore sia in termini di attrito sia in termini di calore generato. La

lubrificazione ad aria e olio prevede la presenza di una centralina di lubrificazione con conseguente aumento del costo; risulta particolarmente adatta quanto si vogliono raggiungere rotazioni molto alte.

I cuscinetti sono fondamentali per supportare l'elemento ruotante, e contemporaneamente ridurre l'attrito di rotazione. A seconda della direzione di applicazione del carico, si dividono in radiali, assiali e obliqui. É l'angolo di contatto α_0 che determina il rapporto tra il carico assiale e il carico radiale. Minore è l'angolo di contatto, maggiore sarà la capacità di sopportare un carico radiale; al contrario, maggiore è l'angolo di contatto, più alta è la capacità di sopportare un carico assiale.

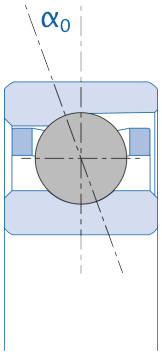


Figura 2.3: Angolo di contatto

Risulta difficile calcolare con esattezza il ciclo di vita di un cuscinetto, ovvero stabilire la quantità di tempo fino al verificarsi della prima sfaldatura sulla pista degli anelli o sulla sfera. Questo deterioramento può causare uno sbilanciamento e disallineamento di tutto il sistema, aggravando in maniera pesante lo sforzo torcente e riducendo sensibilmente il tempo di utilizzo. I fattori che contribuiscono a determinare la durata di vita di un cuscinetto sono diversi, i più influenti e maggiormente individuabili sono:

- carichi sui cuscinetti
- livelli di vibrazione
- qualità e quantità della lubrificazione
- massima velocità
- temperatura media dei cuscinetti

Un calcolo approssimativo per la durata teorica di base, L_{10h} (in ore), considerato solo il carico e la velocità, può essere fornito dalla seguente equazione[1]:

$$L_{10h} = \frac{1}{60n}\left(\frac{C}{P}\right)^3 \tag{2.1}$$

dove C è il coefficiente di carico dinamico di base in kN, P è il carico dinamico equivalente in kN e n la velocità di rotazioni in rpm. Il valore di P è definito come un carico ipotetico, costante per entità e direzione, che agisce in direzione radiale sui cuscinetti radiali e in maniera assiale su quelli assiali. Si ipotizza che l'applicazione di questo carico produca gli stessi effetti sulla durata del cuscinetto dei carichi effettivi a cui il cuscinetto è soggetto. Se sul cuscinetto agiscono la forza radiale F_r e la forza assiale F_a, la formula per trovare P è[2][3]:

$$P = XF_r + YF_a \tag{2.2}$$

in cui X e Y sono rispettivamente fattori relativi al carico radiale e assiale.

Infine un'ultima considerazione sull'elettromandrino riguarda la presenza di un encoder, a non contatto, che permette di rilevare velocità e posizione del mandrino, necessario soprattutto per svolgere l'operazione di cambio dell'utensile.
Dalle caratteristiche interne dell'elettromandrino si determinano le specifiche; la coppia motrice può variare da $50/100\,[N \cdot m]$ fino a $3000\,[N \cdot m]$ e da $12000\ rpm$ fino a oltre $28000\ rpm$. Per ulteriori dettagli sulle specifiche delle macchine utensili Ultrix si rimanda all'appendice A.

2.2 Vibrazione

E' stato visto come uno dei fattori che influiscono maggiormente sulla vita dei cuscinetti sono le vibrazioni. Una vibrazione a vuoto superiore a $7\ mm/s^2$ può ridurre la vita dei cuscinetti fino a un sesto della stessa.
In letteratura sono già presenti ricerche e metodi che sfruttano machine learning e vibrazioni per valutare lo stato di salute dei cuscinetti: in particolare studi recenti mostrano l'utilizzo di diversi tipi di reti neurali per identificare la condizione di degrado del cuscinetto[4], spesso collegati ad una fase

di preprocessing iniziale[5]; altri tipi di ricerche presentano invece la valutazione dello stato di salute dei cuscinetti tramite un'analisi statistica delle vibrazioni, cercando di eliminare il rumore gaussiano ottenuto da segnali grezzi[6].

La diversità di questo studio consiste nell'utilizzare le vibrazioni per valutare se la lavorazione è appropriata o meno. Innanzitutto è necessario comprendere cosa siano le vibrazioni.

Le vibrazioni sono descritte come dei moti oscillatori attorno a una posizione di riferimento, i cui parametri (ampiezza, valor medio) possono non essere costanti nel tempo. A seconda delle caratteristiche è possibile dividere la vibrazione in due categorie: le vibrazioni deterministiche, in cui lo sviluppo di alcuni parametri in successive vibrazioni è noto da una storia precedente e le vibrazioni casuali dove la maggior parte delle proprietà non seguono un andamento statistico. Ad esempio nella prima categoria rientrano le vibrazioni generate dagli elettromandrini, che si distinguono per componenti armoniche con pulsazioni multiple della velocità di rotazione degli alberi rotanti (ovviamente presentano anche componenti casuali causate da altri disturbi e da rumore). Tra i moti random con caratteristiche stazionarie rientra, ad esempio, il moto ondoso.

Il comportamento di una vibrazione può essere descritto usando lo spostamento, o la velocità o l'accelerazione (rispettivamente derivata prima e derivata seconda). La scelta tra questi tre parametri dipende essenzialmente dal range di frequenza del fenomeno considerato. Per misure a basse frequenze è preferibile rilevare gli spostamenti, viceversa l'accelerazione enfatizza le componenti ad alta frequenza[7].

Considerando la sinusoide come la più semplice funzione (o armonica) che rappresenti una vibrazione, il moto di una vibrazione è descritto dalle seguenti equazioni in termini di accelerazione, velocità e spostamento[8]:

$$\begin{aligned} a(t) &= A\sin(\omega t + \phi) \\ v(t) &= \int a(t)\,dt = -\frac{A}{\omega}\cos(\omega t + \phi) \\ d(t) &= \iint a(t)\,dt = -\frac{A}{\omega^2}\sin(\omega t + \phi) \end{aligned} \tag{2.3}$$

Per ogni parametro forma e periodo non cambiano, mentre si modificano fase e ampiezza. Trascurando la fase e mediando nel tempo, le relazioni tra i parametri dipendono solo dalla pulsazione

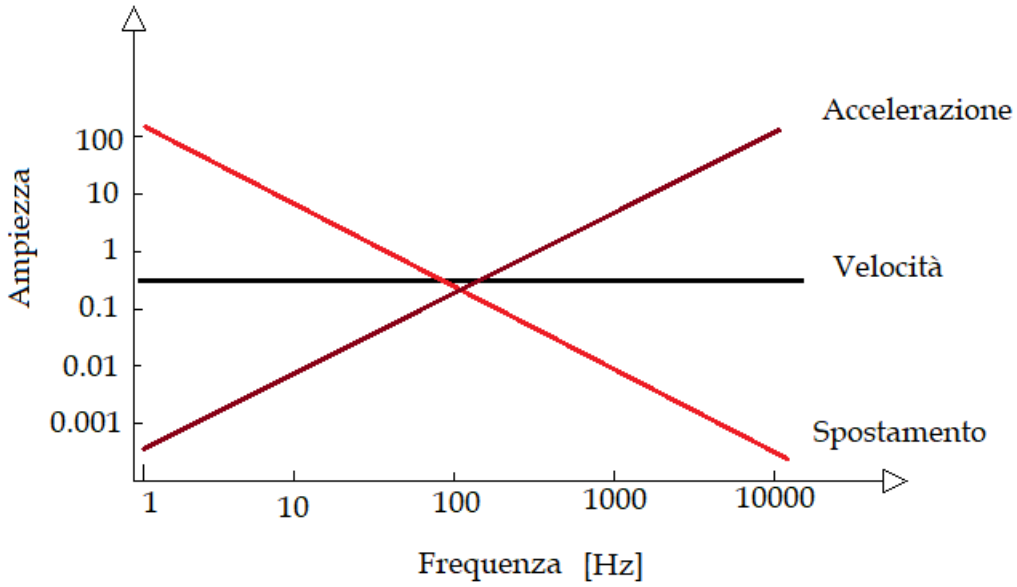


Figura 2.4: Relazione dello spostamento, della velocità, dell'accelerazione relative alla frequenza

$\omega = 2\pi f$:

$$\begin{aligned} a &= A \\ v &= -\frac{A}{\omega} = \frac{A}{2\pi f} \\ d &= -\frac{A}{\omega^2} = \frac{A}{4\pi^2 f^2} \end{aligned} \tag{2.4}$$

I segnali di vibrazione sono caratterizzati da molteplici componenti armoniche aventi differente frequenza e fase; perciò è utile uno studio che comprenda non solo un'analisi nel dominio del tempo con conseguente stima di alcuni parametri di sintesi ma anche un'analisi in frequenza che si rende indispensabile per potere stimare il contributo fornito dalle singole armoniche.
Per quanto riguarda l'analisi nel dominio del tempo si possono individuare alcuni parametri che possono essere utili per confronti con valori di riferimento[9][10]:

- *picco-picco*: indica l'escursione massima della vibrazione, il massimo valore meno il minimo.
- *picco*: indica l'escursione massima positiva o negativa
- *valore quadratico medio (RMS)*: fornisce un valore di ampiezza della vibrazione tenendo conto

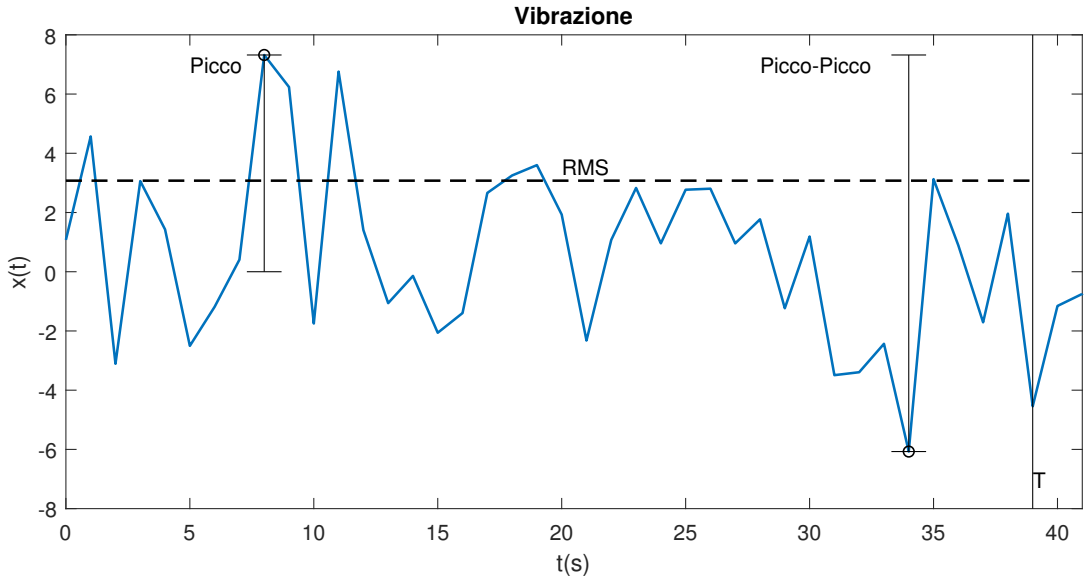


Figura 2.5: RMS, Picco, Picco-Picco

della storia. E' una misura strettamente collegata al contenuto energetico della vibrazione e si ottiene:

$$RMS = \sqrt{\frac{1}{T}\int_0^T x^2(t)dt} \tag{2.5}$$

Nel caso di una vibrazione sinusoidale di ampiezza A il suo valore quadratico medio è dato da $RMS = \frac{A}{\sqrt{2}} \approx 0.707A$, mentre nel caso di segnali digitali di N campioni , la formula per il calcolo del valore RMS è data mediante questo calcolo:

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2} \tag{2.6}$$

- *fattore di cresta*: indica il carattere impulsivo della vibrazione. Viene definito come il rapporto tra il valore di picco e il valore RMS:

$$C_{rf} = \frac{picco}{RMS} \tag{2.7}$$

- *fattore di forma*: è il rapporto tra RMS e il valore medio:

$$S_{hf} = \frac{RMS}{\frac{1}{T}\int_0^T \mid x(t) \mid dt} \tag{2.8}$$

- *kurtosis*: è un metodo statistico che calcola la distribuzione dei dati. Un valore alto di kurtosis indica un dataset con molti valori che si distanziano dalla distribuzione normale, al contrario un valore basso esprime una generale mancanza di valori presenti nella coda della distribuzione.

$$K_v = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^4}{RMS^4} \tag{2.9}$$

- *skewness*: è una misura che riguarda la simmetria, più precisamente la mancanza di simmetria. Dati che rappresentano una distribuzione quasi simmetrica hanno una skewness prossima allo zero.

$$S_k = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^3}{\left(\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2}\right)^3} \tag{2.10}$$

L'altro tipo di analisi, ovvero quella in frequenza, permette invece di catturare le componenti spettrali della vibrazione. Lo strumento usato per ottenere il segnale in frequenza è la trasformata di Fourier[11] la quale ci permette di scomporre il segnale in un numero infinito di armoniche elementari caratterizzate dall'ampiezza A_n dalla pulsazione ω_n e dalla fase iniziale ϕ_n.
La formula generale della *trasformata di Fourier* è:

$$X(f) = \int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft}dt \tag{2.11}$$

Tuttavia in generale si dispone di segnali di tipo digitale per cui i coefficienti di Fourier si ottengono non più per integrazione ma per sommatoria (*trasformata di Fourier discreta*):

$$X_k = \sum_{k=0}^{N-1} x_n e^{-jk\frac{2\pi}{N}n} \quad k = 0, ..., N-1 \tag{2.12}$$

Oltre a descrivere il segnale in frequenza attraverso i parametri elencati in precedenza, è possibile attraverso questa analisi cercare di scovare frequenze fondamentali che sono proprie del sistema o che potrebbero essere indicative di un comportamento non consono da parte del sistema, soprattutto dei cuscinetti. A questo proposito infatti si introducono le frequenze fondamentali di fault che sono frequenze che vengono generate quando la sfera all'interno del cuscinetto incontra un'anomalia o presenta essa stessa un'anomalia. Dipendono essenzialmente dalla geometria dei cuscinetti e dalla velocità relativa tra i due anelli. Sono di 4 tipi: la frequenza che riguarda l'attraversamento della

sfera sull'anello interno ($BPFI$), e quella sull'anello esterno ($BPFO$), quella di rotazione della sfera (BSF) e quella di carattere generale (FTF). Queste frequenze fondamentali di fault (in Hz) possono essere calcolate usando le seguenti equazioni[12]:

$$
\begin{aligned}
BPFI &= \frac{N}{2} \times F \times \left(1 + \frac{B}{P} \times \cos\alpha\right) \\
BPFO &= \frac{N}{2} \times F \times \left(1 - \frac{B}{P} \times \cos\alpha\right) \\
FTF &= \frac{F}{2} \times \left(1 - \frac{B}{P} \times \cos\alpha\right) \\
BSF &= \frac{P}{2B} \times F \times \left[1 - \left(\frac{B}{P} \times \cos\alpha\right)^2\right]
\end{aligned}
\tag{2.13}
$$

dove N è il numero di sfere, α l'angolo di contatto, B il diametro in mm della sfera, P il diametro in mm della circonferenza di mezzo tra i due anelli e F la frequenza dell'albero in Hz.
Si possono fare delle considerazioni ogni qual volta dall'analisi spettrale compaia un picco su queste frequenze di fault. Innanzitutto bisogna sottolineare che la presenza non indica necessariamente uno stato di degrado o di fault, e solo guardando alla storia passata e allo studio delle armoniche successive di queste frequenze si può ottenere un quadro generale maggiormente dettagliato e preciso. Ad esempio, se un picco appare alla frequenza BPFO e anche all'armonica successiva $2 \times BPFO$ è una forte indicazione che il fault è reale[13]; oppure se non appare il picco alla frequenza f_0 di fault considerata, ma appare a $2 \times f_0$, $3 \times f_0$ e $4 \times f_0$ è presumibile che l'anomalia sia presente.

2.3 Accelerometro

Dato che le vibrazioni su macchine utensili si sviluppano su alte frequenze si è deciso di descriverle attraverso il parametro dell'accelerazione, di conseguenza lo strumento più idoneo per catturare e descrivere in maniera adeguata il fenomeno è l'accelerometro.
L'accelerometro è un sensore il cui utilizzo negli ultimi anni ha subito un estensivo aumento del proprio range di applicazioni. Nel mercato vengono offerte numerose tipologie di questi strumenti, ognuna con particolarità funzionali e costruttive differenti, dagli accelerometri a ponte estensimetrico agli accelerometri laser.
Il principio di funzionamento generale si basa sul rilevamento dello spostamento di una massa quando

essa è sottoposta ad un'accelerazione. La massa è collegata ad un elemento elastico, quando si muove la massa si sposta dalla posizione di riposo in modo proporzionale all'accelerazione; il sensore poi trasforma lo spostamento in un segnale elettrico. Ad esempio un accelerometro monoassiale è composto da una massa M collegata ad una molla libera di muoversi in una sola direzione (Fig. 2.6). Attraverso il principio di equilibrio delle forze si ottiene che lo spostamento x per una costante elastica K è proporzionale alla differenza tra l'accelerazione $\overline{a}$ e la accelerazione di gravità $\overline{g}$[14].

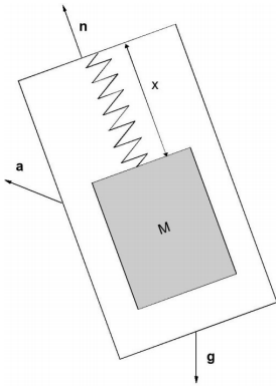


Figura 2.6: Diagramma delle forze

$$Kx = M(\overline{a} - \overline{g}) \tag{2.14}$$

L'uscita poi viene trasformata in tensione, conoscendo a priori l'offset e la sensibilità k.

2.3.1 Struttura

Considerando gli accelerometri in dotazione per lo studio delle vibrazioni ovvero **PCB Piezotronics**, il **SeTAC TK** e il **Montronix**, due sono le strutture interne che si è deciso di approfondire: l'*accelerometro piezoelettrico*, riferito al primo accelerometro, e l'*accelerometro capacitivo con struttura MEMS*, di cui sono composti i restanti.

L'**accelerometro piezoelettrico** si basa sulla proprietà, detta appunto piezoelettricità, di alcuni cristalli secondo la quale generano una differenza di potenziale quando sono soggetti a una deformazione meccanica. Applicando una pressione esterna si posizionano sul cristallo su facce opposte cariche di segno opposto. In maniera simile al condensatore, se le due facce vengono collegate a un circuito, viene generata una corrente detta corrente piezoelettrica. In questo sistema la massa

viene sospesa sul cristallo piezoelettrico; quando il gruppo è a contatto con una vibrazione la massa applica una pressione proporzionale all'accelerazione di vibrazione sul cristallo ottenendo una carica sulle facce proporzionale all'accelerazione del corpo. La costante di proporzionalità dipende sia dal materiale piezoelettrico scelto sia dalla massa sismica. Per quanto riguarda il materiale piezoelettrico usualmente viene usato un materiale di tipo ceramico, come il titanato di bario o il titanato zirconato di piombo, talvolta vengono usati anche cristalli naturali come il quarzo. I cristalli presentano una valore della costante elastica elevatissimo e questo influisce molto sull'equazione che descrive il fenomeno vibratorio.

Esistono sia accelerometri piezoelettrici monoassiali che triassiali al cui interno vengono disposti più cristalli sollecitati lungo tre direzioni mutuamente ortogonali.

Il range di misura di questo tipo di sensori è abbastanza largo: operano da una frequenza minima attorno ai $2Hz$ fino a una frequenza massima che può arrivare fino ai $10kHz$.

Uno degli aspetti da rimarcare per l'accelerometro piezoelettrico riguarda la robustezza e affidabilità nel lungo periodo: è in grado di resistere a shock e sollecitazioni abbastanza elevate senza subire danneggiamenti, per questo trova spesso impiego in applicazioni dove bisogna rilevare vibrazioni dinamiche, soprattutto in ambienti meccanici.

Altri aspetti positivi da sottolineare sono l'operatività per intervalli estesi di temperatura e un'alta resistenza ai disturbi mentre un aspetto critico di cui tener conto consiste sicuramente nella difficoltà di misurare accelerazioni *quasi statiche* con frequenze inferiori ai $2Hz$. Infatti se il cristallo subisce una compressione permanente, il segnale elettrico tende a dissiparsi e di conseguenza si rischia di ottenere una valore anomalo per l'accelerazione.

L'**accelerometro capacitivo con struttura MEMS** presenta invece caratteristiche abbastanza diverse. Il tipo capacitivo, oggi, è la tecnologia più comunemente usata per gli accelerometri.

L'elemento fondante di questa struttura è dato dalla capacità elettrica del condensatore: come metodo per rilevare lo spostamento della massa dalla sua posizione di riposo si utilizza appunto la variazione della capacità elettrica di un condensatore al variare della distanza tra le sue armature.

Con il termine MEMS si fa riferimento a un tipo di tecnologia che consiste di microstrutture meccaniche e circuiteria microelettronica, il tutto integrato all'interno dello stesso chip di silicio. La circuiteria microelettronica è responsabile della trasformazione dell'informazione, data dalle micro-

strutture meccaniche, in segnali digitali/analogici. Il sensore MEMS vero e proprio è costituito da due condensatori collegati in un half-bridge (dispositivo in grado di connettere due o più reti): un'accelerazione muove la massa sensibile (che costituisce una delle armature dei condensatori, mentre l'altra è realizzata sul supporto fisso della membrana) facendo variare la capacità dei due condensatori. La capacità di questi condensatori è dell'ordine dei pF, mentre la massima variazione di capacità si attesta attorno 10-100 fF. In realtà possono esserci più gruppi di condensatori: tutti i condensatori della parte superiore sono connessi in parallelo per una capacità complessiva C_1 e allo stesso modo tutti quelli inferiori per la capacità complessiva C_2. Ciò ci consente di considerare il modello semplificato composto da due condensatori con capacità totali C_1 e C_2.
Il rilevamento capacitivo è indipendente dal materiale di base e fa affidamento sulla variazione della capacità quando la geometria di un condensatore sta cambiando. Trascurando l'effetto vicino ai bordi, la capacità sulle armature si calcola tramite:

$$C = \epsilon_0 \epsilon_r \frac{A}{d} = \epsilon \frac{A}{d} \tag{2.15}$$

Mentre A e d sono parametri geometrici del sistema, rispettivamente l'area delle armature e la distanza tra esse, il resto sono costanti che dipendono dal materiale: ϵ_0 è la *costante dielettrica del vuoto* ($\epsilon \simeq 8.85 \times 10^{12} \frac{C^2}{N \cdot m^2}$); ϵ_r è la *costante dielettrica relativa* del materiale interposto tra le armature (sempre maggiore di uno) ; $\epsilon = \epsilon_0 \epsilon_r$ prende il nome di *costante dielettrica assoluta* del materiale e quantifica la propensione di una sostanza a contrastare l'intensità di un campo elettrico presente al suo interno.
Le capacità dei due condensatori[15] (C_1 e C_2) nell'accelerometro sono funzioni dei corrispondenti spostamenti x_1 e x_2:

$$\begin{aligned} C_1 &= \epsilon \frac{A}{x_1} = \epsilon \frac{A}{d+x} = C_0 - \Delta C \\ C_2 &= \epsilon \frac{A}{x_2} = \epsilon \frac{A}{d-x} = C_0 + \Delta C \end{aligned} \tag{2.16}$$

Quando l'accelerazione risulta nulla, non c'è spostamento e risulta $x_1 = x_2 = d$ (d è a distanza a riposo delle armature); le due capacità restano dunque uguale a C_0.

