

UNIVERSITÀ POLITECNICA DELLE MARCHE

FACOLTÀ DI INGEGNERIA



*Corso di Laurea Magistrale in
Ingegneria Informatica e dell'Automazione*

PROGETTO D1 - PHM ASIA PACIFIC 2023
***Modulo di diagnosi dei dati generati dal simulatore di un
sistema di propulsione***

Docente:
PROF. FREDDI ALESSANDRO

Studenti:
CIUFFREDA SILVIA
SIROCCHI CLAUDIO
PISTAGNESI LAURA

ANNO ACCADEMICO 2023-2024

Indice

1	Stato dell'arte	4
2	Materiali e metodi	7