In caso di spostamento $x \neq 0$, la differenza di capacità risulta[16]:

$$C_2 - C_1 = 2\Delta C = 2\epsilon A \frac{x}{d^2 - x^2} \tag{2.17}$$

Misurando la variazione della carica ΔC è possibile trovare lo spostamento x risolvendo la seguente equazione algebrica di secondo grado:

$$\Delta C x^2 + \epsilon A x - \Delta C d^2 = 0 \tag{2.18}$$

Questa equazione può essere ulteriormente semplificata. Per piccoli spostamenti il termine Δx^2 è trascurabile e può essere omesso:

$$x \approx \frac{d^2}{\epsilon A}\Delta C = d\frac{\Delta C}{C_0} \tag{2.19}$$

Dunque si rileva che lo spostamento è circa proporzionale alla differenza di carica ΔC. Considerando poi il circuito e per l'equilibrio delle correnti tra i due condensatori risulta vero:

$$(V_x + V_0)C_1 + (V_x - V_0)C_2 = 0 \tag{2.20}$$

e utilizzando le equazioni (2.16) e (2.19) si ricava il potenziale di uscita:

$$V_x = V_0 \frac{C_2 - C_1}{C_2 + C_1} = \frac{x}{d} V_0 \tag{2.21}$$

$$x = \frac{V_x d}{V_0} \tag{2.22}$$

Il potenziale di uscita cambia in maniera proporzionale al potenziale alternato di entrata V_0.
Riprendendo la legge di Hook (per le molle) e la seconda legge di Newton per il moto (viste in precedenza) senza considerare la forza di gravità e esplicitando in funzione dell'accelerazione si ottiene:

$$a = \frac{Kx}{M} \tag{2.23}$$

da cui alla fine, sostituendo l'eq. (2.22):

$$a = \frac{Kd}{MV_0} V_x \tag{2.24}$$

Si conclude che l'accelerazione trovata risulta essere proporzionale al potenziale di uscita.
I vantaggi forniti da questi accelerometri capacitivi con struttura MEMS sono evidenti in termini di dimensioni fisica, volume, peso e costo, tuttavia la larghezza di banda è limitata a un valore inferiore ai $4kHz$ e a causa del tipo di struttura si preferisce utilizzarlo per catturare vibrazioni con valori di accelerazioni non troppo elevate. Nonostante queste limitazioni, questi dispositivi offrono una notevole linearità ed una elevata stabilità del segnale di uscita. Accelerometri di questo tipo risultano ideali soprattutto per applicazioni di monitoraggio.

2.3.2 Specifiche

Uno degli aspetti più difficili nel selezionare un accelerometro per una particolare applicazione consiste nel capire e interpretare le specifiche di questo strumento. Spesso si hanno informazioni sui requisiti dei test, ma si incontrano difficoltà nell'accoppiare questi requisiti con i modelli disponibili degli accelerometri.
Esiste quindi la necessità di una comprensiva e dettagliata descrizione delle specifiche sugli accelerometri che i produttori abitualmente utilizzano. Ciò che segue è una spiegazione sulle specifiche chiave di un accelerometro:

- *Sensibilità*: è il rapporto tra l'output elettrico del sensore e l'input meccanico. Spesso viene misurata in mV/g ed è valida per una frequenza, convenzionalmente $100Hz$. Può essere vista anche come la più piccola variazione che può avvenire nella quantità che cambia. La sensibilità fornisce un valore che viene frequentemente utilizzato per programmare un condizionatore di segnale o un sistema di acquisizione dati.

- *Range di frequenza*: rappresenta l'intervallo di frequenze che possono essere misurate dall'accelerometro. Il range di frequenza viene specificato sempre con una banda di tolleranza relativa ai $100Hz$. Tipicamente le misure vengono espresse in percentuale o in dBs. Valori comuni sono: $\pm 10\%$, ± 1 dB e ± 3 dB.

- *Range di misura*: è definito come l'intervallo di ampiezza dell'accelerazione che l'accelerometro è in grado di misurare. Questo valore dipende strettamente dalla sensibilità dello strumento.

- *Risoluzione*: va a indicare il più piccolo valore misurabile di accelerazione. Sapendo il numero di bit a disposizione e il range di misura dell'accelerometro, si riesce a trovare il valore di risoluzione.

- *Sensibilità trasversale*: definisce quanto è sensibile un accelerometro quando subisce accelerazioni a 90 gradi rispetto all'asse di sensibilità del sensore. Questo parametro è espresso in percentuale: idealmente dovrebbe essere 0% ma a causa di tolleranze di produzione la sensibilità trasversale è spesso al 5% o 10%.

- *Non linearità*: misura la deviazione dalla perfetta linearità. La linearità dell'ampiezza misura quanto sia lineare l'output dell'accelerometro lungo il range specificato. In un caso ideale l'accelerometro avrebbe la stessa sensibilità in ogni ampiezza del suo range di misura, tuttavia in un caso reale ciò non si verifica: la non linearità misurerà quanto sarà distante l'output dell'accelerometro dalla sua perfetta linearità. Un valore abbastanza comune è di 1% lungo tutto l'intervallo, tradotto significa che la sensibilità non può variare più di $\pm 1\%$ in tutto l'intervallo di misura.

- *Limite di urto*: indica il valore massimo di accelerazione che può sopportare, prima di essere danneggiato.

- *Range di temperatura*: è il range di temperatura in cui può essere utilizzato il tipo di accelerometro.

2.3.3 Tecniche di montaggio

Un elemento che influisce sulle prestazioni e sull'accuratezza di un accelerometro è sicuramente il modo in cui esso viene montato nelle vicinanze del fenomeno che si vuole acquisire. Le modalità principali del tipo di montatura possono comprendere: 1) l'utilizzo di un magnete, 2) l'utilizzo di una pellicola adesiva o 3) l'impiego di una vite.

Fissando l'accelerometro su una superficie molto liscia con una vite si ottiene la massima frequenza di risonanza ottenendo anche il più ampio intervallo di frequenza utilizzabile. Aggiungere una qualsiasi massa all'accelerometro (come un adesivo o una base di montaggio magnetica) influenza la frequenza di risonanza del sistema riducendo la larghezza di banda utilizzabile, come si può notare

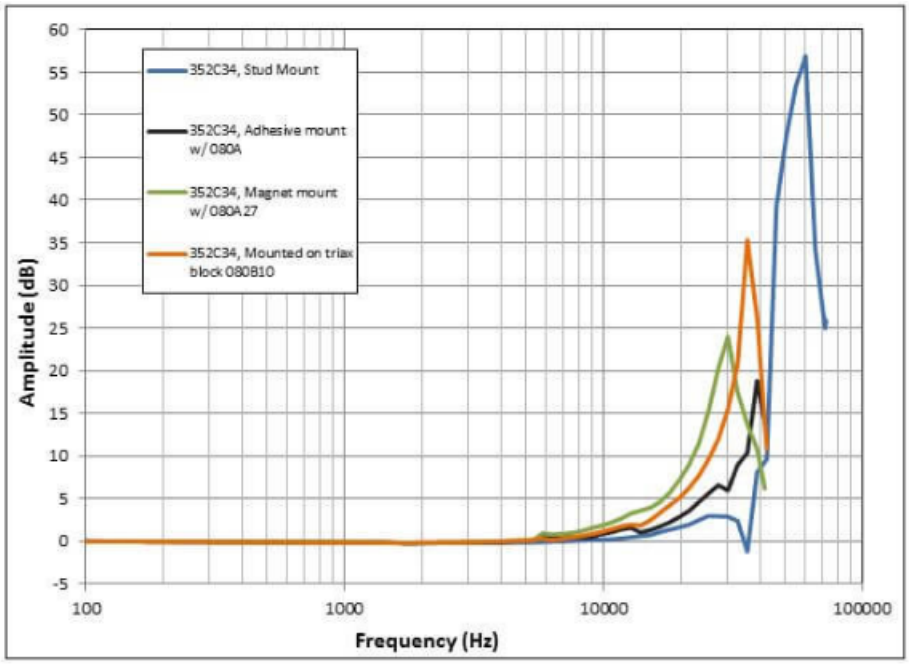


Figura 2.7: Range approssimati di frequenze per le tecniche di montaggio

dalla figura (2.7) presentata nelle specifiche dell'accelerometro PCB Piezotronics.
Nel grafico viene confrontata l'ampiezza in decibel della risposta in frequenza a seconda delle diverse tecniche di montaggio. Il range di frequenze utilizzabile corrisponde alla parte piatta della curva della risposta in frequenza e si estende fino a circa la metà della frequenza di risonanza (dove è presente il picco). Ad alte frequenze invece il segnale di accelerazione può essere mal interpretato a causa del guadagno non unitario della risposta in frequenza. Se le alte frequenze non sono richieste si può sfruttare un filtro passa-basso per rimuovere le componenti ad alta frequenza dopo una certa soglia.

Si analizzano ora le caratteristiche degli accelerometri usati per i test.

2.3.4 PCB Piezotronics

L'accelerometro *PCB Piezotronics* è un accelerometro monoassiale piezoelettrico dalle dimensioni ridotte e compatte ($41.9mm \times 18.8mm \times 21.5mm$). Ha un range di misura elevato, infatti arriva a coprire un intervallo di accelerazioni di ampiezza uguale a $\pm 50g$. Anche il range di frequenza è quello più esteso tra gli accelerometri considerati, da $0.5Hz$ fino a $8000Hz$. Il picco di risonanza si verifica attorno a $25kHz$. Viene influenzato leggermente nelle misure delle accelerazioni a 90 gradi

Figura 2.8: PCB Piezotronics

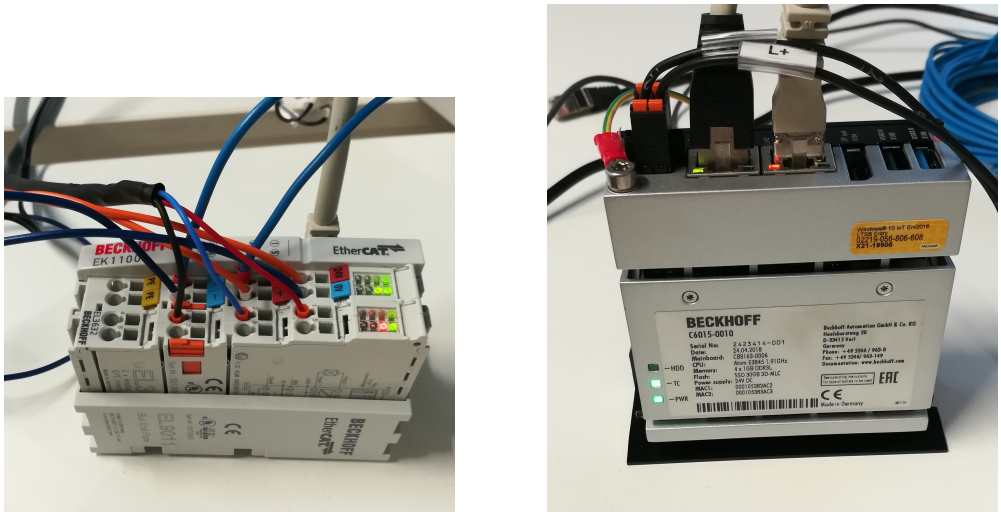
rispetto all'asse di misura anche se questo errore non va oltre il 7%. Ha una sensibilità di $100mv/g$ e può resistere ad accelerazioni fino a $5000g$ senza danneggiarsi.

I dati forniti da questo accelerometro sono stati poi acquisiti grazie a un sistema di acquisizione dati *Beckhoff*. Il segnale grezzo fornito dall'accelerometro piezoelettrico viene filtrato da **Beckhoff EL3632**, un trasduttore di misura composto da due canali indipendenti ciascuno con una fonte di corrente integrata, con filtri a parametri configurabili separatamente e con una frequenza di campionamento anch'essa parametrizzabile. Il sistema *EL3632* trasporta i dati di vibrazione di ampiezza continua al master EtherCat.

Attraverso un accoppiatore, **Beckhoff EK1100** (Fig. 2.9a), si connette il terminale , in questo caso il beckhoff EL3632, alla rete del protocollo EtherCAT (*EtherCAT device protocol*). L'accoppiatore fornisce anche ai terminali connessi la corrente necessaria per la comunicazione (può fornire fino a un massimo di $5V/2A$).

EtherCAT è una tecnologia Ethernet industriale deterministica sviluppata originariamente da Beckhoff Automation, il cui focus è rivolto a ottenere tempi di ciclo veloci e allo stesso tempo un basso costo hardware. Viene utilizzata soprattutto per applicazioni real time nell'automazione, in sistemi di test e di misura. Il principio di funzionamento generale prevede un nodo EtherCAT master che invia un pacchetto che attraversa tutti i nodi slave EtherCAT. Quest'ultimi scrivono i dati da loro prodotti nel frame mentre il frame si propaga ai nodi successivi; l'ultimo nodo invia il messaggio al master completando un ciclo.

L'accoppiatore poi, attraverso un cavo ethernet, consente la connessione con il pc industriale



(a) Beckhoff EL3632 e Beckhoff EK1100 (b) Beckhoff C6015

Figura 2.9: Componenti Beckhoff

Beckhoff C6015(Fig.2.9b). É un pc che presenta tutti gli standard che sono richiesti per un pc industriale: un esteso range di temperatura, una compatibilità con protocollo EtherCAT e un'elevata resistenza a vibrazioni e urti. Inoltre è un pc compatto, non ingombrante, senza ventola; ha un meccanismo di montaggio nella parte posteriore e tutti i connettori allo stesso livello, tra cui la possibilità di utilizzare una connessione USB. I led di status forniscono indicazioni sul funzionamento real-time.

Attraverso questo pc possono essere modificati i parametri dei filtri, delle frequenze, del campionamento, e possono essere creati i programmi che consentono di ottenere un'analisi in tempo reale del fenomeno misurato: *TwinCAT 3* è il software utilizzato per queste applicazioni e per interagire con i parametri precedentemente elencati.

2.3.5 Montronix Pulse NG

Il *Montronix PulseNG* (Fig. 2.10) è un accelerometro triassiale di tipo capacitivo con struttura MEMS. Ha sia un range di misura notevolmente inferiore all'accelerometro precedente, $\pm 6g$, sia una larghezza di banda molto meno estesa, $0 - 1350Hz$. In compenso la non-linearità è ridotta fino allo 0.1% e anche la sensibilità trasversale è ridotta fino al 2% grazie alla misurazione su tre assi. La struttura è leggermente più piccola e la forma di prisma a base rettangolare consente un posizionamento più semplice e immediato. Il sensore è in grado di rilevare collisioni in meno di $1ms$ e reagire immediatamente; gli eventi successivi a urti importanti vengono memorizzati nella scatola

Figura 2.10: Montronix PulseNG

nera. L'accelerometro viene poi connesso al sistema di acquisizione dati **Montronix IBU-NG** (Fig. 2.11). Il kit completo consente oltre a monitorare il livello di vibrazioni, di registrare diversi tipi eventi nel trasmettitore. Inoltre altre funzioni aggiuntive possono riguardare l'auto-diagnosi della macchina e il monitoraggio dei dati di taglio e l'ottimizzazione del processo.

La comunicazione tra interfaccia IBU-NG e computer avviene tramite cavo ethernet. Il software fornito come interfaccia utente per il sistema di monitoraggio PulseNG si chiama *PulseNG -hmi*. Fornisce la visualizzazione del processo per i diversi scenari e serve per controllare e impostare i singoli parametri e limiti.

Per evitare di ottenere un dato già elaborato (magari attraverso filtri che rimuovono determinate componenti) si è preferito utilizzare direttamente i pacchetti forniti dall'accelerometro creando un programma in *Visual Studio*. Il programma è stato elaborato seguendo le specifiche del pacchetto:

Figura 2.11: Montronix IBU-NG

45 byte per pacchetto, 15 per ogni limite (X,Y,Z).

2.3.6 SeTAC TK

Figura 2.12: SeTAC TK

L'accelerometro *SeTAC TK* (Fig. 2.12) presenta caratteristiche similari al *Montronix*. É un accelerometro capacitivo con struttura MEM triassiale. Ha un range di frequenza che va da 0 a $2500Hz$, mentre lo strumento supporta valori tra i $\pm 18g$; buoni sono anche i valori del range di temperatura e di resistenza agli urti.

Il sensore può registrare al suo interno e in modo permanente fino a 12000 eventi vibrazionali con indicazione del tempo, dell'ampiezza dei fenomeni e discriminandone la tipologia. Il sensore è collegato alla **SeTAC TK Intreface** (Fig. 2.13) che svolge le funzioni di programmatore del sensore e di interfaccia per la comunicazione con il computer, PLC o sistema di controllo. É dotata di diverse interfacce per comunicare con i terminali tra cui l'USB, il Modbus TCP e il Modbus RTU. Il Modbus è un protocollo di comunicazione spesso usato per connettere un computer supervisore con un'unità terminale remota in sistemi di acquisizione dati.

Anche in questo caso la casa produttrice fornisce il software da installare nel proprio computer per visualizzare l'andamento delle vibrazioni, dopo essersi collegati alla SeTAC TK Interface via ethernet.

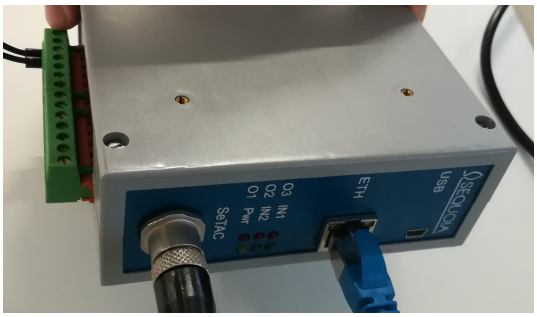


Figura 2.13: SeTAC TK Interface

2.3.7 Confronto

Nella tabella 2.1 si riporta un sommario delle principali specifiche trovate per i tre accelerometri.

	PCB Piezotronics	**SeTAC TK**	**Montronix PulseNG**
Risoluzione:	$0.01 m/s^2$	$0.0075 m/s^2$	$0.01 m/s^2$
Range di misura:	$\pm 50g$	$\pm 18g$	$\pm 6g$
Range di frequenza($\pm 3dB$):	$0.5 - 8000$ Hz	$0 - 2500$ Hz	$0 - 1350$ Hz
Non-Linearità:	$\pm 1\%$		$\pm 0.1\%$
Sensibilità Trasversale:	$< 7\%$		$< 2\%$
Range di temperatura:	-54C a $+121$C	0C a 70C	0C a 70C
Limite di urto:	$5000g$	$10000g$	$100g$
Configurazione:	Piezoelettrico	Capacitivo MEM	Capacitivo MEM
Assi:	Monoassiale	Triassiale	Triassiale
Dimensioni:	$41.9 \times 18.8 \times 21.5$ mm	$30 \times 55.5 \times 15$ mm	$30 \times 40 \times 11$ mm
Peso:	74 grammi	55 grammi	100 grammi
Corrente:	$2 - 20$ mA	200 mA	
Classe di protezione:	IP68	IP67	IP67
Campioni:	1000/2000 dati/s	25 dati/s	15 dati/s

Tabella 2.1: Confronto tra specifiche accelerometri

2.4 Set-Up Sperimentale

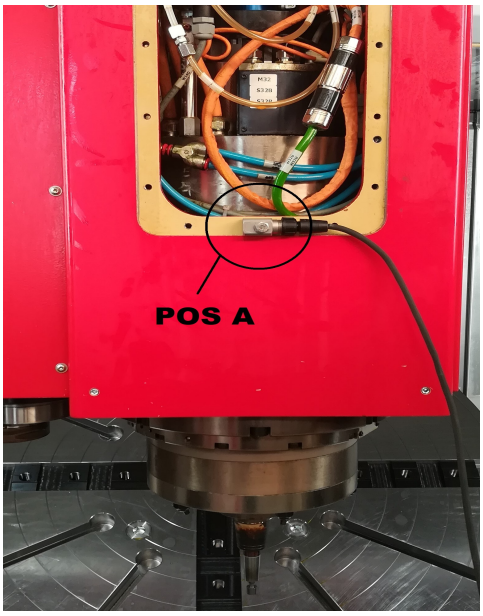
Dopo lo studio della strumentazione, è iniziata la fase della raccolta dei dati, necessari per l'approfondimento successivo.
Le diverse prove sono state svolte sulla Ultrix 1000 e gli accelerometri usati sono stati montati con l'utilizzo di viti, che permettono di avere una descrizione del fenomeno maggiormente accurata e precisa, e di non ridurre larghezza di banda disponibile.
Viste la caratteristiche simili tra gli accelerometri SeTAC e Montronix, si è deciso di condurre le prove utilizzando solo i sistemi Montronix e Beckhoff comparando quindi la tecnologia MEMS con quella piezoelettrica.

2.4.1 Test Ultrix1000: Posizione ottima

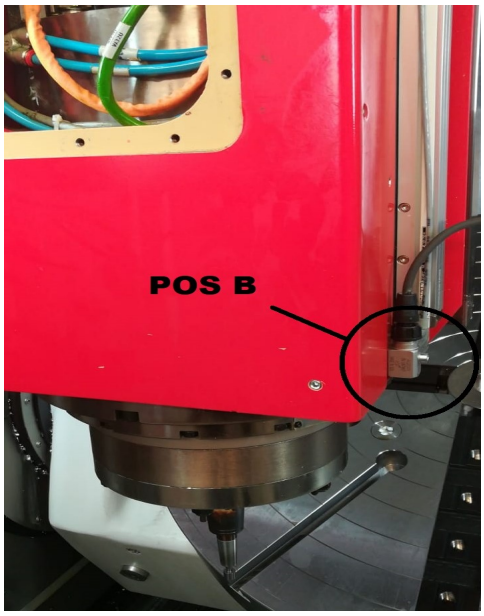
Il primo test eseguito è servito a svolgere un'analisi comparativa necessaria per proseguire nei successivi esperimenti. L'obiettivo era quello di capire quale potesse essere la posizione migliore da cui poter misurare il livello di vibrazioni del mandrino, dei cuscinetti e dell'utensile. Risulta abbastanza immediato comprendere come la scelta di una posizione rispetto ad un'altra influisca notevolmente sulla descrizione del fenomeno considerato, soprattutto quando si considerano vibrazioni di un'intensità non troppo elevata.
Non è solo la distanza fisica dalla vibrazione misurata a stabilire la qualità del dato ottenuto, bisogna anche tenere a mente sia altri fattori, come ad esempio la reiezione al rumore, sia la possibilità di subire influenze esterne non appartenenti al campo considerato (ad esempio forti vibrazioni non direttamente collegate al mandrino che influiscono sulla macchina modificando leggermente il dato).
Riassumendo i fattori che hanno inciso nella scelta della posizione ottima sono stati:

- *Distanza*: maggiormente vicini si è alla vibrazione di interesse, minore dovrebbe essere la dispersione del fenomeno.

- *Complessità del posizionamento*: si è cercato una posizione dell'accelerometro che non ostacolasse le componenti dell'elettromandrino, ma fosse facilmente accessibile.



(a) Posizione A



(b) Posizione B

Figura 2.14: Zone migliori di posizionamento dell'accelerometro

- *Cablaggio*: a volte un'estesa lunghezza del cavo può provocare un deterioramento del dato e un aumento del rumore.

- *Rigidità*: si è cercato una posizione che permettesse una posizionamento stabile e rigido e che permettesse l'utilizzo delle viti.

Basandosi su queste motivazioni, si sono individuate due zone principali di posizionamento dell'accelerometro: la **posizione A** (Fig. 2.14a), localizzata sulla parete anteriore, ottenuta dopo aver rimosso la protezione, in linea retta rispetto all'utensile a circa 30 cm dal bordo e la **posizione B** (Fig. 2.14b), situata lateralmente, nella parete di destra, ma attaccata al bordo e dunque leggermente più vicina all'utensile.

Le prove di confronto sono state svolte installando l'accelerometro PCB Piezotronics prima nella posizione A e svolgendo i test e in seguito ripetendo gli stessi esperimenti con l'accelerometro nella posizione B.

Le prove sono state svolte *"a vuoto"*, ovvero senza lavorare direttamente sul materiale, ma eseguendo degli spostamenti programmati, impostati con il computer a controllo numerico.

Per questo tipo di test è stato utilizzato un utensile da calibratura. Rispetto ad altri utensili, questo tipo di utensile è utilizzato specificatamente per calibrare i carichi sui cuscinetti e sul mandrino; in questo modo si dovrebbero limitare al minimo le vibrazioni indesiderate che colpiscono utensili

sbilanciati.

I due schemi riassumono brevemente le caratteristiche e le prove svolte nel primo test.

	Test	*Accelerometro*	*Lavorazione*	*Utensile*
TEST 1:	POS A/POS B	PCB	A vuoto	Utensile da calibratura

TEST 1:	*Prove*
	· Statico, 14000rpm
	· Quadrato 30 m/min, 14000rpm
	· Quadrato 60 m/min, 14000rpm

Tabella 2.2: Test 1

I segnali catturati, successivamente, sono stati riprodotti in *Matlab* ottenendo una rappresentazione grafica utile per eseguire un confronto visivo delle differenze delle due posizioni (osservare Fig.2.15).

Per prima cosa è stato identificato e rappresentato un tratto del segnale che presentasse caratteristiche simili su entrambe le posizioni. Ovviamente la porzione temporale considerata non è lo stessa perché le prove sono state svolte in maniera non contemporanea ma singolarmente una dopo l'altra. Questi segnali sono stati tratti dalle prove in cui l'utensile eseguiva un quadrato con una lunghezza per lato di 50 cm senza incisione sul materiale: infatti si può notare come ci siano delle parti ripide che portano a picchi improvvisi, i quali rappresentano il comportamento dell'utensile in prossimità degli angoli e parti in cui il segnale è più stabile che indicano lo scorrere dell'utensile su un lato del quadrato.

Dal confronto dei due grafici emerge chiaramente come nella posizione A il segnale sia molto più distorto rispetto alla posizione B. In particolare è dallo sviluppo del segmento centrale che si nota la maggiore differenza: nella posizione A l'accelerometro è in grado di catturare anche disturbi e rumori non necessari ai fini dello studio come si può constatare dal fatto che si ottenga una notevole escursione dei valori, da $+0.2$ a -0.2 m/s^2; al contrario la posizione B risulta più stabile, lineare e maggiormente specifica nella descrizione della vibrazione. Indicativo di questo fatto è la considerazione sull'escursione massima del tratto centrale: si va da un valore della vibrazione in esame

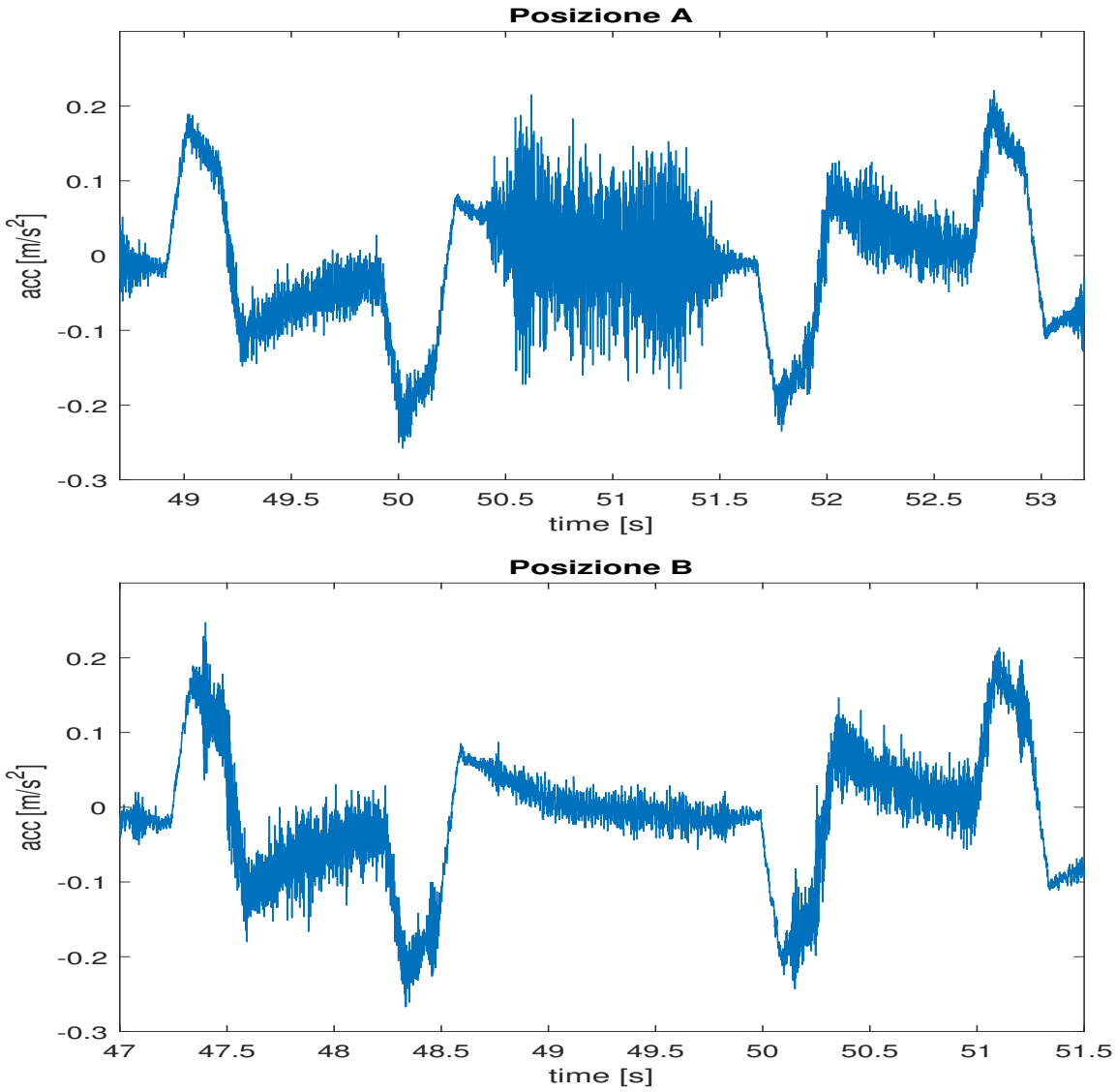


Figura 2.15: Confronto segnali posizione A/B

di circa $+0.09$ a -0.02 m/s^2. A parte la citata porzione centrale, il segnale, come si vede dal grafico, presenta caratteristiche molto similari soprattutto considerando l'andamento che è identico in entrambe le posizioni.

In conclusione si è deciso di adottare la posizione B come posizione ottimale da cui descrivere le vibrazioni. A supporto di questa decisione viene mostrato un altro grafico comparativo ottenuto facendo ruotare l'utensile a 14000 giri al minuto senza nessuna movimentazione. Anche in questo caso si può osservare che le misure e le escursioni tra valore massimo e valore minimo della posizione A sono più elevate rispetto alla posizione B; sembra che sulla posizione A si inserisca un termine di rumore additivo che si somma alla "vera" vibrazione da misurare. Una possibile spiegazione di ciò

risiede sul fatto che la posizione A è ottenuta rimuovendo la copertura anteriore e installando l'accelerometro: questo potrebbe incidere sulla stabilità e rigidità dell'intera parete, la quale non essendo più uniforme e strettamente collegata (vedi infatti tutte le viti necessaria per tenere la copertura della figura 2.14a), è soggetta a leggere vibrazioni che si sommano al complesso delle vibrazioni del mandrino e dell'utensile.

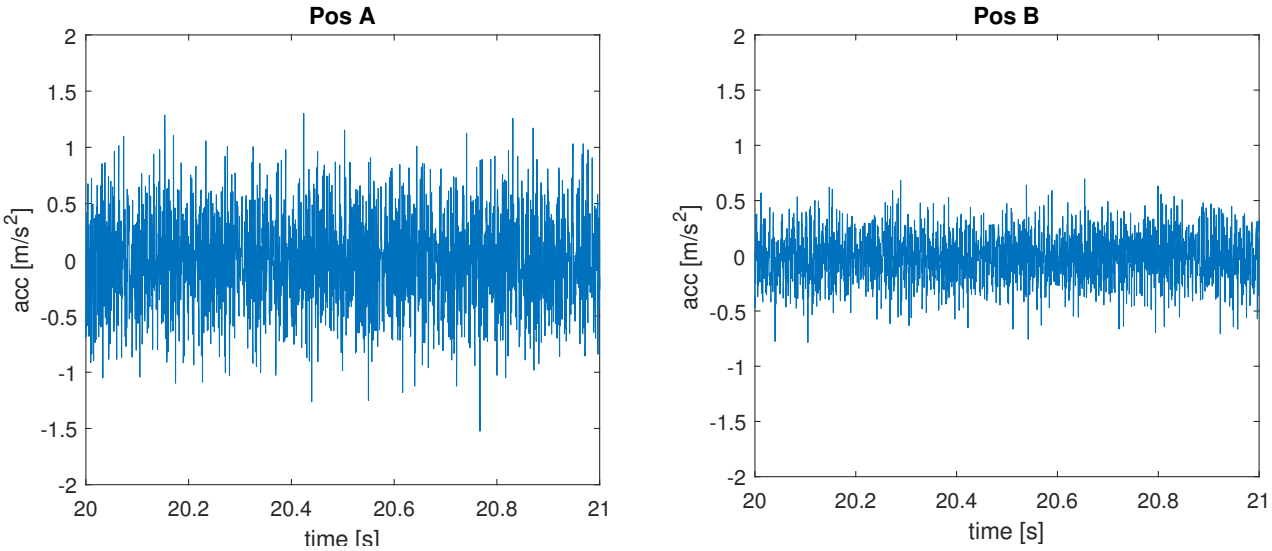


Figura 2.16: Confronto segnali posizione A/B senza movimento

2.4.2 Test Ultrix1000: Raccolta Dati

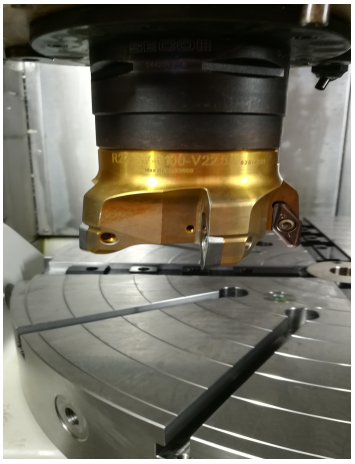
Dopo l'analisi della posizione ottimale, si sono svolte le prove di *lavorazione corretta* e *non corretta* che andassero a costituire il dataset finale. Montati gli accelerometri in prossimità della posizione B, sono stati eseguiti tre test diversi con la finalità di costruire il più possibile un'informazione varia e descrittiva: il **test 2** , svolto a vuoto con un'utensile di diametro $100\ mm$ a 5 taglienti, presenta sia segnali catturati durante il movimento della macchina utensile sia durante il solo movimento rotatorio dell'utensile; il **test 3**, svolto con lo stesso utensile, è andato a catturare il comportamento delle vibrazioni ottenute durante una lavorazione di un pezzo di alluminio sottoposto all'asportazione di materiale; il **test 4** ricalca lo stesso tipo di prove descritte nel test precedente con un utensile più leggero, più piccolo (diametro $60\ mm$) e con un numero minore di taglienti.

Dalla Figura 2.17 si può osservare le caratteristiche differenti dei tipi di utensili utilizzati. Si nota principalmente la differenza di struttura tra l'utensile usato solitamente per la calibrazione e

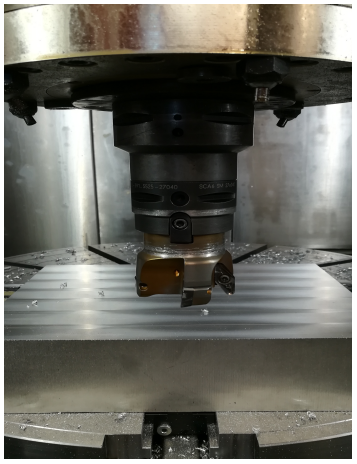
	Test	*Accelerometro*	*Lavorazione*	*Utensile*
TEST 2:	Corr/Non corr	PCB/MTX	A vuoto	Ut. diametro $100mm$ a 5 taglienti
TEST 3:	Corr/Non corr	PCB/MTX	Asportazione	Ut. diametro $100mm$ a 5 taglienti
TEST 4:	Corr/Non corr	PCB/MTX	Asportazione	Ut. diametro $60mm$ a 4 taglienti

Tabella 2.3: Riassunto Test

(a) Utensile da calibratura



(b) Ut. diametro $100mm$ a 5 taglienti



(c) Ut. diametro $60mm$ a 4 taglienti

Figura 2.17: Utensili per i Test

gli utensili usati per le lavorazioni.

Per quanto riguarda le misurazioni si è considerato un lasso temporale che va dai 40 secondi al minuto e si è cercato il più possibile di far partire la registrazione dei segnali contemporaneamente per entrambi gli accelerometri.

La difficoltà principale riguardo questi test è stata quella di riuscire a ottenere delle prove che indicassero una lavorazione impropria della macchina utensile.

Diversamente da prove che tentano di determinare il malfunzionamento di un componente come ad esempio l'utilizzo di un mandrino con cuscinetti usurati o danneggiati, o di un utensile sbilanciato, le quali presentano elevati costi di implementazione di un settaggio del genere e di distruzione del materiale, si è cercato di ottenere prove che simulassero una lavorazione impropria attraverso una maniera originale. L'idea è stata quella di simulare la lavorazione non corretta attraverso due approcci differenti: il primo che simulasse l'errore e quindi la creazioni di vibrazioni casuali attraverso la rimozione di una placca sull'utensile (nella tabella 2.4 viene indicato con la sigla *SP1*) o per un errore più deciso due placche (*SP2*), il secondo che andasse ad agire direttamente sull'utensile

causando una leggera ammaccatura (*D*) o un danno più intenso (*D2*).

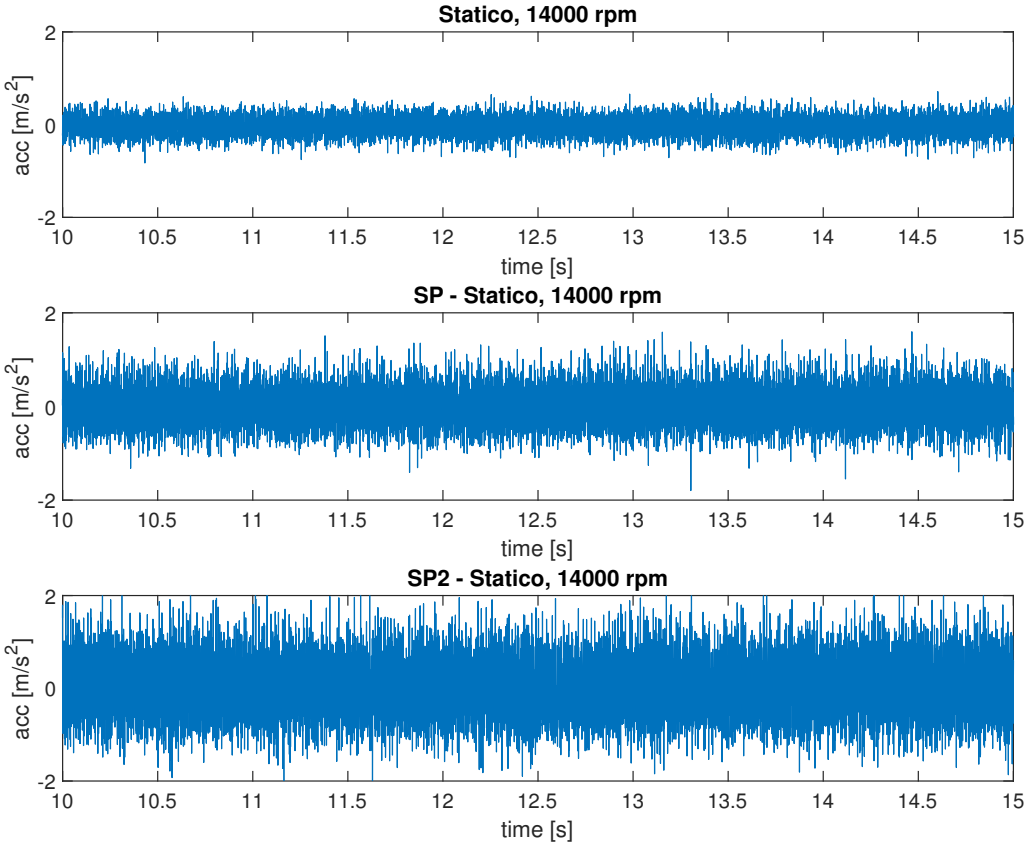


Figura 2.18: Confronto dell'effetto della rimozione delle placche

A tal proposito la Figura 2.18 fornisce una rappresentazione della variazione dei segnali quando viene rimossa una placca. Il caso in esame si riferisce alla prova senza movimentazione in cui l'utensile ruota a $14000\ rpm$. Si osserva immediatamente come il valore quadratico medio della vibrazione aumenti ogni qual volta si rimuova un tagliente, arrivando a picchi molto elevati specialmente nel caso *SP2*. Soprattutto il segnale considerato risulta utile per mostrare l'incidenza che ha la condizione dell'utensile sulla misura della vibrazione generale e come questo sia un buon approccio per simulare le lavorazioni non appropriate.

É importante sottolineare come in questo caso il fenomeno risulti ingigantito: infatti date le particolari condizioni di misura l'accelerometro è in grado di catturare in maniera ottimale la vibrazione dell'utensile, mentre in lavorazioni con movimenti della macchina utensile (sia a vuoto che lavorazioni

di asportazione) la vibrazione dell'utensile viene mitigata e il fenomeno risulta molto più contenuto non consentendo un immediato riconoscimento del tipo di lavorazione ottenuta.

TEST 2:	*Prove corrette*	*Prove non corrette*	
	· Statico, 5000rpm	· *SP* - Statico, 5000rpm	· *SP2* - Statico, 5000rpm
	· Statico, 10000rpm	· *SP* - Statico, 10000rpm	· *SP2* - Statico, 10000rpm
	· Statico, 14000rpm	· *SP* - Statico, 14000rpm	· *SP2* - Statico, 14000rpm
	· Quadrato 30 m/min	· *SP* - Quadrato 30 m/min	· *SP2* - Quadrato 30 m/min
	· Quadrato 60 m/min,	· *SP* - Quadrato 60 m/min	· *SP2* - Quadrato 60 m/min

TEST 3:	*Prove corrette*	*Prove non corrette*
	· Asportazione 0.5mm	· *SP* - Asportazione 0.5mm
	· Asportazione 1mm	· *SP* - Asportazione 1mm

TEST 4:	*Prove corrette*	*Prove non corrette*	
	· Asportazione 0.5mm	· *D* - Asportazione 0.5mm	· *D2* - Asportazione 0.5mm

Tabella 2.4: Test 2 / Test 3 / Test 4

Capitolo 3

Creazione del Dataset

Dopo la fase dei test, si è svolta la fase di *preprocessing* in cui i segnali grezzi sono stati elaborati in modo tale da avere come risultato finale una collezione di dati che fosse il più possibile uniforme e descrittiva del fenomeno.
Per ottenere il dataset finale dal segnale originario, l'approccio seguito prevede in maniera sequenziale tre step:

1. **Allineamento**
2. **Finestratura**
3. **Estrazione delle features**

Si sottolinea che in questo caso si è preferito svolgere questo tipo di metodologia, tuttavia altre analisi e altri algoritmi di machine learning, in particolare gli approcci di deep learning[17], prevedono direttamente l'utilizzo del segnale originario per costituire l'intero dataset senza subire una precedente elaborazione. Ci sono pregi e difetti nella scelta di una o dell'altra strada, in particolare l'approccio standard consente di ottenere ottimi risultati se le features che descrivono il fenomeno riescono a cogliere distintamente le caratteristiche del determinato segnale.

Il primo passo dunque, riguarda l'*allineamento* dei segnali. Le misurazioni dei due accelerometri per quanto si è cercato di farle partire in maniera contemporanea, presentano sempre una differenza temporale da tenere conto. Per questo motivo sarebbe efficace implementare un comportamento iniziale standard della macchina utensile che possa essere interpretato come uno start delle misure

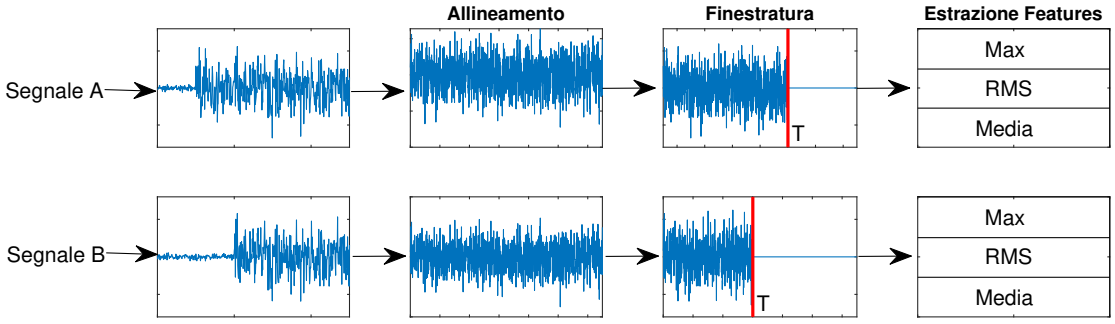


Figura 3.1: Procedura di estrazione delle features

degli accelerometri. Soprattutto per un'applicazione futura real time, questo movimento sarebbe il punto comune da cui poter allineare le misure dei due accelerometri e anche di Sentinel. Un esempio di possibile comportamento iniziale potrebbe essere quello di uno spostamento frontale di 10 centimetri a bassa velocità.

Nel nostro settaggio si è evitato questo problema attuando l'allineamento dei segnali *"a mano"*: per ogni test è stata stabilita una zona comune in cui lo sviluppo del segnale fosse più o meno simile, nella maggior parte dei casi considerando come punto di inizio una rapida salita (ad esempio nel caso dei movimenti per formare un quadrato), e da lì allineando le misure degli accelerometri tenendo conto dei diversi campionamenti.

Nel secondo step è stata adoperata la tecnica della *finestratura* per ottenere porzioni del segnale di lunghezza identica. Una funzione finestra nell'elaborazione numerica dei segnali è una funzione che vale zero al di fuori di un certo intervallo. Esistono vari tipi di funzioni finestra che possono essere usate a seconda della circostanza, in cui N rappresenta il numero dei campioni di una finestra a tempo discreto, e n un numero intero che assume valori tra $0 \leq n \leq N-1$[11][18]:

- *Finestra Rettangolare*:

$$w(n) = 1$$

- *Finestra di Hann*:

$$w(n) = 0.5\left(1 - \cos\left(2\pi \frac{n}{N-1}\right)\right)$$

- *Finestra di Gauss*:

$$w(n) = e^{-\frac{1}{2}\left(\frac{n-N/2}{\sigma N/2}\right)^2} \qquad \sigma \leq 0.5$$

- *Finestra di Blackman*:

$$w(n) = \frac{1-\alpha}{2} - 0.5\cos\left(\frac{2\pi n}{N}\right) + \frac{\alpha}{2}\cos\left(\frac{4\pi n}{N}\right) \qquad \alpha = 0.16$$

Si è deciso di adoperare la finestra rettangolare perché è l'unica che non modifica il segnale di partenza nell'intervallo da essa considerato. L'utilizzo di questo tipo di finestra consente di preservare meglio le caratteristiche del segnale originario ed è quindi più robusta nel descrivere il fenomeno osservato e nel riportare il segnale grezzo, tuttavia bisogna presentare attenzione al fatto che talvolta potrebbero presentarsi dati in larga parte affetti da rumore e disturbi esterni che questa finestra non riesce ad attenuare o eliminare.
Dal punto di partenza identificato con l'allineamento, veniva quindi applicata una finestra con una larghezza di 1 secondo per entrambi gli accelerometri ponendo a zero tutto il resto dei campioni. Con un intervallo temporale di un secondo si è ottenuto un numero di campioni diversi a seconda dell'accelerometro considerato: 2000 samples per il PCB piezotronics, 15 per il Montronix.
Un altro parametro di cui tener conto, oltre alla larghezza della finestra, è la sovrapposizione che ci permette di stabilire a quale distanza partirà la successiva finestra. Ad esempio una finestra con sovrapposizione nulla (0%) indica che la finestra successiva a quella di partenza, la quale si suppone essere compresa tra 0 e N, sarà applicata immediatamente dopo la fine della prima finestra, quindi tra N e $2N$, mentre una finestra con sovrapposizione al 50% indicherà che la finestratura successiva sarà applicata tra $N/2$ e $3N/2$.
É stato scelto il parametro di sovrapposizione al 50% in quanto in questo modo si riesce ad avere una descrizione del segnale di partenza più lineare e omogenea senza troppi sbalzi eccessivi e un dataset composto da un numero maggiore di dati.

Infine con il segnale finestrato si è passati all'*estrazione delle features*. Le features sono caratteristiche del segnale che dovrebbero fornire una panoramica generale su di esso in ogni specifico slot temporale, permettendoci di creare il dataset.

Come si è accennato nella sezione 2.2 esistono alcuni parametri che ci permettono di descrivere in maniera completa una vibrazione; affidandoci a quelle features e aggiungendone di nuove, si è elaborata la lista di descrittori:

Lista 1: *Media, Deviazione Standard, RMS (Valore Quadratico Medio), Valore Massimo, Skewness, Kurtosis, Picco-Picco, Fattore di Cresta, Fattore di Forma, Fattore Impulsivo, Fattore di Margine, Energia*

Mentre la media e la deviazione standard sono indici noti, il fattore impulsivo e il fattore di margine contribuiscono a una descrizione più approfondita. Il primo è definito come il rapporto tra il valore massimo e la media del segnale mentre il secondo si differenzia perché al denominatore presenta il termine al quadrato:

- *fattore impulsivo*:

$$F_i = \frac{max(x)}{\frac{1}{N}\sum_{i=1}^{N}|x_i|} \tag{3.1}$$

- *fattore di margine*:

$$F_m = \frac{max(x)}{\left(\frac{1}{N}\sum_{i=1}^{N}|x_i|\right)^2} \tag{3.2}$$

L'energia invece è data dalla somma dei campioni al quadrato:

- *Energia*:

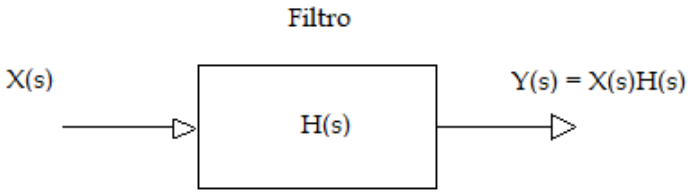
$$E = \sum_{i=1}^{N} x_i^2 \tag{3.3}$$

Il segnale, come si è visto, può essere studiato non solo nel dominio del tempo ma anche nel dominio della frequenza. Convertito il segnale finestrato tramite la trasformata di Fourier è stata utilizzata la stessa lista di descrittori per analizzare il segnale in frequenza. Questa analisi permette di ottenere uno studio completo sullo spettro di frequenze e sulle armoniche più influenti.

Lista 2: *fft_Media, fft_RMS, fft_Max, fft_Skewness, fft_Kurtosis, fft_Picco-Picco, fft_Fattore di Cresta, fft_Fattore di Forma, fft_Fattore Impulsivo, fft_Fattore di Margine, fft_Energia*

Sviluppare un dataset ricco di features descrittive è fondamentale per l'analisi successiva. Per questo motivo si è deciso di prendere in considerazione l'utilizzo di un filtro sul segnale finestrato per avere un'ulteriore serie di descrittori. Il filtraggio consente di modificare leggermente lo spettro di un

segnale mediante la funzione di trasferimento del filtro.



Il più semplice filtro digitale con risposta all'impulso finita che sia non ricorsivo sostituisce ogni campione x_i del segnale finestrato con una combinazione lineare y_i di un sottoinsieme di campioni vicini:

$$y_i = \sum_{k=n_L}^{n_R} a_n x_{i+n}$$

Ad esempio considerando il filtro *a media mobile*, fissati il numero di campioni di sinistra n_L e il numero di campioni di destra n_R, esso calcola ogni campione y_i come l'effettiva media dei campioni tra x_{i-n_L} e x_{i+n_R}. Ciò avviene se si pongono i coefficienti a_i come $a_i = \frac{1}{n_L+n_R+1}$. Si ricorda anche che un filtro si dice causale se risulta che non vengono considerati campioni successivi a quello in esame, ovvero $n_R = 0$.

Il filtro che è stato adoperato, invece, è noto come *filtro di Savitzky-Golay*[19]; questo è un filtro digitale che applicato al segnale cerca di creare un comportamento più lineare e appiattito di esso, aumentando la precisione senza distorcere la tendenza generale. Ciò si ottiene, attraverso la convoluzione, adattando il comportamento di un sottoinsieme di punti adiacenti con un polinomio di basso grado grazie al metodo dei minimi quadrati. In pratica, diversamente da un filtro a media mobile che applica una finestratura rettangolare in cui la pesatura dei suoi coefficienti è costante, nel filtro Savitzky-Golay si applica una finestra di tipo polinomiale eseguendo per ogni campioni x_i un fitting polinomiale ai minimi quadrati dei $n_L + n_R + 1$ campioni della finestra scorrevole, prendendo come elemento y_i l'elemento in posizione i-esima. Da un punto di vista computazionale si nota che la procedura di fitting richiede solo l'inversione di una matrice.

I filtri di Savitky-Golay sono più efficienti nel preservare le componenti ad alta frequenza rispetto ai filtri standard che mediano i segnali e per il nostro caso, questo filtro è stato implementato nell'ambiente *Matlab* tramite la funzione *sgolayfilt(ordine)* con ordine del polinomio che approssima pari a

3.

Lista 3: *filt_Media, filt_STD, filt_RMS, filt_Max, filt_Skewness, filt_Kurtosis, filt_Picco-Picco, filt_Fattore di Cresta, filt_Fattore di Forma, filt_Fattore Impulsivo, filt_Fattore di Margine, filt_Energia*

Come si accennava nell'introduzione, un'ulteriore fonte di informazione sono i dati forniti da Sentinel che sono in grado di riportare un'accurata descrizione sull'utilizzo del mandrino e in generale della macchina utensile. Per il nostro specifico caso sono state considerate le informazioni sulla potenza sviluppata dal mandrino e sulla coppia, inserendole nello stesso slot temporale considerato dai due accelerometri.

Lista 4: *Potenza Media, Potenza Massima, Potenza RMS, Coppia Media, Coppia Massima, Coppia RMS*

L'insieme delle prime tre liste per entrambi gli accelerometri unito alla quarta lista ottenuta da Sentinel è andato a comporre il dataset finale.

Dai test svolti con questo procedimento si ottiene che il dataset completo è composto da 1401 campioni e 76 features.

	Campioni	Features
DATASET 1:	1401	76

Capitolo 4

Fault Detection: Approccio Supervisionato

Raccolti i dati necessari, si è passati alla fase finale del progetto, quella di analisi dei dati attraverso l'applicazione di algoritmi di machine learning.

Dalla letteratura correlata a questo genere di esperimenti non si è trovato un algoritmo principale che descrivesse in maniera accurata e specifica il caso in esame, infatti spesso e volentieri non è possibile conoscere a priori la soluzione che fornisca il miglior risultato. Per questo, dato che ogni algoritmo presenta un proprio carattere distintivo e un proprio stile, si è cercato di variare la scelta dei plausibili metodi applicabili, cercando di indicare il percorso migliore da cui intraprendere un'analisi successiva maggiormente sistematica.

L'obiettivo dunque è quello di trovare algoritmi che siano in grado di apprendere in modo autonomo dai set di dati che si ottengono. É questa la novità che caratterizza il machine learning: non è più l'uomo a dover trovare nei dati pattern fondamentali che ci consentano di definire l'appartenenza a classi differenti, ma sono gli algoritmi che vagliando tra le strutture dei dati trovano e imparano a riconoscere le caratteristiche sostanziali che dividono al meglio queste classi. Perciò appunto si parla di machine learning, come detto, chiamato anche *"apprendimento automatico"* indicando tutta quella serie di meccanismi che permettono a un algoritmo o una macchina intelligente di migliorare le propria abilità di risolvere un compito e dunque di ottenere performance migliori nel tempo.

Il machine learning non è una materia indipendente e a sè stante, ma è strettamente collegata a rami dell'informatica, della statistica e dell'ottimizzazione di sistemi. La combinazione di questi

settori diversi è molto comune e fondamentale per potere realizzare soluzioni di problemi complessi. Una delle branche di maggiore interesse e successo è quella del *data mining* con cui si cerca di individuare informazioni e pattern estrapolandoli da banche di dati. La ricerca è finalizzata a scovare associazioni, anomalie e schemi ricorrenti in modo tale che da informazioni criptiche e nascoste si riesca ottenere una conoscenza sfruttabile per vari scopi.

In generale il machine learning viene diviso in tre categorie principali a seconda della natura del comportamento dell'apprendimento automatico. Ogni categoria presenta le sue particolari caratteristiche e i suoi algoritmi. I tre diversi tipi di approcci sono: l'apprendimento supervisionato, l'apprendimento non supervisionato e l'apprendimento per rinforzo:

1. **Apprendimento Supervisionato**: è un tipo di algoritmo di apprendimento in cui si inferisce una funzione da un set di dati classificati che consistono in un set di dati di esempi. Questi esempi consistono tipicamente di un vettore di dati di ingresso (l'oggetto di input) con il corrispettivo output e servono sostanzialmente a istruire un algoritmo per renderlo successivamente autonomo nell'elaborazione di dati non ancora classificati. Per questo motivo viene appunto chiamato "supervisionato", in quanto vengono forniti precedentemente degli esempi etichettati alla macchina.

   Il classico esempio di apprendimento supervisionato riguarda la classificazione di email giudicate come spam o no spam; prima si fornisce una serie di esempi per imparare a riconoscere la differenza e in seguito si ottiene il modello che dovrebbe riconoscere autonomamente quando un'email ricevuta è uno spam o no.

   Ci sono sostanzialmente due famiglie nell'apprendimento supervisionato: la *classificazione* e la *regressione*. Nella classificazione si vuole decidere se un dato appartenga a una determinata classe o meno (l'esempio precedente della pubblicità nella posta elettronica è un caso di classificazione) mentre con la regressione si cerca di predire il valore numerico di un dato (ad esempio tentare di predire la quotazione futura di un azione).

2. **Apprendimento Non-Supervisionato**: è un algoritmo di apprendimento che a differenza del precedente non utilizza dati classificati e etichettati in precedenza, non si conosce a quale classe essi appartengano. Per questo l'obiettivo cambia radicalmente: si cerca di estrarre una

regola che raggruppi i casi presentati secondo caratteristiche che si ricavano dai dati stessi o una relazione per capire come essi siano collegati. Una delle applicazioni principali, infatti, è il clustering, ovvero il raggruppamento dei dati in gruppi omogenei definiti cluster.
Essendo assente una classificazione non etichettata a monte, questa metodologia è molto più problematica per determinare l'affidabilità del risultato.
L'apprendimento non supervisionato, quindi, serve generalmente ad estrarre informazioni non ancora note. Per questo motivo è un metodo spesso utilizzato nel correlare diversi dati ed estrarre informazioni non note e nello scoprire anomalie in un dataset.

3. **Apprendimento Per Rinforzo**: è un algoritmo che ha un approccio completamente differente. L'algoritmo parte compiendo un'azione in risposta a dei dati, successivamente, ricevuto un segnale di ricompensa (rinforzo) che testimonia il livello di correttezza della decisione presa, l'algoritmo modifica la propria strategia in base a questo segnale per ottenere la ricompensa maggiore.
É un approccio molto comune sia nel applicazioni di IoT (Internet of Things) sia nella robotica, in cui da letture dei sensori di un robot in un certo momento si deve scegliere l'azione successiva da compiere.

Si può considerare anche un quarto tipo di approccio che si insinua esattamente a metà tra l'approccio supervisionato e quello non supervisionato: l'**apprendimento semi-supervisionato** (anche detto parzialmente supervisionato). Essenzialmente si basa su dati misti, costituiti da una parte di dati etichettati e da una senza etichettatura. Spesso è un approccio che fa da complemento a quello non supervisionato, cercando di migliorarne le previsioni.

Nonostante gli innegabili pregi e la potenza del machine learning, per questo tipo di tecniche è necessario considerare anche alcuni aspetti delicati, prima fra tutti la quantità di dati. Un abbondante numero di dati nella maggioranza dei casi è fondamentale per ottenere un algoritmo che sia realmente utile e applicabile in un caso reale; dataset composti da un numero abbastanza contenuto di misure potrebbero inficiare la qualità del risultato finale ottenendo risultati fuorvianti, per questo motivo si sfruttano tecniche attraverso le quali si riesce a ottenere un risultato maggiormente robusto.

Un altro aspetto da sottolineare riguarda la generalizzazione degli algoritmi a dati che non sono ancora stati osservati. La procedura generale prevede la divisione del dataset in un due macro-categorie: il *training set* e il *test set*. Il primo è costituito da un set di dati utilizzati per addestrare l'algoritmo mentre il secondo utilizza i dati per verificarne la correttezza e le performance. La separazione dei dati in set di training e set di testing rappresenta una parte importante della valutazione dei modelli di machine learning. In genere, quando si separa un set di dati in un set di training e un set di testing, la maggior parte dei dati viene utilizzata per il training e una quantità più esigua per il testing: in proporzione di solito la divisione del dataset assicura il 70% di dati al set di training e il 30% al set di testing. Ovviamente i dati di "allenamento" e i dati di set devono essere il più possibile simili e descrittivi dello stesso fenomeno altrimenti si rischiano di avere livelli di accuratezza molto bassi. A questo proposito i dati sono divisi in maniera casuale per evitare discrepanze nei dati stessi e valutare al meglio le caratteristiche del modello.
Dopo aver elaborato il modello tramite il set di training, esso viene testato eseguendo stime sul set di test. L'errore di generalizzazione nel test set stabilisce l'accuratezza del modello sul sistema considerato.

Tuttavia questa procedura di divisione casuale del dataset risulta poco generalizzante e potrebbe non considerare pattern e schemi nei dati che finiscono nel test set, scontrandosi inevitabilmente con il problema dell'*overfitting*. L'overfitting è sicuramente una delle maggiori criticità nell'ambito del machine learning. In pratica con il termine overfitting si indica il sovradattamento dei dati durante il processo di apprendimento: il modello è soggetto a overfitting dei dati di addestramento quando si nota un funzionamento ottimale con i dati di addestramento, ma non con i dati di testing. Ciò accade perché il modello ha memorizzato i dati di training precedenti e non è in grado di generalizzare agli esempi dei dati di test che non ha visto. Se valutando l'accuratezza dell'algoritmo per entrambi i set di dati si trova che le performance sul set di training sono di gran lunga migliori di quelle del test set allora molto probabilmente si è in presenza di un caso di overfitting (ad esempio sarebbe indicativo il caso in cui ci sia presente un'accuratezza superiore al 95% nel training set e un'accuratezza attorno al 50% nel test set).
Il problema opposto è quello dell'*underfitting*. Il modello è soggetto a underfitting dei dati se ha scarse prestazione sui dati di training; risulta essere troppo semplice e di conseguenza non riesce a

essere flessibile nell'apprendimento dai dati.

Come si può riscontrare dalla Figura 4.1 sia il modello soggetto a overfitting sia quello soggetto

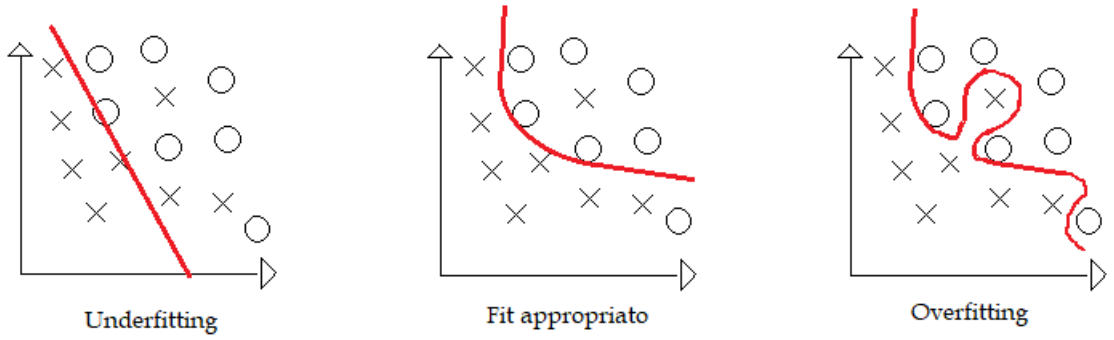


Figura 4.1: Differenti adattamenti ai dati

a underfitting risultano inadatti a descrivere la struttura dei dati: il primo è troppo complesso e non riesce a cogliere il comportamento generale, il secondo troppo semplice e inadatto a cogliere la separazione.

Una delle tecniche usate per affrontare il problema dei dati sottoposti a overfitting è la *cross-validation*[20]. Per un data scientist la cross validazione è probabilmente una delle tecniche più importanti in quanto consente di convalidare la stabilità del modello di apprendimento automatico generalizzando ai nuovi dati e assicurandosi che il modello presenti la maggior parte dei dati e dei pattern corretti.

In particolare è definita come una tecnica di validazione del modello per stimare quanto i risultati di un'analisi generalizzeranno a dati di testing indipendenti.

L'idea di base del funzionamento di questa tecnica è semplice: si generano multiple divisioni del training e del test set, usando queste divisioni per adattare il modello.

Ci sono due principali tipologie di cross-validazione. La prima è la k-*fold cross validation*, che consiste inizialmente nel dividere il dataset in k cartelle e poi nell'utilizzarne $k-1$ per il training dell'algoritmo e la restante per il testing dei dati. Questa procedura viene svolta appunto per k iterazioni, ogni volta considerando un cartella di test diversa; in questo modo l'accuratezza finale dipende dalla media delle accuratezze dei singoli round.

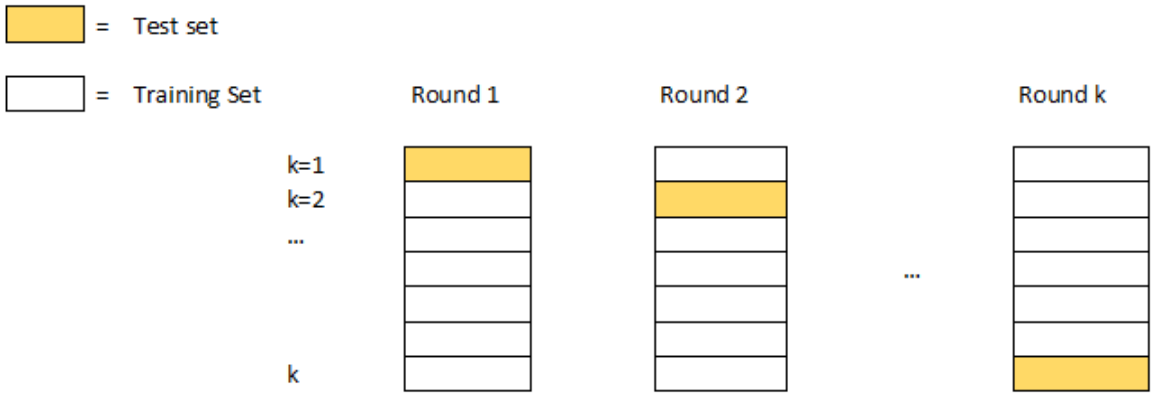


Figura 4.2: k-fold cross validation

Questa è un'ottima tecnica quando si ha un numero sufficiente di dati e solitamente come regola generale si utilizza un valore per k corrispondente a 5 o a 10.

Il secondo tipo, invece, viene chiamato *leave-one-out cross validation*. In questa tecnica il dataset di training è composto da tutte le misure eccetto una che viene usata per il test. Le cartelle del caso precedente sarebbero dunque composte da solo una misura. Questo processo ci consente di eseguire l'intera procedura di training più testing un numero di volte pari al numero di esempi nel dataset perciò si rischia di ottenere un processo molto oneroso da un punto di vista computazionale. Siccome si utilizza quasi l'intero dataset per il training in ogni round, il modello sarà molto simile al modello reale; tuttavia si introduce una notevole varianza nel testing, dato che si considera solo una misura per volta.

Similare in molti aspetti alla k-fold cross-validation è la *stratified k-fold*. Nella stratificazione si cerca di assicurare che ogni cartella sia rappresentativa di tutti i dati: ad esempio supponendo che un dataset A sia composto dal 60% di prove del tipo A, dal 20% di prove di tipo B e dal 20% di prove di tipo C, quando si applica la *stratified k-fold* ogni cartella sarà costituita da un numero di prove che rappresenta le percentuali di partenza (si intuisce la diversità rispetto a una divisione dei dati in maniera casuale nelle diverse cartelle). É una tecnica che risulta utile soprattutto su dataset contenenti poche prove o dataset sbilanciati ed aiuta a rendere la validazione più stabile.

Applicando queste considerazioni di carattere generale al progetto si è deciso di affrontare il problema seguendo due strade parallele: una basata sull'approccio supervisionato e una basata su quello

non supervisionato e convalidando i risultati ottenuti attraverso la tecnica della cross-validation.

Il primo approccio considerato è stato quello supervisionato; avendo a disposizione un dataset che presenta per ogni dato un'etichettatura (lavorazione corretta e lavorazione non corretta), l'obiettivo fissato è stato quello di cercare di trovare un algoritmo che riuscisse in maniera accurata a riconoscere quando si trattasse di una prova propria o impropria.
É possibile appunto utilizzare questo tipo di approccio proprio perché si è a conoscenza della provenienza del dato (a quale categoria esso appartiene) e grazie a ciò risulta semplice e immediato valutare la correttezza e l'accuratezza dell'algoritmo.
Aumentando gradualmente i dati forniti per l'allenamento, i quali dovrebbero essere via via più descrittivi e caratterizzanti dei tipi di lavorazioni non corrette, si dovrebbe riuscire a stabilire un algoritmo finale da implementare nelle macchine che riesca a riconoscere la lavorazione svolta impropriamente e avvertire prontamente l'operatore che sta utilizzando la macchina.

4.1 Riduzione del numero di features

Un aspetto che fino ad ora non è stato considerato nel quadro generale è la relazione tra il numero di dati e il numero di features. Questo confronto risulta spesso determinante per evitare successivi problemi nell'applicazione degli algoritmi e aumentare la bontà del modello.
Un procedimento abbastanza comune prevede che quando si costruisce un dataset si cerchi di utilizzare il maggior numero possibile di descrittori per avere una descrizione del fenomeno a 360 gradi. Tuttavia questo è un percorso abbastanza rischioso soprattutto nel caso in cui il rapporto tra il numero di dati raccolti e il numero di features considerate non fosse elevato perché potrebbe portare l'algoritmo di apprendimento ad avere overfitting dei dati. Il problema è sempre lo stesso, il modello diventa troppo complesso e si adatta solamente ai dati di training perdendo il carattere generale. Questa situazione è nota anche come "Curse of dimensionality" (*Maledizione della dimensionalità*), dove la dimensionalità semplicemente si riferisce al numero di features all'interno di un dataset: a causa di ciò certi algoritmi faticano ad istruire modelli efficienti, specialmente nei casi di algoritmi di clustering (è sicuramente più difficile cercare relazioni in uno spazio di dimensioni elevata rispetto a uno a poche dimensioni).

Più precisamente, data la grande quantità di features, il modello potrebbe cercare di adattarsi ad attributi che risultano poco caratteristici per il comportamento generale trascurando o portando in secondo piano gli attribuiti rilevanti nel processo di machine learning. Il risultato sarebbe un algoritmo essenzialmente inutile che non coglie il comportamento generale che c'è dietro l'esperimento. Si può affermare allora che all'aumentare del numero di features, aumenti la possibilità di incontrare un attributo irrilevante che indichi un pattern differente e contrario a quello cercato e, di conseguenza, un aumento esponenziale di avere overfitting dei dati.
D'altra parte utilizzare un numero di features ridotto non è una soluzione perché non si riesce a cogliere la particolarità del dataset originale creando un algoritmo che ha una visione parziale e soggettiva del fenomeno; bisognerebbe cercare di trovare il numero ideale di features per descrivere il dato, un numero che dipenderà ovviamente dalla particolare situazione in cui ci si trova.

Un metodo che consente di raggiungere un punto di incontro tra le situazioni precedentemente descritte, evitando overfitting e underfitting, prevede l'utilizzo del maggior numero di attributi e l'applicazione successiva dei metodi di **features reduction** (riduzione del numero di features).
Sono tecniche che tentano di migliorare la generalizzabilità dell'algoritmo rimuovendo gli attributi meno rilevanti; si cercano di rimuovere sia gli attributi ridondanti sia quelli irrilevanti ottenendo come risultato finale un sottogruppo delle features di partenza.
Oltre questa tipologia di tecniche, esistono infatti anche algoritmi di apprendimento supervisionato che possiedono già funzioni implementate che applicano la selezione di certe features, tra i quali la regressione logistica regolarizzata, approfondita nella sezione 4.2, e le foreste casuali.
Considerando soltanto i metodi standard e non le funzioni, le principali tecniche di riduzione del numero di features sono:

- *Soglia di varianza* : la soglia di varianza elimina gli attributi che non hanno una variazione sufficiente da osservazione a osservazione, in altre parole la varianza di questi attributi è inferiore a una certa soglia. Spesso queste features forniscono poco valore in termini generali all'algoritmo[21].

  Per esempio se si considera un dataset che voglia studiare la salute pubblica e il 98% delle osservazioni viene svolto su uomini e donne di 35 anni, allora l'attributo che tiene conto dell'età potrebbe essere eliminato senza una grossa perdita di informazione.

Siccome la varianza è una proprietà che dipende dalla scala di misure, si dovrebbe sempre normalizzare la lista degli attributi precedentemente. La normalizzazione permette di aggiustare i valori misurati su differenti scale e confrontarli su una scale comune. Tra i metodi di normalizzazione più diffusi si cita il *Min Max Scaler*, che per ogni campione x_i applica la seguente formula, portando i valori nel range $[0, 1]$:

$$x_i' = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{4.1}$$

e la *Standardization* (da non utilizzare per la soglia di varianza) ottenuta tramite:

$$x_i' = \frac{x_i - \mu}{\sigma} \tag{4.2}$$

dove μ e σ sono rispettivamente la media e la deviazione standard dei valori dell'attributo in considerazione.

La soglia di varianza è un metodo che presenta punti di forza notevoli poichè si basa su una solida intuizione ed è un metodo relativamente sicuro e facile da implementare per ridurre la dimensionalità, nonostante ciò, come metodo è raramente sufficiente e lo svantaggio principale risiede nel fatto di dover settare manualmente il valore della soglia di varianza sotto alla quale eliminare tutti gli attributi.

- *Soglia di correlazione*: la soglia di correlazione rimuove le features che hanno valori molto correlati gli uni con gli altri, in sintesi viene eliminato l'attributo che ha valori molto simili a un altro attributo. Il motivo è semplice, l'informazione portata dall'attributo simile è ridondante e quindi la sua eliminazione non comporta la perdita di informazioni.

  Per prima cosa il funzionamento di questa tecnica prevede il calcolo di tutte le coppie di correlazioni, successivamente se la correlazione di una coppia di features è superiore a una prefissata soglia allora una delle due verrà rimossa. La scelta dell'attributo nella coppia da rimuovere dipenderà dalla correlazione con gli altri attributi: verrà calcolata per entrambe la media delle correlazioni con tutte le altre features e quella che risulta con il valore più grande sarà eliminata (la sua descrizione è maggiormente ripresa dalle altre features quindi è una candidata migliore per essere rimossa).

Alcuni algoritmi sono influenzati negativamente da features correlate per cui rimuoverle può incrementare notevolmente le performance, tuttavia si presenta nuovamente il problema del settaggio manuale del valore di soglia il quale se è troppo basso rischia di rimuovere informazioni utili.

- *Test Statistici*: la selezione delle features avviene selezionando le migliori features basandosi su dei test statistici tra cui gli f-test e i test di informazione reciproca. I metodi basati sul test F stimano il grado di dipendenza lineare tra due variabili; d'altra parte, i metodi di informazione reciproca possono catturare qualsiasi tipo di dipendenza statistica, ma essendo non parametrici, richiedono più campioni per una stima accurata.

Questi sono metodi che ottengono un sottogruppo delle features di partenza, da cui il nome *features selection*, esistono però altri metodi il cui procedimento prevede la creazione di un nuovo set di features, costituito da meno attributi, che riesce comunque a mantenere la maggior parte dell'informazione. Questa categoria prende il nome di *features extraction*. Viene sottolineato in particolare il carattere distintivo di questa tecnica che prevede la costruzione di features derivate in modo tale da facilitare la fase di apprendimento e i passi di generalizzazione.
Alcune tecniche famose di features extraction sono:

- *Principal Component Analysis* (PCA): l'analisi delle componenti principali è un algoritmo non supervisionato che crea una combinazione lineare delle features originali e ci consente di rappresentare con un numero inferiore di variabili rappresentative la variabilità degli attributi del dataset originario.

  Nella PCA l'idea è quella di trovare un nuovo sistema di riferimento in modo da massimizzare la varianza delle variabili rappresentate lungo gli assi; le nuove features saranno ortogonali, perciò saranno anche non correlate a coppie.

  La *prima componente principale* (PC1) è caratterizzata dalla combinazione lineare normalizzata di un set di features $x_{i1}, x_{i2}, ..., x_{ip}$, che ha la più grande varianza possibile[22]:

$$z_1 = \phi_{11}x_{i1} + \phi_{21}x_{i2} + ... + \phi_{p1}x_{ip} \tag{4.3}$$

  Con la normalizzazione intendiamo che i pesi $\phi_{11}, ..., \phi_{p1}$ soddisfino la seguente condizione: $\sum_{j=1}^{p} \phi_{j1}^2 = 1$. La prima componente principale (PC1) considera la maggior varianza nel

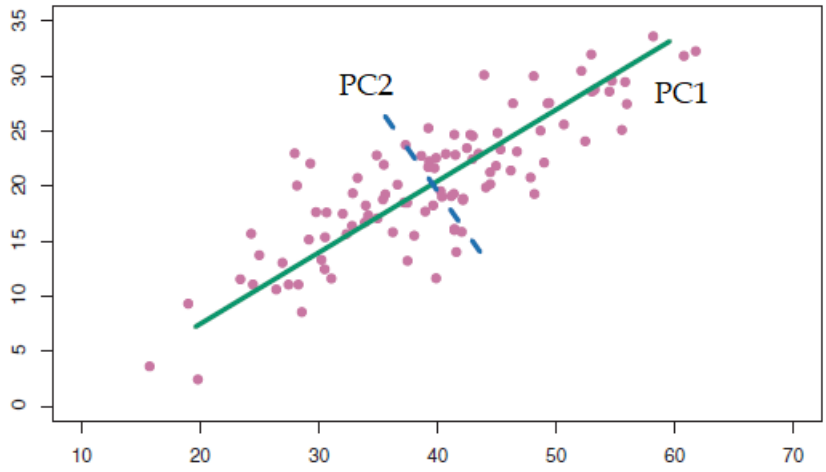


Figura 4.3: Prima e seconda componente principale

dataset, infatti i pesi $\phi_{11}, \phi_{21}, ..., \phi_{p1}$ definiscono nello spazio una direzione lungo la quale i dati variano in maniera massima. Se si proiettano gli n campioni lungo questa direzione, si ottengono gli n valori per la prima componente principale, quindi la prima features del nuovo dataset.

Determinata la prima componente si passa alla ricerca della seconda componente principale z_2 (PC2): è anche essa data dalla combinazioni lineare di $x_{i1}, x_{i2}, ..., x_{ip}$ che ha la varianza massima, tuttavia si considerano solo le combinazioni che sono non correlate a z_1. Costringere z_1 a essere non correlato a z_2 equivale a imporre che la direzione nello spazio definita da $\phi_{12}, \phi_{22}, ..., \phi_{p2}$ sia ortogonale alla direzione ϕ_1.

Nell'esempio della Figura 4.3 i dati giacciono in uno spazio dimensionale a due dimensioni, e quindi una volta trovata la prima componente principale (la linea verde continua) lungo la direzione di massima varianza c'è solo un'opzione per la seconda componente principale (la linea blu tratteggiata) che sia ortogonale alla precedente.

Ovviamente in un dataset con una dimensionalità maggiore di due, una volta trovata la prima componente principale, ci sarebbero più possibilità di direzioni per le successive componenti; il numero massimo di componenti che si possono ottenere è pari al numero di features di partenza.

Perciò, si può decidere di ridurre il numero di variabili limitando il numero di componenti principali da considerare. Ad esempio si potrebbe considerare di tenere solo le componenti

principali necessarie a raggiungere una varianza spiegata al 90%.

La PCA è una tecnica molto popolare e estremamente duttile e versatile. É facile da implementare e offre diverse variazioni e estensioni per affrontare problemi in diverse modalità. Il problema principale riguarda il fatto che le nuove componenti principali non sono interpretabili e il dataset viene completamente mutato, inoltre anche in questo caso è presente l'inconveniente di dover stabilire un parametro di soglia per la varianza cumulativa.

- *Linear Discriminant Analysis* (LDA): è una tecnica supervisionata basata anch'essa sulla combinazione lineare delle features di partenza. Piuttosto che massimizzare la varianza, questo metodo cerca di massimizzare la separabilità tra le classi.

  Per questo motivo LDA è un metodo che può essere utilizzato solo quando si ha un'etichettatura sulle classi. Anche in questo caso è sempre meglio applicare una normalizzazione del dataset prima di applicare l'algoritmo.

  LDA quindi cerca di trovare un nuovo spazio delle features per proiettare i dati in modo tale da massimizzare la separabilità tra le classi. Un metodo proposto per misurare la capacità di separazione di ogni nuovo spazio delle features consiste nel massimizzare la funzione che rappresenta la distanza tra le classi minimizzando la diffusione dei dati all'interno delle classi. In ogni caso questa formulazione è possibile solo assumendo di aver una distribuzione gaussiana del dataset.

  Si può riassumere il comportamento del linear discriminant analysis in 5 passi:

  1. Per ogni classe del dataset si calcola il vettore delle medie di ogni attributo.

  2. Si calcolano le matrici *within-class* S_W e *between-class* S_B:

  $$S_W = \sum_{i=1}^{c} \sum_{x \in D_i}^{n} (x - \mu_i)(x - \mu_i)^T \qquad S_B = \sum_{i=1}^{c} N_i(\mu_i - \mu)(\mu_i - \mu)^T \tag{4.4}$$

     dove μ è la media considerata su tutti i campioni mentre μ_i e N_i sono la media e il numero di campioni delle rispettive classi.

  3. Si trovano gli autovalori $(e_1, e_2, ..., e_d)$ e gli autovettori $(\lambda_1, \lambda_2, ..., \lambda_d)$ per la matrice

$S_W^-1S_B$ risolvendo

$$S_W^{-1}S_Bv = \lambda v \tag{4.5}$$

4. Si ordinano gli autovettori a seconda degli autovalori decrescenti. Gli autovettori con gli autovalori più piccoli sono quelli portano minore informazione sulla distribuzione dei dati e sono quelli che si vuole rimuovere. Si scelgono i primi k autovettori ($k < d$) ottenendo la matrice W di dimensione $d \times k$ dove ogni colonna è un autovettore.

5. Vengono trasformati i campioni originali del dataset X, di dimensione $n \times d$, nel nuovo sottospazio:

$$Y = X \times W \tag{4.6}$$

con i nuovi campioni Y che avranno dimensionalità ridotta $n \times k$.

LDA è un ottimo metodo per la riduzione della dimensionalità soprattutto in una classificazione multiclasse. Non sempre ottiene risultati ottimali per problemi complessi e presenta la stessa tipologia di svantaggi che affligge la PCA; al contrario di quest'ultima, il suo utilizzo è circoscritto solo nel caso di un apprendimento supervisionato.

4.2 Regressione Logistica Regolarizzata L1

Per evitare di regolare manualmente parametri di soglia per il progetto si è preferito utilizzare algoritmi con funzioni che permettessero di ridurre il numero di features.

Per l'approccio supervisionato è stato deciso di considerare il problema come uno di classificazione: lo scopo era quello di comprendere a quale delle due categorie opposte di lavorazione appartenesse il dato e non di prevedere l'andamento di un determinato valore, perciò i metodi di classificazione sono risultati la scelta più idonea.

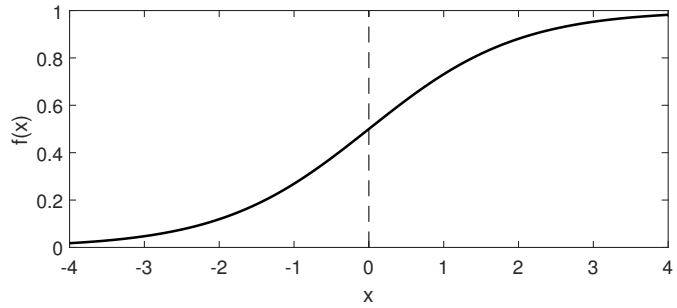
Il primo algoritmo selezionato nel caso supervisionato è stato quello della *regressione logistica regolarizzata*.

La regressione logistica è un metodo statistico di classificazione che tenta di stimare i parametri di un modello logistico. Si parla di regressione logistica binaria quando il modello ha una variabile dipendente con due possibili esiti (ad esempio successo - fallimento, lavorazione corretta - lavorazione non corretta) dove i due valori sono etichettati con 0 e 1. Le variabili indipendenti, ovvero le features

del dataset, contribuiscono attraverso una combinazione lineare a definire l'esito dell'appartenenza: il risultato finale è un numero compreso tra $[0, 1]$ che stabilisce la probabilità che il dato appartenga a una delle due classi. Se il valore sarà 0, certamente il dato apparterrà alla classe 0, se il valore sarà 1 apparterrà sicuramente alla classe 1.

La funzione che converte i valori del modello in valori di probabilità tra 0 e 1 è la funzione logistica (da cui il nome dell'algoritmo):

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}}$$



Per capire meglio questo tipo di modello, bisogna considerare l'idea che vi sta alla base: cercare di trovare un modello per la relazione tra la probabilità $P(x_i) = P(y = 1|x_i)$ e i dati x_i. Prima tutto volendo ottenere un modello che restituisca un output tra 0 e 1 non si può considera soltanto il modello di regressione lineare $p(x) = \beta_0 + \beta_1 x$ ma bisogna utilizzare la funzione logistica[20]:

$$p(x) = \frac{e^{\beta_0+\beta_1 x}}{1 + e^{\beta_0+\beta_1 x}} \tag{4.7}$$

Per adattare questo modello e trovare i migliori parametri si usa un metodo chiamato *massima verosomiglianza*, il quale cerca di massimizzare la funzione di verosimiglianza.

Dopo opportune manipolazioni si giunge alla seguente equazione:

$$\frac{p(x)}{1 - p(x)} = e^{\beta_0+\beta_1 x} \tag{4.8}$$

e prendendo il logaritmo:

$$\log\left(\frac{p(x)}{1 - p(x)}\right) = \beta_0 + \beta_1 x \tag{4.9}$$

si arriva alla valutazione finale del modello. La parte sulla sinistra viene anche chiamata *logit* o *log-odds*. I modelli sono ancora "lineari", di conseguenza l'algoritmo ha ottime performance quando le classi sono linearmente separabili.

La regressione logistica ha il vantaggio di poter utilizzare *la regolarizzazione* per ridurre il sovradattamento del modello.

La regolarizzazione ha come fine quello di abbattere l'impatto di alcune features in modo da ridurre la complessità del modello. Esistono sostanzialmente due tipi principali di regolarizzazione: la regolarizzazione L1 e la regolarizzazione L2 le quali differiscono per il modo di considerare la loss-function: mentre la prima minimizza la differenza in valore assoluto tra il target value y_i e il valore stimato $f(x_i)$, la seconda minimizza la somma dei quadrati delle differenze[23].

$$L1: \qquad S = \sum_{i=1}^{N} |y_i - f(x_i)| \tag{4.10}$$

$$L2: \qquad S = \sum_{i=1}^{N} (y_i - f(x_i))^2 \tag{4.11}$$

Applicando queste considerazione al metodo dei minimi quadrati, utilizzato per trovare i parametri migliori, si ottiene:

$$L1: \qquad \beta^* = arg\min_{\beta} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{4.12}$$

$$L2: \qquad \beta^* = arg\min_{\beta} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{4.13}$$

L'algoritmo ottenuto dall'applicazione della regolarizzazione L1 al modello di regressione logistica viene talvolta denominato *lasso* (mentre la regolarizzazione L2 applicata al modello di regressione viene chiamata anche *ridge regression*, per ulteriori approfondimenti vedere appendice C).

La regolarizzazione L2 presenta un notevole svantaggio: diversamente dagli algoritmi proposti nel capitolo precedente che trovavano modelli con un numero ridotto di features, la regolarizzazione L2 introduce un termine di penalità ($\lambda \sum_{j=1}^{p} \beta_j^2$) che restringerà tutti i coefficienti del modello verso zero ma nessuno sarà posto esattamente a zero a meno che il parametro λ non sia posto uguale a infinito (quindi direttamente non si elimina nessun attributo). Spesso questo non è un problema nell'accuratezza delle predizioni, tuttavia potrebbe risultare dannoso quando il numero di attributi è

grande.

Per questo motivo si è deciso di usare la tecnica del lasso (*Least Absolute Shrinkage and Selection Operator*) che ha un termine di penalità leggermente diverso $\lambda \sum_{j=1}^{p} |\beta_j|$. In questo caso la riduzione delle features arriva a un punto tale da riuscire a porle esattamente uguali a zero, quando il parametro da sintonizzare è scelto in maniera accurata ottenendo dunque una sorta di *features selection* del dataset di partenza. Come risultato i modelli generati dal lasso sono solitamente più facili da interpretare rispetto a quelli di ridge regression.

Il parametro λ svolge un ruolo critico e decisivo nella scelta dell'impatto dei due termini delle equazioni 4.12 e 4.13. Quando esso è nullo, il termine di penalità non ha nessuno effetto, e si produce solamente la stima ai minimi quadrati; al contrario se la λ va verso infinito l'impatto del termine di penalità cresce e la stima dei termini con il metodo dei minimi quadrati diventa irrilevante.

Per comprendere meglio perché la tecnica lasso pone a zero alcuni dei coefficienti del modello si può partire considerando una diversa formulazione per il problema[24]:

$$minimize_{\beta} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \quad subject \quad to \quad \sum_{j=1}^{p} |\beta_j| < s \tag{4.14}$$

in altre parole, per ogni valore di λ dell'equazione 4.12 esiste un parametro s per il quale si avranno gli stessi coefficienti del lasso (a un valore di s molto grande corrisponde un valore di λ molto piccolo). Se si suppone per esempio che la dimensionalità del dataset di partenza sia p $= 2$, allora la stima dei coefficienti del lasso è data dal più piccolo valore della somma al quadrato dei residui tra tutti i punti che giacciono nel rombo definito da: $|\beta_1| + |\beta_2| \leq s$.

Il grafico successivo aiuta a comprendere meglio intuitivamente la situazione.

Le ellissi che sono centrate in β^* (il punto blu) rappresentano le regioni dove la somma dei residui al quadrato è costante. Più le ellissi si allontanano, più il valore cresce. L'equazione (4.14) stabilisce che i coefficienti del lasso sono dati dal primo punto di contatto tra le regione vincolata dalla regolarizzazione (il rombo tratteggiato) e le ellissi. Siccome il vincolo del lasso ha angoli per ogni asse, l'ellisse talvolta intersecherà la regione vincolata proprio su un asse e quando ciò avviene, uno dei coefficienti sarà nullo.

In dimensioni più alte potrebbe accadere che contemporaneamente molti dei coefficienti siano nulli. Si ottiene quindi con il lasso un modello più semplice e facilmente interpretabile che coinvolge un

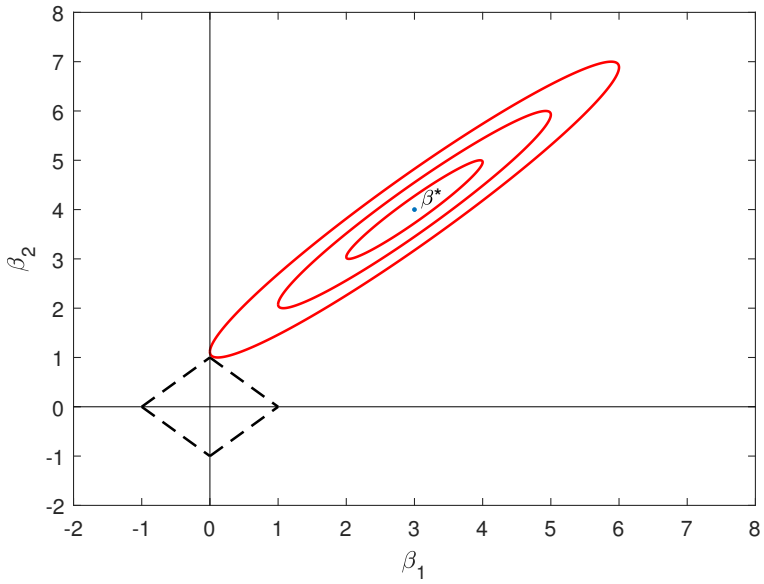


Figura 4.4: Lasso a due dimensioni

numero inferiore di attributi, tuttavia proprio per questa suo funzionamento, lasso risulta avere spesso performance inferiori rispetto a ridge regression nella predizione dell'errore.

Il parametro di regolarizzazione λ, come abbiamo visto, è fondamentale. Piuttosto di stabilire un valore più o meno casualmente, si è deciso di confrontare i risultati ottenuti implementando tre metodi differenti: la cross-validazione, il metodo BIC e il metodo AIC.
La tecnica della cross-validation per il parametro di regolarizzazione λ riprende il comportamento spiegato all'inizio del capitolo; la procedura prevede:

1. l'identificazione di un intervallo campionato di possibili parametri: $\lambda_0, \lambda_1, ..., \lambda_n$ (per esempio si potrebbe scegliere da un valore di $\lambda = 10^{-3}$ a un valore di $\lambda = 10$ con passo fisso).

2. la divisione del dataset in K-cartelle (K-fold cross validation)

3. il calcolo per ogni λ dell'errore di cross-validazione, dato dalla media degli errori per ogni cartella di test.

4. la scelta finale del parametro λ che ha fornito il valore minimo dell'errore

I metodi AIC e BIC[25] sono metodi di selezione del modello migliore che possono essere anche applicati per determinare il parametro λ più efficiente. Sono tecniche molto veloci e assumono che

il modello di generazione dei dati sia corretto, tuttavia hanno risultati deboli quando il problema ha tante features rispetto al numero dei campioni.
Il criterio AIC (criterio di informazione di akaike) è in generale definito come:

$$AIC = 2p - 2\ln(L) \tag{4.15}$$

dove p è il numero di parametri del modello e L il valore massimizzato della massima verosimiglianza del modello stimato. La regola prevede di selezione i modelli con l'AIC più basso.
Il criterio BIC (*criterio di informazione bayesiano*), invece viene definito in maniera generale dalla seguente equazione:

$$BIC = -2ln(L) + pln(n) \tag{4.16}$$

In corrispondenza a modelli con un basso valore per l'errore nel dataset di testing, si avrà un basso valore di BIC; di conseguenza anche in questo caso verranno selezionati i modelli con i minori valori di BIC.

4.3 K-Nearest Neighbours

Il secondo tipo di algoritmo considerato per valutare l'accuratezza delle previsioni sulla lavorazione corretta e lavorazione non corretta è stato il *K-nearest neighbors*[24]. É sicuramente uno tra gli algoritmi più semplici nell'ambito del machine learning, nonostante ciò è molto diffuso grazie al fatto di sapersi adattare con buoni risultati in differenti applicazioni anche con ottimi tempi di calcolo.
K-NN è un algoritmo di apprendimento supervisionato che utilizza la conoscenza della classificazione dei K punti più vicini per assegnare la determinata classe a un dato.
Il suo funzionamento si basa dunque sulla somiglianza, più un campione è vicino a un determinato punto, più il K-NN li considererà simili: solitamente viene utilizzata la distanza euclidea come misura della somiglianza (minore la distanza, maggiore sarà la somiglianza). Il numero di vicini da considerare è determinato dal parametro K, scelto arbitrariamente; la selezione di un K accurato è cruciale per la buona riuscita dell'algoritmo; ad esempio su un problema di classificazione binaria

si preferisce utilizzare un K dispari per evitare di avere un pareggio nell'attribuzione del risultato. Tra i metodi per trovare il K migliore si utilizza spesso la cross-validation, la quale risulta tuttavia abbastanza onerosa da un punto di vista computazionale per questa situazione, per cui talvolta si preferisce confrontare i risultati su un sottogruppo ristretto di K selezionati a priori grazie ad informazioni in possesso.

Il procedimento dell'algoritmo K-nearest neighbors può essere descritto in cinque step:

1. Scelta del valore per K
2. Fase di normalizzazione per rendere le misure confrontabili
3. Calcolo della distanza euclidea tra il dato in esame e i dati di training
4. Riordino delle distanze calcolate dalla più piccola alla più grande
5. Scelta delle prime K distanze. La classe che si presenta in maggior numero per i K punti considerati stabilirà l'esito della classificazione.

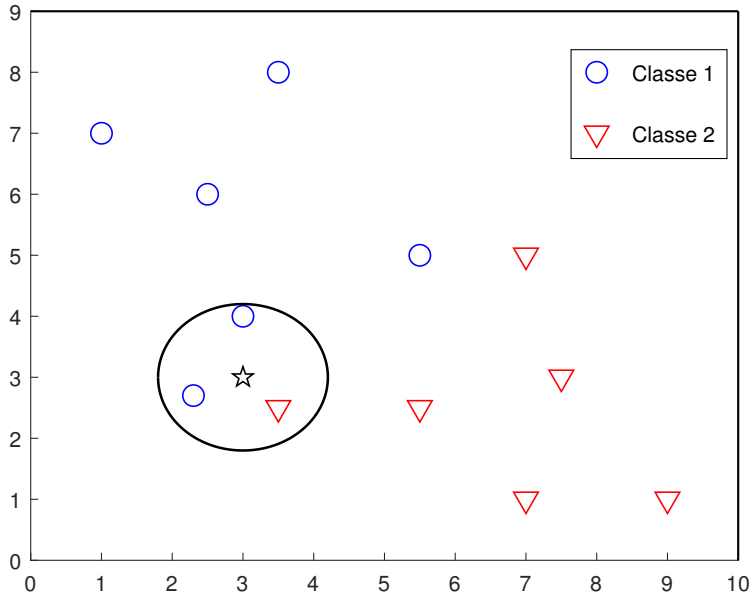


Figura 4.5: K-NN per classificazione binaria

La figura (4.5) mostra un esempio di applicazione dell'algoritmo K-NN con parametro K $= 3$. In questo caso la previsione sul dato di test (indicato con una stella) viene fatta considerando i tre punti più vicini; essendo maggiore la presenza della classe uno, il dato sarà classificato appartenente

a tale classe.

Un'estensione del K-NN è data dal *weighted K-NN*, in cui l'idea è quella di pesare il contributo di ogni vicino in base alla distanza dal punto considerato. In questo modo più un punto è vicino più è importante per la classificazione.

Un vantaggio da evidenziare per l'algoritmo K-NN è il non fare alcun ipotesi di distribuzione dei dati che analizza. Anzi, la struttura del modello è proprio determinata dai dati e ciò è piuttosto utile soprattutto nella maggior parte delle applicazioni reali in cui i dati non obbediscono a modelli predeterminati. Quindi è un modello che dovrebbe essere sfruttato quando si ha poca o nessuna conoscenza precedente sulla distribuzione dei dati.

4.4 Decision Tree Classifier

Un approccio diverso è stato intrapreso grazie all'algoritmo di classificazione basato sugli alberi decisionali[24].

L'albero decisionale è una struttura dati composta da nodi e archi che va interpretata dall'alto verso il basso. Ogni nodo rappresenta una variabile, o meglio una funzione condizionale che ha due o più diramazioni verso il basso. Partendo dal nodo radice (il nodo iniziale), il processo prevede lo svolgersi di una sequenza di test e a seconda dell'esito della rilevazione, la scelta di una direzione oppure dell'altra. Il cammino si conclude in una foglia (nodo esterno) che indicherà il valore della previsione finale.

Quindi solitamente da un set di dati di test si trovano le condizioni primarie delle prove per creare successivamente l'albero di decisione e verificarlo sui dati di test. É importante sottolineare come sia fondamentale definire un criterio di arresto, un limite massimo di profondità dell'albero oltre il quale non si può andare. Il motivo è semplice: un aumento spropositato delle dimensioni di un albero porta a una crescita esponenziale dell'aspetto computazionale e, tuttavia, ciò non garantisce una sufficiente crescita correlata dell'accuratezza. Tra l'altro maggiore è la profondità, maggiore è il rischio di overfitting.

Per questa ragione l'obiettivo è quello di addestrare il modello in modo tale che esso riesca a raggiungere una foglia svolgendo il numero minore di test possibili. Esistono diverse tecniche,

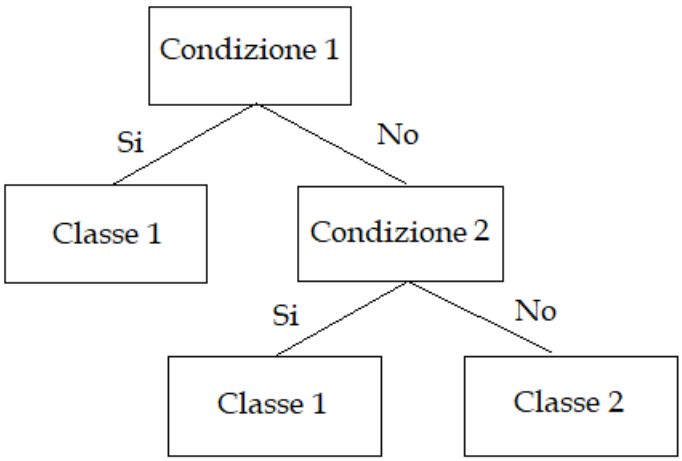


Figura 4.6: Decision Tree Classifier

chiamate *metriche di impurità*, per stabilire quanto sia efficiente la divisione di un nodo; si dice che il nodo è puro e il suo valore è zero, quando il nodo contiene esempi di una sola classe, ovvero quando l'albero ha raggiunto una foglia.

Le più popolari metriche di impurità sono:

- *indice Gini*:

$$I_G(i) = 1 - \sum_{j=1}^{m} f(i,j)^2 \tag{4.17}$$

- *entropia*

$$I_E(i) = -\sum_{j=1}^{m} f(i,j) \log f(i,j) \tag{4.18}$$

dove in entrambe le formule f rappresenta la frequenza del valore j nel nodo i. Portano entrambe a risultati molto simili, anche se è maggiormente utilizzato l'indice Gini perché è meno dispendioso in termini di risorse di calcolo.

E' sicuramente un algoritmo semplice e facilmente implementabile al computer, inoltre mostra chiaramente come la macchina giunga alla fase di decisione e questo alcune volte può essere un notevole vantaggio. Tuttavia sono chiari anche i difetti: la rappresentazione ad albero decisionale è poco adatta a problemi complessi, in quanto le ipotesi assumono una grandezza esagerata e la complessità computazionale diventa proibitiva. Uno sviluppo degli alberi decisionali è rappresentato dalle *foreste casuali* che consentono di sfruttare più alberi contemporaneamente (APPENDICE D).

4.5 Support Vector Machine

Come ultimo approccio per la classificazione è stato preso in considerazione l'algoritmo denominato *support vector machine* (macchina a vettori di supporto)[24].

Il support vector machine nasce generalizzando e estendendo ai casi non lineari un semplice e intuitivo classificatore chiamato *maximal marginal classifier*(classificatore a massimo margine) o anche *hard margin classifier*. Quest'ultimo, che viene utilizzato quando le classi di partenza sono perfettamente linearmente separabili, si basa sul fatto di dover trovare l'iperpiano in grado di separarle; tra le molte possibili soluzioni, si cerca l'iperpiano definito dai valori $\beta_0, \beta_1, ..., \beta_p$ in grado di massimizzare la separazione tra le classi:

$$\begin{aligned} &\underset{\beta_0,\beta_1,...,\beta_p}{maxizime} \quad M \qquad\qquad subject \quad to \quad \sum_{j=1}^{p} \beta_j^2 = 1 \\ &y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip}) \geq M \quad \forall i = 1, ..., N \end{aligned} \tag{4.19}$$

dove M indica il margine e (x_i, y_i) il dato di training con la classe di appartenenza. A seconda che il dato si trovi sopra o sotto l'iperpiano sarà classificato di conseguenza.

Questa situazione tuttavia è troppo restrittiva, il caso ideale in cui le classi siano perfettamente separabili è a dir poco raro in una situazione reale. Perciò si fa di solito riferimento al *support vector classifier* (o *soft margin classifier*), il quale preferisce trovare un iperpiano che non sia perfetto nella separazione nell'interesse di trovare una maggiore robustezza alle osservazioni individuali e una migliore classificazione della maggior parte dei dati. In altre parole vale la pena sbagliare poche classificazioni nei dati di training per ottenere un migliore risultato nel resto delle osservazioni.

Questo classificatore è la soluzione a un problema di ottimizzazione differente, con più condizioni:

$$\begin{aligned} &\underset{\beta_0,\beta_1,...,\beta_p,\epsilon_1,...,\epsilon_N}{maxizime} \quad M \qquad\qquad subject \quad to \quad \sum_{j=1}^{p} \beta_j^2 = 1 \\ &\epsilon_i \geq 0, \quad \sum_{i=1}^{N} \leq C \\ &y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip}) \geq M(1 - \epsilon_i) \quad \forall i = 1, ..., N \end{aligned} \tag{4.20}$$

in questo caso le variabili ϵ_i sono quelle che consentono alle osservazioni di essere nel lato sbagliato del margine mentre C è un parametro da sintonizzare per determinare la tolleranza sulla violazione

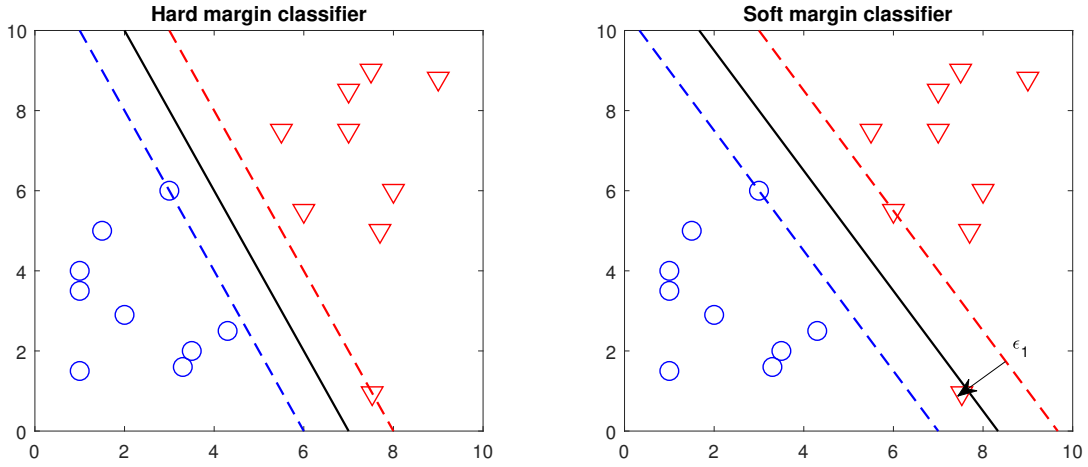


Figura 4.7: Hard e Soft Margin Classifier

del margine e in caso dell'iperpiano. Se $C = 0$, non c'è alcuna tolleranza sulle violazioni, perciò $\epsilon_1 = ... = \epsilon_N = 0$, invece quando C aumenta c'è più tolleranza, e quindi il margine diventa più largo.

Il problema di ottimizzazione 4.20 ha una proprietà molto interessante: risulta infatti che solo le osservazioni che giacciono nel margine o quello che lo violano influiscono nel determinare l'iperpiano e quindi il classificatore stesso. Questi dati influenti sono noti anche come *vettori di supporto*; al contrario le osservazioni che giacciono distanti dal margine non saranno interessanti per il classificatore, e un loro non spostamento non modificherà la situazione, a meno che non diventino vettori di supporto.

Dall'ottimizzazione si può ricavare la soluzione del linear support vector classifier che può essere rappresentata come:

$$f(x) = \beta_0 + \sum_{i=1}^{N} \alpha_i < x, x_i > \tag{4.21}$$

dove ci sono N parametri α_i, uno per ogni dato di training, che risultano diversi da zero solo se sono vettori di supporto.

Come prima, il dato di test x^* viene classificato in base al segno della funzione ricavata dall'iperpiano (si deve calcolare $f(x^*) = \beta_0 + \beta_1 x_1^* + \beta_2 x_2^* + ... + \beta_p x_p^*$).

L'algoritmo support vector machine (SVM) riprende questi concetti e li estende a casi non lineari compiendo un aumento dello spazio degli attributi in un modo specifico, attraverso i *kernels*. L'idea alla base è quella di aumentare lo spazio delle features proprio per cercare di descrivere il confine di

decisione non lineare tra le classi.

Nel caso di support vector machine la funzione si modifica presentando una forma del tipo:

$$f(x) = \beta_0 + \sum_{i=1}^{N} \alpha_i K(x, x_i) \tag{4.22}$$

I kernels $K(x_i, x_{i'})$ sono funzioni che cercano di misurare la somiglianza tra due osservazioni[26]. Riprendendo il caso precedente del support vector classifier, il kernel assume una forma lineare descritta dalla seguente equazione:

$$K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j} \tag{4.23}$$

Scelte popolari per kernel di tipo non lineare sono:

- *Kernel Gaussiano* (RBF):

$$K(x_i, x_{i'}) = e^{\gamma \|x_i - x_{i'}\|^2} \tag{4.24}$$

  Il parametro da sintonizzare γ svolge un ruolo significativo nelle performance del kernel e dovrebbe essere selezionato a mano con particolare attenzione in riferimento al problema . Se si dovesse scegliere un valore troppo piccolo, l'esponenziale si comporterebbe quasi linearmente e la proiezione in una dimensione superiore inizierebbe a perdere il suo carattere non lineare. D'altra parte, se si dovesse scegliere un valore troppo alto, la funzione risulterebbe troppo sensibile al rumore nei dati di allenamento, aumentando il rischio della presenza di overfitting dei dati.

- *Sigmoid Kernel*:

$$K(x_i, x_{i'}) = \tanh(\alpha x_i^T x_{i'} + c) \tag{4.25}$$

  Ci sono due parametri aggiustabili a seconda dei casi: la pendenza α e l'intercetta c. Un valore comune per α è dato da $1/N$ dove N è il numero di dati.

In conclusione si può ritenere l'algoritmo support vector machine un approccio molto remunerativo soprattutto nel caso di una classificazione non lineare. Altri pregi da sottolineare sono la capacità di essere efficiente e flessibile in dimensioni elevate, e di non occupare troppa memoria (infatti sono da salvare solo i vettori di supporto). Il fatto di non avere un'interpretazione chiara e limpida e di

non restituire un'interpretazione probabilistica diretta per l'appartenenza alla classe, rende palesi anche gli svantaggi, i quali sono assolutamente da tenere in considerazione in una panoramica generale.

4.6 Risultati

Gli algoritmi descritti nelle sezioni precedenti sono stati poi testati per osservare le diverse qualità di accuratezza per la tipologia di dataset ottenuto alla fine del capitolo tre.

Prima di applicare le diverse funzioni, è stato notato come il numero di features del dataset di partenza fosse abbastanza grande (76) in confronto a una quantità non troppo elevata di esempi (1400). Valutato ciò, si è deciso allora di partire con l'utilizzo di alcuni algoritmi di features reduction, per ottenere un modello ridotto e semplificato che evitasse il fenomeno dell'overfitting.

Date le scarse prestazioni ottenute con metodi quali la soglia di varianza e la soglia di correlazione, e non volendo modificare il dataset di partenza per non compromettere l'interpretabilità, quindi evitando la principal component analysis e la linear discriminant analysis, si è fatto affidamento sulla regressione logistica regolarizzata L1. Questo metodo, oltre a restituire un valore di accuratezza per il modello, permette di selezionare le migliori features da considerare per i test successivi.

Per quanto riguarda il miglior parametro λ per la regolarizzazione, sono stati utilizzati i metodi BIC,

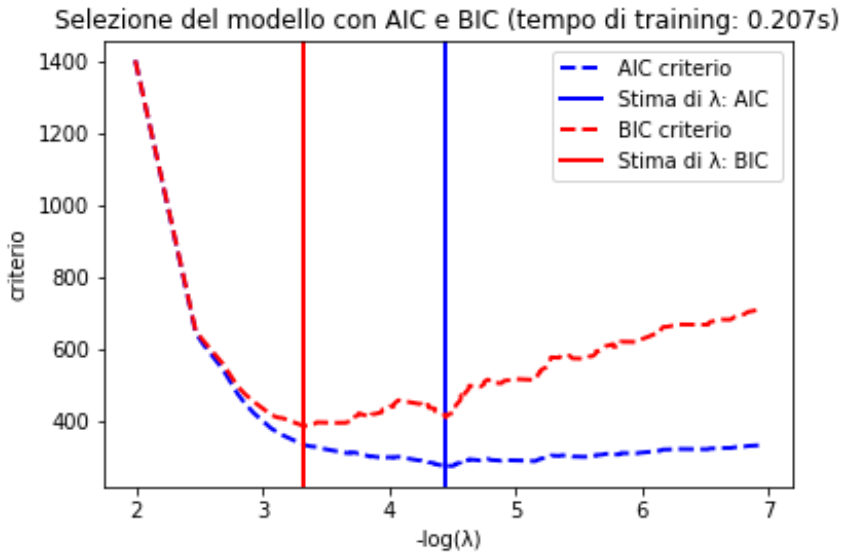


Figura 4.8: Miglior λ con BIC e AIC

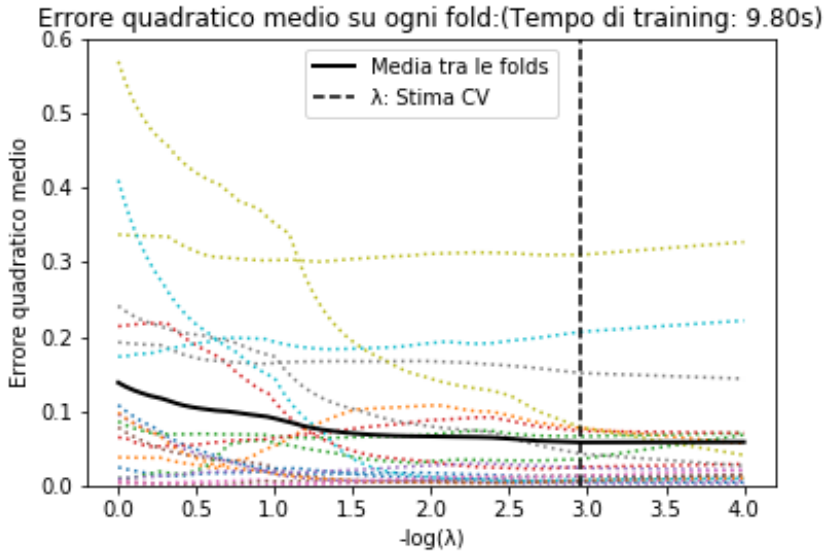


Figura 4.9: Miglior λ con Cross Validation

AIC e la cross- validation che hanno fornito rispettivamente: $\lambda_{BIC} = 4,7 \times 10^{-4}$, $\lambda_{AIC} = 3,5 \times 10^{-5}$, $\lambda_{CV} = 1,1 \times 10^{-4}$. La decisione è ricaduta sul parametro dato dalla cross validation che fornisce il valore maggiore tra i lambda migliori e anche l'errore quadratico medio minore.

Così facendo si è ottenuto un dataset ridotto composto da 7 features: con gli attributi rimasti che

	Campioni	Features
DATASET 1R:	1401	7

sono:

Lista 1R: *filt_Fattore di Margine e fft_Fattore di Margine per il Montronix; Energia, filt_Fattore di Margine, filt_Energia, ftt_Kurtosis, fft_Fattore di Margine per il PCB piezotronics*

Per valutare le performance di accuratezza del modello di regressione logistica si è utilizzato di nuovo la cross validation con un numero di cartelle pari a dieci. Si sono utilizzati due parametri per misurare la qualità dei risultati forniti: l' *accuratezza*, che tiene conto di quante classificazioni errate sono state ottenute e il cui valore ideale sarebbe 1, e la *log-loss* che misura le performance di un modello di classificazione che restituisce un valore di probabilità p; un modello perfetto avrà una log-loss pari a zero; l'obiettivo dunque sarà quello di minimizzare questo valore. La log-loss tiene conto dell'incertezza della predizione basandosi su quanto varia rispetto alla vera predizione. In una

classificazione binaria se la vera classe è 1 il valore della log-loss sarà $-log(p)$, altrimenti se la vera classe a cui appartiene il dato è 0 il valore sarà dato da $-log(1-p)$.
Per la regressione logistica nei dati di testing sono stati ottenuti i seguenti valori: 0.914 per l'accuratezza e 0.223 per la log-loss.
Il dataset ridotto con le migliori features è stato utilizzato come nuovo punto di partenza per l'analisi degli algoritmi successivi.

Un metodo che ha fornito risultati simili ma leggermente inferiori è stato KNN. La decisione sul valore ottimo per il parametro K, il quale determina il numero di vicini da considerare, è stata risolta reiterando l'algoritmo con diversi valori e scegliendo quello che avesse complessivamente un valore ottimo sia per l'accuratezza sia per la log-loss. Infatti per un numero di K abbastanza piccolo (pari a 1 o 3) sono risultati valori ottimi per l'accuratezza ma valori non sufficienti per la log-loss, al contrario per valori troppo grandi di K le performance diminuivano drasticamente per l'accuratezza a fronte di un leggero miglioramento nella log-loss. Si è deciso di applicare l'algoritmo considerando i primi 13 vicini influenti ottenendo i seguenti risultati: 0.871 per l'accuratezza e 0.332 per la log-loss. Probabilmente questo è un tipo di modello troppo semplice e non riesce ad adattarsi in maniera ottimale proprio perché semplifica troppo ciò che distingue una lavorazione corretta da una non corretta.

Il classificatore basato sugli alberi decisionali, invece, ottiene risultati sorprendenti sia nei dati di test che nei dati di training. Per non rendere il calcolo computazione troppo oneroso si è deciso di imporre un livello massimo di profondità dell'albero pari a 6, mentre per quanto concerne il criterio da adoperare, la scelta è ricaduta sull'indice Gini. Nonostante questi valori molto alti per l'accuratezza (oltre il 98% nei dati di training e circa il 95% nei dati di testing) il decision tree classifier non è un approccio robusto. Oltre ad adattarsi esageratamente ai dati di allenamento, il modello non riuscirebbe a generalizzare a vibrazioni leggermente diverse che caratterizzano un tipo di lavorazione non simile a quelle simulate. Per questo motivo sostanzialmente questo approccio è stato scartato.

Infine per osservare il comportamento di un classificatore non lineare si è deciso di utilizzare la support vector machine con due diversi tipi di kernel: il kernel gaussiano (o radial basis function) e il kernel sigmoidale.

	Train		Test	
	Acc.	Log Loss	Acc.	Log Loss
Reg Log L1:	0.925	0.196	0.914	0.223
KNN:	0.902	0.219	0.871	0.332
DTC:	0.981	0.212	0.946	0.243
SVM rbf:	0.9661	X	0.9537	X
SVM sigm:	0.7857	X	0.7687	X

Tabella 4.1: Risultati algoritmi di classificazione

Siccome la SVM non restituisce un valore di probabilità la log loss non può essere usata. Dal confronto delle due diverse tipologie si può comprendere come la scelta di un kernel risulti determinata nelle performance della support vector machine: infatti mentre la SVM con kernel gaussiano ottiene i migliori risultati nella classificazione, la SVM con kernel sigmoidale ottiene di gran lunga i risultati peggiori, come si può vedere dalla Tabella 4.1. Il valore selezionato per γ è stato quello standard di $1/p$ dove p indica il numero di features. A causa della sua applicazione a contesti non lineari, questo algoritmo riesce in maniera ottimale a distinguere quando la lavorazione è corretta e quando la lavorazione non lo è, cercando di massimizzare il margine di separazione tra le classi. Anche da un punto di vista computazionale è un'ottima scelta perché non risulta troppo dispendioso e oneroso, fornendo risultati ottimi e veloci anche in tempo reale.

Capitolo 5

Anomaly Detection: Approccio Non Supervisionato

Il secondo tipo di approccio considerato è stato quello non-supervisionato. Questo caso trova notevoli applicazioni in contesti reali, soprattutto qualora non si abbia o non si possa ottenere un'etichettatura immediata che indichi l'esito di una classificazione.

É questo il carattere distintivo dell'apprendimento non-supervisionato[27]: cercare di scoprire pattern non precedentemente sconosciuti da dati non etichettati. Le applicazioni più comuni per questa branca del machine learning riguardano il tentativo di caratterizzare i dati dividendoli in gruppi a seconda della loro somiglianza e il catturare punti che hanno una diversità troppo elevata, e che per questo vengono considerati anomali[28].

Per il nostro problema si è pensato di utilizzare questo approccio quando in una situazione real-time si volesse cercare di prevedere l'andamento di una lavorazione prima che essa completi una lavorazione non corretta, cercando di estrarre una regola dai dati o una relazione che leghi lavorazioni simili tra loro, oppure quando l'esito di una classificazione non fosse verificabile prontamente in specifiche situazioni.

Un altro possibile approccio è quello di cercare di identificare e trovare comportamenti anomali rispetto a un gruppo di lavorazioni corrette, i quali saranno categorizzati come *outlier*, il tutto senza sapere la classe di appartenenza. L'*anomaly detection* si riferisce proprio a questo: l'identificazione di eventi, dati o esempi rari i quali si evidenziano per essere significativamente diversi rispetto alla maggioranza del dataset. Nelle tecniche di anomaly detection non supervisionate si cerca di

riconoscere gli outlier in un data set non classificato, sotto l'assunzione che la maggioranza dei dati sia una rappresentazione normale del fenomeno (*inlier*), tra tutte le istanze che si adattano meno al resto del gruppo di dati.

Per quanto riguarda i diversi approcci di anomaly detection si distinguono due tipi principali: il primo basato sulla costruzione di un modello che sfrutta la distinzione presente tra un sistema reale di ingresso-uscita e un sistema di riferimento e il secondo basato sui dati senza la necessità di stabilire a priori la costruzione di un modello del sistema. Per il nostro caso, si è preferito essere più elastici e non costruire un modello di riferimento del sistema ma essere guidati solamente dai dati.

A sua volta questo secondo approccio fa riferimento o alle features o alle serie temporali. Nel primo caso gli input sono osservazioni con una dimensione pari a p, corrispondente al numero di attributi; nel secondo caso gli ingressi per l'algoritmo di anomaly detection sono serie temporali composte da un numero M di campioni e un dimensionalità del segnale pari s. Data la composizione del dataset, si farà riferimento a quegli algoritmi che fanno affidamento sulle features delle osservazioni.

Un'altra categorizzazione per i metodi di anomaly detection può essere ottenuta dividendoli tra algoritmi che si basano sulla distanza (*distance-based*) e algoritmi che si basano sulla distribuzione (*distribution-based*)[29]:

- *distance-based*: questi metodi si basano sull'idea che osservazioni normali siano adiacenti ad altre osservazioni mentre osservazioni anomale siano situate a distanza notevole da tutti gli altri punti.

- *distribution-based*: questi metodi prevedono che gli inlier siano radunati in regioni densamente popolate da dati, mentre le anomalie sono quelle osservazioni presenti in regioni a bassa densità di dati. Tra gli algoritmi costruiti su questa idea c'è sicuramente il *Local Outlier Factor*.

Prima di procedere con l'analisi degli algoritmi utilizzati, è stato necessario ottenere un dataset diverso da quello di partenza. Il diverso obiettivo che pone l'anomaly detection ha imposto l'utilizzo di un dataset differente composto al 90% da prove corrette e al 10% da prove di lavorazione non corrette. In questa maniera abbiamo ottenuto gli outlier da ricercare attraverso gli algoritmi di anomaly detection proposti.

Per la creazione di questo dataset si è fatto riferimento solo alle prove di lavorazione effettiva (dunque il test 3 e il test 4), per questo motivo il numero di dati è di molto inferiore; si hanno un

totale di 232 osservazioni: 209 delle quali di lavorazione corretta e 23 di lavorazione anomala. Il numero di features considerato non è stato cambiato, si utilizzano ancora 76 attributi per descrivere il fenomeno nel suo complesso.

É importante sottolineare nuovamente come gli algoritmi non sappiano se il dato provenga da una lavorazione corretta o meno, altrimenti non si rientrerebbe nel caso di apprendimento non supervisionato; questa informazione, di cui disponiamo per la particolare costruzione del dataset, è stata inserita "a mano" successivamente proprio per valutare l'accuratezza dei diversi algoritmi proposti: è una scorciatoia per cercare di stabilire quale metodo in un caso non supervisionato fornisca i risultati migliori.

	Campioni	Features
DATASET 2:	232	76

5.1 Riduzione del numero di features

Come sottolineato nei capitoli precedenti, i dati con un numero elevato di features sono soggetti a un rischio di overfitting assai notevole, perciò la riduzione degli attributi svolge un ruolo importante anche in un contesto non supervisionato. Allo stesso modo del caso supervisionato, due sono le vie principali secondo con cui ottenere la riduzione: la features selection e la features extraction. Mentre con la prima tecnica si riesce a ottenere un sottogruppo delle features del dataset originario di partenza, con la seconda si ottiene invece un dataset completamente modificato da cui poi vengono rimossi alcuni attributi.

Per la features selection oltre agli algoritmi che limitano la varianza o la correlazione precedentemente elencati, si può utilizzare un metodo noto come *sequential backward selection* (SBS)[30].

É una tecnica che può essere descritta in pochi passaggi:

- si seleziona il numero finale d di features che si vuole ottenere
- viene selezionata una funzione che calcola la perdita di performance ottenuta dalla rimozione di una feature

- per ogni attributo viene calcolato il valore di questa funzione

- la feature che presenta il valore più piccolo per la funzione verrà eliminata (è quella dalla quale si perde minor informazione)

- si reitera l'algoritmo fino a quando non si raggiunge un numero di features pari a d

É una buona tecnica specialmente nel caso si riesca a trovare una buona funzione che descriva la perdita di performance dell'algoritmo; d'altra parte stabilire a priori un numero di attributi da ottenere non è sempre un approccio molto robusto.

Considerando la features extraction l'algoritmo più famoso e sicuramente tra i più utilizzati è la principal component analysis. Come abbiamo visto nella sezione 4.1 la PCA è una tecnica che calcola le componenti principali e estrae quelle più rilevanti, rendendolo un algoritmo molto utile per rimuovere dati molto rumorosi e per visualizzare dati rappresentati da molte dimensioni.
Un altro algoritmo abbastanza noto che si può utilizzare in questi casi è denominato *indipendent component analysis* (ICA)[31]. Diversamente dalla PCA che cerca di massimizzare la varianza, ICA assume che le features siano generate da una mescolanza di sorgenti indipendenti e cerca di isolare queste sorgenti indipendenti mischiate nel datasest.

Questi due approcci hanno un rischio intrinseco nella loro applicazione: quando si selezionano le features che massimizzano la varianza o che cercando di estrarre le componenti indipendenti, può capitare che il carattere anomalo di un dato venga evidenziato da componenti che sono ritenute più insignificanti le quali di conseguenza non sarebbero incluse nel dataset ridotto. Così facendo non sarebbe possibile riconoscere le anomalie in quanto esse si rispecchiano in queste variabili isolate.
Per questo spesso è più utile cercare di comprendere e evidenziare gli attributi che caratterizzano maggiormente un' osservazione da cui si estrae un gruppo di features, piuttosto che abbracciare l'applicazione di un algoritmo a scatola chiusa che potrebbe eliminare le anomalie che si vogliono studiare.

5.2 Isolation Forest Detector

La maggior parte dei modelli esistenti per approcciarsi all'anomaly detection costruisce un profilo di un'istanza normale e successivamente identifica gli esempi che non sono conformi al profilo normale come outlier. Questo approccio ha essenzialmente due svantaggi principali: il primo riguarda il riconoscitore dell'anomalia che viene ottimizzato per delineare un'istanza normale e non per riconoscere le anomalie e come conseguenza potrebbe avere risultati non eccelsi nell'anomaly detection, causando troppi falsi allarmi o troppo poche anomalie identificate; il secondo si riferisce al fatto che molti di questi metodi possono essere utilizzati solo su dimensioni ridotte e dataset contenuti a causa dell'elevato costo computazionale.

Un modello che esplicitamente cerca di isolare le anomalie piuttosto di profilare un'istanza normale è l'*isolation forest detector*[32]. Per utilizzare in maniera efficace questo metodo il dataset dovrebbe essere composto da anomalie che sono in numero nettamente inferiore e con valori degli attributi completamente diversi rispetto a quelli delle osservazioni normali.

Il principio di funzionamento si basa sulla struttura dati ad albero che può essere costruita per ottenere risultati performanti nell'isolare un'anomalia. Il tipo di albero utilizzato in questa tecnica viene denominato *isolation tree*: ogni nodo di quest'albero risulta essere o un nodo esterno con nessun figlio, o un nodo interno rappresentato da un test e con esattamente due figli. Il test consiste in un attributo q e un valore di divisione p scelti casualmente, cosicché se $q < p$ si proseguirà sul nodo sinistro, altrimenti si andrà sul nodo destro.

Per loro natura le anomalie saranno più isolate e quindi anche più vicine al nodo dell'albero al contrario degli inlier che saranno nei livelli più profondi dell'albero. Il metodo si propone di costruire un gruppo di alberi che compiono un partizionamento dei dati in maniera casuale; a questo punto si definisce un *anomaly score* che quantifica il numero medio di partizioni per isolare un osservazione in esame: le anomalie saranno quelle osservazioni che hanno i percorsi più brevi in questi alberi di isolamento. Il numero di alberi da utilizzare è un parametro che bisogna tenere in considerazione, si può dimostrare che le performance di iForest convergono abbastanza velocemente con un numero non elevato di alberi.

La figura 5.1 è esplicativa per descrivere l'approccio dell'*isolation forest*: mentre per isolare un osservazione normale, x_1, ci vogliono molte partizioni (in questo caso 7), per isolare un dato anomalo,

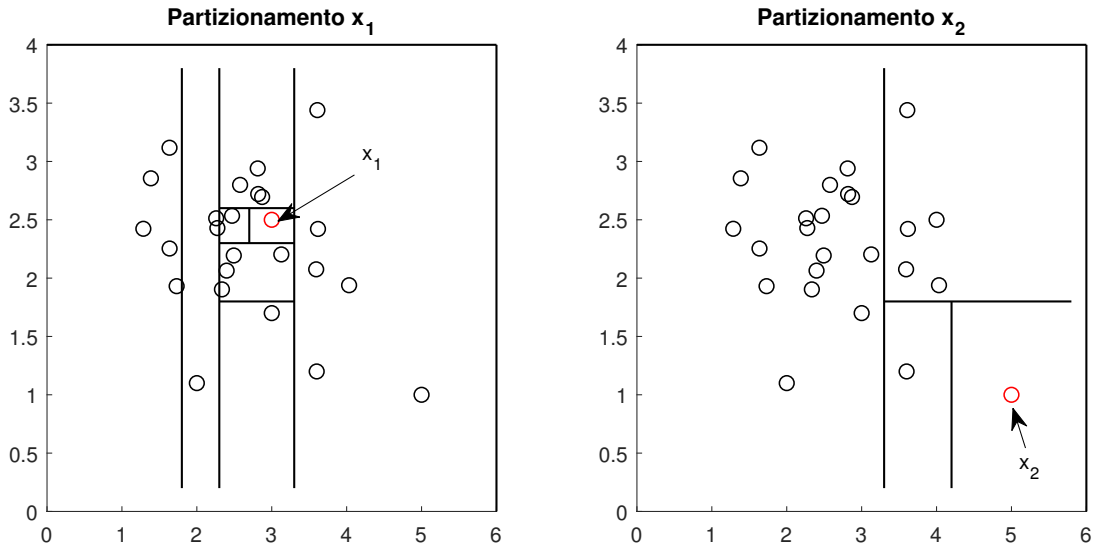


Figura 5.1: iForest

rappresentato da x_2, le partizioni utilizzate sono molto meno (per il caso specifico 3). Reiterando il procedimento utilizzando sempre partizioni casuali, si otterrà che in media x_2 avrà un anomaly score decisamente più alto di quello di x_1.

Tra i punti di forza di questo metodo c'è sicuramente quello di non utilizzare metriche di densità o distanza tra le osservazioni per riconoscere un'anomalia; ciò elimina la maggior parte dei costi computazionali rendendolo un algoritmo molto più veloce rispetto a tutti i metodi basati sulla distribuzione o sulla distanza.

Inoltre come algoritmo ha una complessità tempo lineare con un basso utilizzo di memoria, rendendolo di fatto uno dei modelli più utilizzati. Infine riesce ad adattarsi adeguatamente a dataset molto grandi con un numero elevato di features anche irrilevanti.

5.3 Angle Based Outlier Detector

Un algoritmo che introduce un nuovo paradigma per definire il grado di anomalia di un dato è l'*angle based outlier detector* (ABOD)[33].

Come si intuisce dal nome, questo metodo fa affidamento sugli angoli, più specificatamente sulla varianza degli angoli. L'idea di base è piuttosto intuitiva: se si considera un'osservazione normale, è molto probabile essa sia contenuta in un raggruppamento composto da altri dati simili e quindi considerando gli angoli che si ottengono da questa osservazione verso le altre, si ottiene un valore

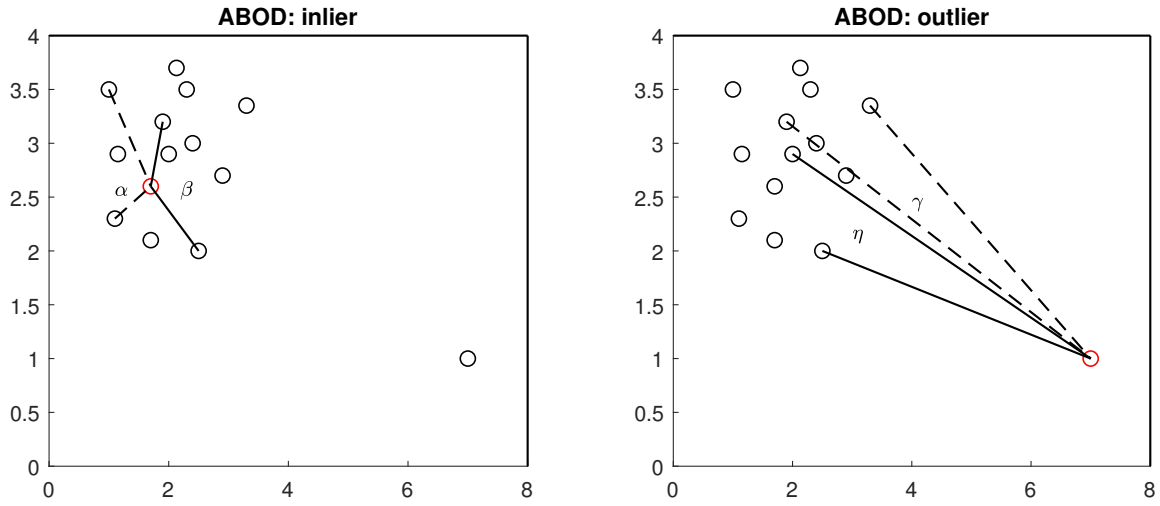


Figura 5.2: Intuizione angle based outlier detector

di essi in uno spettro molto ampio, e quindi una varianza molto elevata; al contrario le osservazioni anomale saranno presumibilmente distanti da questo cluster e dunque gli angoli da questo dato verso i dati del cluster saranno simili ottenendo dunque una varianza sensibilmente più bassa. In aggiunta la varianza è pesata dalla lunghezza dei vettori dei punti considerati, in questo modo l'angolo sarà meno influente se i punti considerati sono distanti dall'osservazione in esame.

Da un punto di vista matematico si definisce un fattore, noto come *angle based outlier factor* (ABOF) per ogni istanza del dataset. L'ABOF($\bar{A}$), dove $\bar{A}$ è un punto del dataset, è dato dalla varianza su tutti gli angoli ottenuti dal punto $\bar{A}$ e tutte le coppie di punti $\bar{B}$ e $\bar{C}$ nel dataset, pesati dalla distanza dei punti stessi:

$$ABOF(\bar{A}) = VAR_{\bar{B},\bar{C}\in\mathcal{D}}\left(\frac{\left\langle \overline{AB}, \overline{AC} \right\rangle}{||\overline{AB}||^2 \cdot ||\overline{AC}||^2}\right) \tag{5.1}$$

Dopo aver assegnato il fattore ABOF ad ogni punto del dataset, l'algoritmo ABOD ritorna la lista di punti ordinati a seconda del valore di questo fattore. I punti classificati in cima alla lista saranno con tutta probabilità degli outlier, invece quelli con il rank più basso saranno degli inlier.

L'algoritmo ABOD è in grado di lavorare ottimamente anche in alte dimensioni a differenza di altri algoritmi che perdono di accuratezza; oltre a questo, permette di avere una diversa classificazione per i dati: riesce a segnalare quando si tratta di un'osservazione al confine (quelle osservazioni che vengono subito dopo le anomalie nella lista) rispetto a una situata all'interno di un cluster , cosa

che gli altri algoritmi non determinano.

In ogni caso il vantaggio principale di questo algoritmo è l'essere completamento libero dalla dipendenza da parametri che devono essere calibrati; il lato negativo però è che è un algoritmo dispendioso da un punto di vista computazionale e non riesce a catturare in maniera adeguata strutture complesse. Per ridurre il tempo di calcolo si può considerare una versione approssimata di questo algoritmo in cui vengono selezionate solo le K osservazioni più vicine al punto in esame.

5.4 Local Outlier Factor

Il terzo metodo studiato è stato il *local outlier factor* (LOF)[34]. Questa tecnica basata sulla densità di distribuzione dei dati assegna ad ogni oggetto un grado che stabilisce quanto quell'oggetto sia un outlier; questo fattore viene chiamato LOF di un'osservazione. Questo algoritmo si caratterizza per il fatto di assegnare ad ogni osservazione uno score di quanto quello specifico dato possa essere un'anomalia

Questo algoritmo si applica solo localmente in quanto dipende da quanto sia isolata l'osservazione in esame rispetto ai propri vicini. Unisce dunque il concetto dei K vicini più prossimi a quello della densità dei cluster (è infatti un metodo distribution-based).

Per una descrizione più analitica di questo fattore, bisogna prima definire i seguenti parametri:

- *K-distanza* $d_k(p)$: è la distanza dall'osservazione p del K-esima istanza più vicina.
- *K-distanza vicinato* $N_k(p)$: è l'insieme dei vicini che hanno una distanza inferiore rispetto a $d_k(p)$.
- *distanza di raggiungibilità* $rd_k(p, o)$: è data dal massimo tra $d_k(o)$ e la distanza $d(p, o)$.

Si definisce poi una misura, nota come *densità di raggiungibilità locale* $lrd(p)$, che individua quanto siano densamente abitati i dintorni del punto. Questa è data dall'inverso della media distanza di raggiungibilità basata sui K punti più prossimi:

$$lrd_k(p) = \left(\frac{\sum_{o \in N_{k(p)}} rd_k(p, o)}{|N_k(p)|} \right)^{-1} \tag{5.2}$$

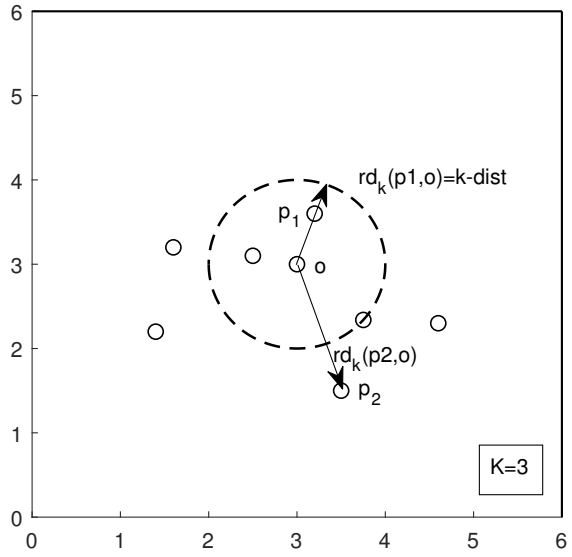


Figura 5.3: Distanza di raggiungibilità

Da cui infine si può ricavare il valore LOF per una misura p dato da:

$$LOF_k(p) = \frac{\sum_{o \in N_{k(p)}} \frac{lrd_k(o)}{lrd_k(p)}}{|N_k(p)|} \tag{5.3}$$

il quale tenta di catturare il grado di anomalia di quel dato: un valore di circa 1 per il LOF(p) rappresenta un inlier, al contrario un valore di circa 0 testimonia la presenza di un outlier.

Tra le proprietà più utili per LOF c'è sicuramente il fatto di essere un algoritmo per il quale si può trovare un limite superiore e un limite inferiore. Infatti da questo intervallo si può stabilire un range di valori per k, proprio perché a differenza del metodo precedente le performance di questo algoritmo dipendono dal numero di vicini che si vuole considerare.

LOF è uno dei metodi più popolari per rilevare un'anomalia soprattutto per la sua facile implementazione e per le sue superiori abilità di riconoscimento rispetto ad algoritmi di base come può essere K-NN. Inoltre fornire uno score dell'anomalia per tutti i punti può essere un punto di forza notevole, specialmente nel caso di situazione più complesse e delicate.

D'altro canto come algoritmo risulta abbastanza inefficiente con scenari a bassa densità e presenta un costo computazionale assai elevato che talvolta ne pregiudica l'utilizzo principalmente in situazioni real-time (in questi casi si preferisce usare versioni più complesse di questo algoritmo che prevedono l'uso di iterazioni)

5.5 Locally Selective Combination Detector

L'ultima soluzione proposta è essenzialmente una combinazione delle precedenti tecniche sfruttate in un unico algoritmo. Questo metodo sviluppato recentemente viene chiamato *locally selective in parallel outlier ensembles* (LSCP)[35].

Nel problema non-supervisionato l'assenza di un'etichettatura del problema rende la selezione di un metodo di anomaly detection un compito arduo, in particolare la scelta di uno o dell'altro metodo può influenzare notevolmente l'accuratezza e la stabilità.

Per tal motivo questo metodo affronta il problema selezionando una regione locale attorno a un'istanza di test dove, utilizzando i vicini più prossimi, vengono applicati contemporaneamente più metodi. Quelli più performanti vengono selezionati e combinati per decretare l'output finale del modello.

Spesso infatti i singoli approcci sono suscettibili a rilevare con un tasso elevato sia falsi-positivi che falsi negativi, l'idea è quella di utilizzare più metodi contemporaneamente con il fine di migliorare l'accuratezza e l'affidabilità del risultato. É importante ottenere un modello che sia robusto, e che riesca di volta in volta a adeguarsi al caso specifico: infatti può accadere che le buone performance di un riconoscitore siano limitate a causa della presenza di un riconoscitore differente che ha risultati negativi.

L'algoritmo per ogni istanza di test prevede una serie di passi:

1. Viene definita la regione locale costituita dai k punti più vicini con il sottospazio delle features più frequente tra quelli che vengono scelti casualmente .

2. Usando la regione locale, vengono calcolate le performance di ogni metodo nella lista degli algoritmi.

3. Vengono selezionati solo i metodi competenti per la determinata osservazione di test.

4. Lo score finale sarà dato dalla media degli score di anomalia di ogni metodo rimasto.

I vantaggi di questo algoritmo sono chiari e notevoli, in particolare una buona selezione della combinazione di algoritmi renderebbe il modello assai accurato nella previsione. Tuttavia bisogna fare i conti con dei limiti ben precisi: primo la regione locale viene definita tramite i k vicini e ciò non è

ideale soprattutto per l'alto tasso di complessità temporale che se ne ricava; secondo per le limitate performance che si ottengono su un dataset composto da numerose features irrilevanti.

5.6 Risultati

Con l'applicazione degli algoritmi precedentemente descritti sul dataset delle lavorazioni si è cercato di verificare quale algoritmo fosse più idoneo per il riconoscimento di lavorazioni anomale. Come si è già accennato per valutare le performance degli algoritmi si è deciso di utilizzare la scorciatoia della conoscenza della provenienza del dato in esame per rendere visibile il confronto.

La prima operazione compiuta è stata quella dell'eliminazione delle features meno rilevanti. Dato anche che il numero di campioni nel dataset è diminuito notevolmente (DATASAET 2), il rischio era di ottenere algoritmi che non fossero in grado di estrapolare il carattere generale di un algoritmo da così tante features, per cui la fase di eliminazione degli attributi meno rilevanti è stata fondamentale e necessaria.
In principio la scelta è ricaduta sull'applicazione della principal component analysis, come è previsto per la riduzione nella maggior parte dei dataset non supervisionati. Tuttavia gli scarsi risultati e l'impossibilità di tenere traccia delle features più significative hanno reso obbligatorio un cambio di strategia. Non è stato possibile utilizzare neanche la indipendent component analysis poiché nelle semplificazioni degli attributi il carattere anomalo veniva assai mitigato o addirittura rimosso ottenendo così performance molto scadenti.
Si è scelto dunque di utilizzare un sottoinsieme delle features di partenza che avesse la maggiore descrizione possibile del contenuto del dato con il minor numero di attributi possibile. Il procedimento ha previsto l'utilizzo del meotodo sequential backward selection, supportato da un'analisi grafica. Il numero di attributi rimasto è stato pari a 9:

Lista 2R: *Media, Energia, Fattore di Margine, filt_RMS, filt_Media e fft_Picco-Picco per il Montronix; Energia, RMS, filt_RMS, per il PCB piezotronics*

Dai seguenti grafici (Figura 5.4) si può visualizzare come siano state selezionate le features più rilevanti e che mantengano l'informazione sull'anomalia (si sfrutta l'informazione a priori della pro-

	Campioni	Features
DATASET 2R:	232	9

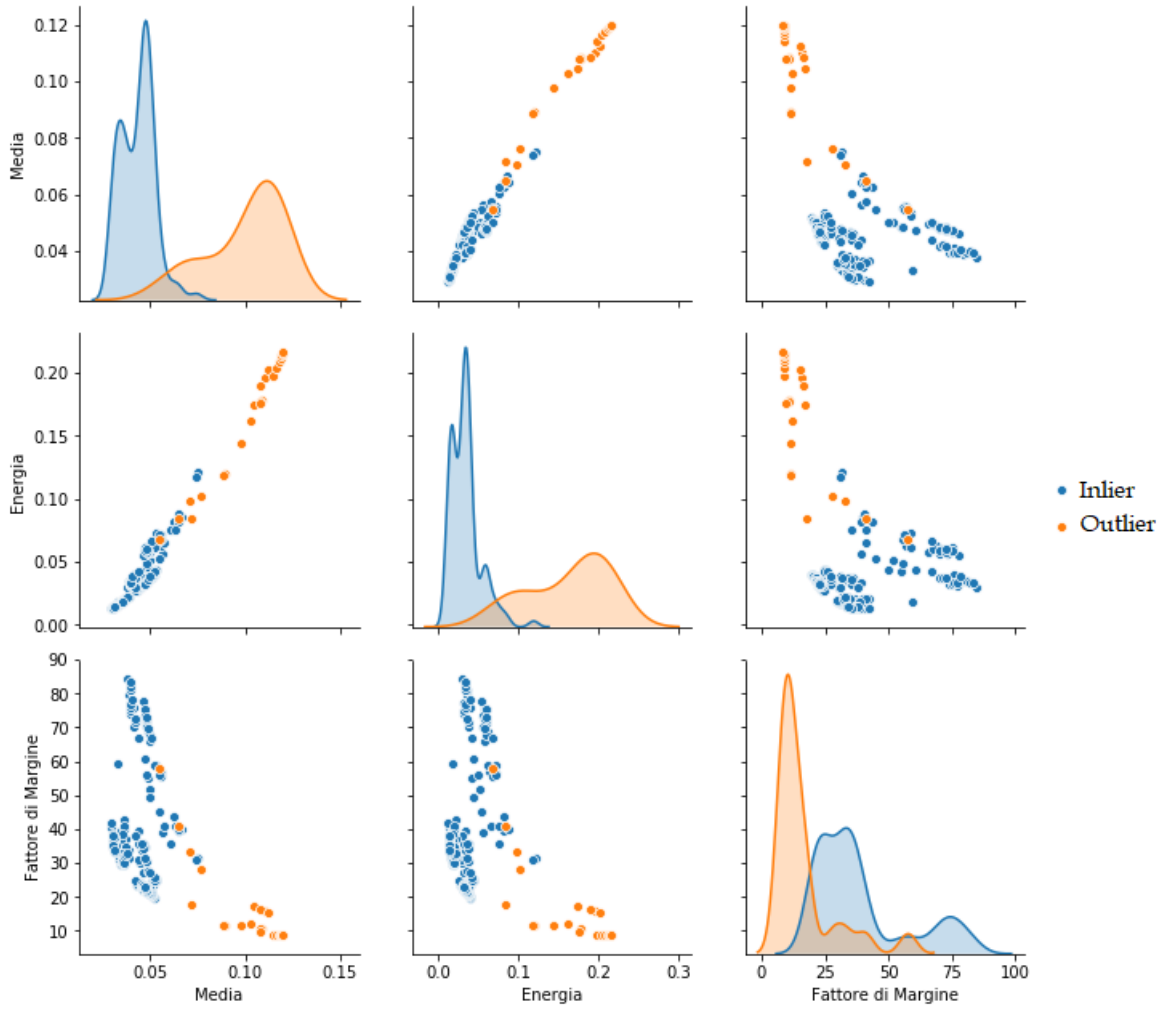


Figura 5.4: Confronto features significative

venienza del dato per verificare se la scelta delle features più significative è corretta). Questo tipo di grafici mostra le relazioni a coppie tra differenti features in un dataset.

Per quanto concerne la valutazione dell'accuratezza e dell'affidabilità degli algoritmi di anomaly detection si sono utilizzate due metriche principali: la *precision* e la *receiver operating characteristic*(ROC). La prima metrica riguarda la probabilità che un campione valutato come anomalia sia effettivamente un'anomalia; questo valore è dato dal rapporto tra i veri-positivi e la somma tra veri-positivi e falsi-positivi (quando si identificano erroneamente delle anomalie, anche detti falsi allarmi). La ROC invece è una curva di prestazione utile per confrontare diversi classificatori; lungo le ascisse

	Train		Test	
	AUC-ROC	precision	AUC-ROC	precision
iForest:	0.872	0.750	0.812	0.677
ABOD:	0.705	0.584	0.680	0.449
LOF:	0.886	0.762	0.819	0.692
LSCP:	0.912	0.813	0.894	0.747

Tabella 5.1: Risultati algoritmi di anomaly detection

è presente il numero dei falsi-positivi percentuali mentre in ordinata è presente la percentuale di riconoscimenti corretti. Un buon classificatore deve essere migliore del classificatore casuale, ovvero la retta che congiunge i punti $(0, 0)$ e $(1, 1)$ del grafico. Un metodo per valutare l'accuratezza dal grafico è calcolare l'area di questa receiver operating characteristic (AUC-ROC).

Il primo algoritmo testato è stato l'isolation forest detector con un numero di alberi considerati pari a 100: riesce bene a distinguere le anomalie nel dataset anche se è un po' troppo alto il tasso di falsi allarmi. I valori ottenuti per il set di test sono di 0.677 per la precision, e di 0.812 per l'area sotto la curva ROC, come si può rilevare dalla Tabella 5.1.

Lo stesso tipo di problema affligge anche l'algoritmo LOF. Considerando un numero di vicini pari a 5, l'algoritmo ottiene performance molto simili al precedente. Si riescono a identificare bene le lavorazioni anomale ma troppo spesso si fermerebbe l'impianto per lavorazioni particolari ma che sono essenzialmente corrette.

Per quanto riguarda l'algoritmo angle based outlier detector le performance ottenute sono state particolarmente scarse. Per entrambe le metriche i valori sono stati i peggiori rispetto a tutti gli altri algoritmi. La struttura abbastanza complessa di un dataset del genere non favorisce l'implementazione di un algoritmo poco robusto come è ABOD, nonostante abbia il vantaggio di essere libero da parametri da settare.

L'algoritmo, tra quelli considerati, che ha l'accuratezza migliore è LSCP. Selezionando nella list di detectors algoritmi LOF che hanno un parametro K diverso tra loro si riesce a combinare ottimamente le proprietà ed aumentare l'affidabilità complessiva. Nel nostro caso si sono selezionati cinque valori per K: $3, 5, 7, 9, 11$.

La precisione di questo metodo risulta ancora un po' troppo bassa per essere realmente implementata in un contesto reale come unica soluzione, invece potrebbe essere utile come base di appoggio per fornire suggerimenti agli operatoti.

Soluzioni future potrebbero prevedere di migliorare le performance complessive utilizzando metodi completamente diversi nella lista da utilizzare per LSCP, addirittura mescolando i diversi metodi precedentemente elencati.

Capitolo 6

Conclusione

Si possono individuare tre punti chiave per quanto riguarda i risultati trovati in questo progetto.
Il primo punto riguarda la posizione ottima da cui descrivere le lavorazioni in esame: la posizione B riesce a cogliere in maniera maggiormente distinta il fenomeno delle vibrazioni e pertanto si consiglia di utilizzare questa posizione qualora si voglia analizzare un comportamento simile.
Il secondo e terzo punto si riferiscono ai migliori algoritmi trovati per il caso supervisionato e per il caso non supervisionato. Nell'approccio supervisionato si è cercato di stabilire quale fosse il metodo migliore che riuscisse a discernere quando una lavorazione fosse impropria (o non corretta) e quando al contrario fosse una lavorazione legittima e senza errori; dallo studio è emerso chiaramente che la support vector machine con kernel gaussiano, principalmente a causa del suo funzionamento non lineare, fosse la tecnica migliore da questo punto di vista. Per l'approccio non supervisionato l'algoritmo che meglio si adatta per trovare le lavorazioni anomale in un dataset composto prelevantemente da lavorazioni corrette è il locally selective combination in parallel outlier detector. La combinazione di diversi metodi permette una potenza e un'accuratezza superiore che talvolta non possono essere raggiunte con l'utilizzo di un solo modello. Le performance trovate per questo algoritmo non sono tuttavia sufficienti per un'applicazione reale in quanto il rischio di identificazione di falsi positivi è troppo elevato per questa situazione, ma sicuramente potrebbe essere una base di appoggio utile per gli operatori.
Questo progetto ha voluto dimostrare la fattibilità dell'applicazione del machine learning al contesto delle lavorazione delle macchine utensili.
Per migliorare e continuare lo studio sarebbe opportuno raccogliere e descrivere il maggior numero di

casi possibili, in modo da ottenere un dataset maggiormente robusto e descrittivo del tipo di lavorazioni che avvengono. Con un aumento del numero di campioni sarebbe poi possibile utilizzare altre tecniche di cui testare poi i risultati; tra queste tecniche sicuramente le reti neurali e gli autoencoder per l'anomaly detection, con i quali si potrebbero ottenere risultati incoraggianti.

Altri possibili sviluppi futuri includono il tentativo di studiare direttamente il dato di vibrazione senza prima elaborarlo con conseguente estrazione di informazioni al fine di compiere analisi predittive.

La caratterizzazione dello stato di salute di una macchina utensile, in particolare il calcolo della quantità di vita rimanente dai dati di vibrazioni, rimane uno degli aspetti più importanti che potrebbe essere maggiormente analizzato in uno sviluppo ulteriore di questo progetto.

L'obiettivo finale sarà quello di sviluppare un sistema completamente automatizzato che sia in grado di prevedere l'andamento di una lavorazione e di attuare le modifiche sui parametri o gli avvertimenti necessari per rendere la lavorazione sempre corretta e qualitativa.

Appendice

Appendice A

- Specifiche Ultrix.

	ULTRIX 800	*ULTRIX 1000*	*ULTRIX 1200*
Assi interpolati:	5	5	5
Corsa asse "X" [mm]:	800	1150	1700
Corsa asse "Y" [mm]:	900	1000	1600
Corsa asse "Z" [mm]:	600	700	1000
Diametro tavola tornitura [mm]:	700	1000	1200
Max dimensione tornibile [mm]:	800	1000	1200
Velocità di rapido assi "X" "Y" [m/min]:	60	60	50
Velocità di rapido assi "Z" [m/min]:	40	40	40
Rotazione asse A:	$-30^\circ \div +110^\circ$	$-30 \div +120^\circ$	$\pm 120^\circ$
Rotazione asse A:	continuo	continuo	continuo
Coppia mandrino fresatura [Nm]:	$137/100 \div 22/16$	$480/300 \div 137/100$	$480/300 \div 137/100$
Velocità di rotazione [rpm]:	$18000 \div 40000$	$14000 \div 28000$	$14000 \div 28000$
Attacco utensile:	HSK-A63	HSK-A63 o A100	HSK-A63 o A100
Velocità di rapido asse C [rpm]:	100	60	60
Velocità di rapido asse C [rpm]:	100	60	60

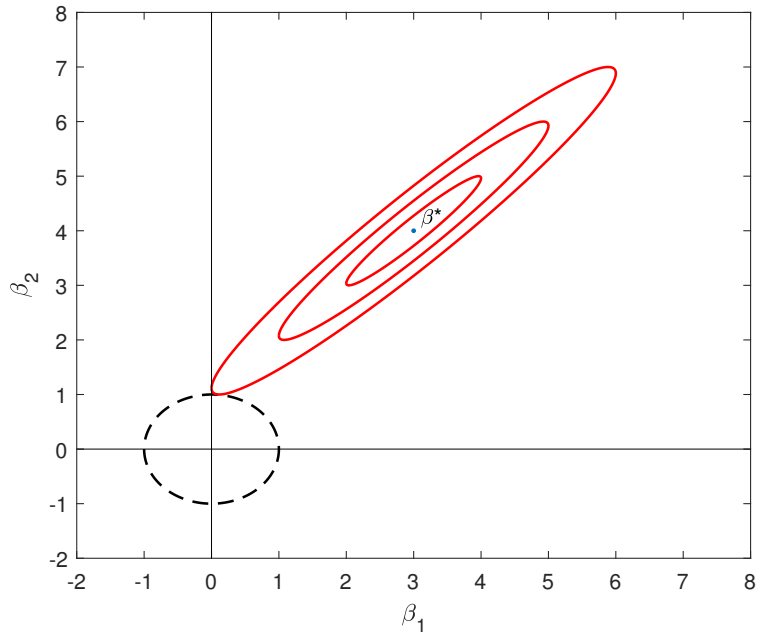
Appendice B

- Specifiche Beckhoff EL3632.

	Beckhoff EL3632
Numero di ingressi:	2
Range di misura:	preset $\pm$ 5V
Tensione fornita:	24 V_{DC} via contatti di potenza
Monitoraggio dello stato del sensore:	si, attraverso il monitoraggio del bias della tensione
Massimo campionamento:	50 kSamples/s
Risoluzione:	16 bit
Errore di misura:	0.5%
Consumo corrente via E-Bus:	circa 220mA
Configurazione:	Twincat System Manager
Range di temperatura:	$0°C ... + 55°C$
Peso:	65g
Classe di protezione:	IP20

Appendice C

- **Ridge Regression**



$$minimize_\beta = \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \quad subject \quad to \quad \sum_{j=1}^{p} \beta_j^2 < s \tag{1}$$

Il cerchio rappresenta il vincolo nel caso di regolarizzazione L2 a due feature.

Diversamente dal caso del lasso l'intersezione tra le ellissi e il cerchio non necessariamente avverrà su un asse e quindi i coefficienti stimati saranno esclusivamente non zero.

L2 non è un algoritmo robusto nella ricerca delle anomalie in quanto la regolarizzazione tenta di sopprimere le differenze penalizzando i pesi dissimili. In generale ridge regression ha performance migliori quando tutte le features hanno un influenza nell'output finale e con i pesi più o meno della stessa grandezza.

Ricapitolando, la regolarizzazione penalizza la somma dei pesi al quadrato, non ha una soluzione sparsa, non fa feature selection e non è robusta agli outlier, nonostante ciò ha una migliore accuratezza quando l'output è una funzione di tutte le variabili ed è in grado di apprendere pattern complessi dei dati.

Appendice D

- **Random Forest**

Le random forest (o foreste casuali) si basano su un tipo di apprendimento conosciuto come *apprendimento ensemble* (o apprendimento d'insieme) che consiste nel combinare più modelli per ottenerne uno più robusto che abbia una migliore generalizzazione e un rischio minore di avere overfitting dei dati.
La foresta casuale è un apprendimento ensemble che mette insieme più alberi decisionali. Si può riassumere il comportamento di questo modello in cinque step:

1. Selezionare un numero di alberi per la foresta casuale
2. Selezionare un numero di esempi a caso dal dataset
3. Addestrare l'albero su questo sottoinsieme di dati
4. Ripetere step 2 e 3 fino ad avere costruito tutti gli alberi
5. Per eseguire una predizione, si utilizzano tutti gli alberi e poi si fa una media delle varie predizioni.

Solitamente un numero maggiore di alberi conduce a un modello migliore ma richiede anche un maggior impiego di risorse di calcolo.

Indice delle Immagini

Bibliografia

- [1] Petrucci G., *Cuscinetti a rotolamento*, Lezioni di Costruzione di Macchine, Università degli Studi di Palermo

- [2] SKF, *Durata di Base dei Cuscinetti* www.skf.com/it/products/bearings-units-housings/principles/bearing-selection-process/bearing-size/size-selection-based-on-rating-life/bearing-rating-life/index.html

- [3] NTN, *Ball and Roller Bearings*, https://www.ntn-snr.com/sites/default/files/2017-05/ntn-ball-and-roller-bearings.en.pdf, 2015

- [4] Chen Z., Deng S., *Deep neural networks-based rolling bearing fault diagnosis*, Microelectronics Reliability, March 2017

- [5] He M., *A Deep Learning-based Approach for Fault Diagnosis of Roller Element Bearings*, IEEE Transactions on Industry Applications, January 2017

- [6] Sikora A., *Detection of bearing damage by statistic vibration analysis*, IOP Conference Series: Materials Science and Engineering, April 2016

- [7] Tosi B., *Misure di Vibrazioni*, Politecnico di Milano, 2014

- [8] *Misura delle Vibrazioni*, Università di Catania, Diim

- [9] Saidi, Lotfi, *Wind turbine high-speed shaft bearings health prognosis through a spectral Kurtosis-derived indices and SVR*, Applied Acoustics 120, 2017

- [10] Coble, Baalis J., *Merging data sources to predict remaining useful life–an automated method to identify prognostic parameters*, 2010.

- [11] Mian G., *Elaborazione numerica dei segnali*, Università degli studi di Padova, 2010

- [12] Graney B., Starry K., *Rolling Element Bearing Analysis*, Materials Evolution Vol.1 pag.78-85, The American Society for Nondestructive Testing Inc, 2011

- [13] Saruhan H., Sarademir2 S.,Çiçek A., Uygur I., *Vibration Analysis of Rolling Element Bearing Defects*, Journal of Applied Research and Technology, 2014

- [14] Mazzoldi P., Nigro M., Voci C., *Fisica I*, Edises, 2014

- [15] Mazzoldi P., Nigro M., Voci C., *Fisica II*, Edises, 2014

- [16] Capineri L., *Elettronica dei Sistemi Analogici e Sensori*, Università degli studi di Firenze, Ingegneria Elettronica, 2015

- [17] Goodfellow I., Bengio Y., Courville A., *Deep Learning*, Mit Press, 2016

- [18] Pavon M., *Appunti Segnali e Sistemi*, Università degli studi di Padova, Ingegneria dell'informazione, 2016

- [19] Persson P., Strang G., Rosenthal J., Gilliam S.,*Smoothing by Savitzky-Golay and Legendre Filters*, Mathematical Systems Theory in Biology, Communications, Computation, and Finance, pag 301-315, Springer, 2003

- [20] James G., Witten D., Hastie T., Tibshirani R., *An introduction to statistical learning*, Springer, 2013

- [21] *Dimensionality Reduction algorithms: Strenghs and Weakness*, https://elitedatascience.com/dimensionality-reduction-algorithms/feature-selection, May 2017

- [22] Shalev-Shwartz S., Shai B., *Understanding Machine Learnings: from theory to algorithms*, Cambridge University Press, 2014

- [23] Nagpal A., *L1 and L2 Regularization Methods*, https://towardsdatascience.com/l1-and-l2-regularization-methods, October 2017

- [24] Hastie T., Tibshirani R. Friedman J., *The Elements of Statistical Learning*, Springer, Second Edition, 2008

- [25] Bishop C., *Pattern Recognition and Machine Learning*, Springer, 2006.

- [26] QuantStart, *Support Vector Machine: A guide for beginners*, https://www.quantstart.com/articles/Support-Vector-Machines-A-Guide-for-Beginners

- [27] Vandin F., *Unsupervised Learning*, Machine Learning, Università degli studi di Padova, Ingegneria delle Telecomunicazioni, Dicembre 2016 - [28] Vegard Flovik, *How to use machine learning for anomaly detection and condition monitoring* https://towardsdatascience.com/ how-to-use-machine-learning-for-anomaly-detection-and-condition-monitoring-6742f82900d7, December 2018

- [29] Kotu V., *Anomaly Detection* Predictive Analytics and Data Mining, pag 329-345, 2015

- [30] Yildirim A., Ozdogan C., Watson D., *Parallel Data Reduction Techniques for Big Datasets*, 2014

- [31] Langlois D., Chartier S., Gosselin D., *An Introduction to Indipendent Component Analysis: InfoMax and FastICA algorithms*, Tutorials in Quantitative Methods for Psychology, 2010

- [32] Liu F.T., Ting K.M., Zhou Z.H., *Isolation Forest*, Eight IEEE International Conference on Data Mining, 2008.

- [33] Kriegel H., Schubert M., Zimek A., *Angle-Based Outlier Detection in High Dimensional Data*, International Conference on Knowledge Discovery & Data Mining, Las Vegas, 2008

- [34] Breunig M., Kriegel H.P., Ng R., Sandert J., *LOF: Identifying Density-Based Local Outliers*, International Conf. On Management of Data, Dallas, TX, 2000

- [35] Zhao Y., Nasrullah Z., Li Z., Hryniewicki M., *Locally Selective Combination in Parallel Outlier Ensembles*, SIAM International Conference on Data Mining, Canada, 2019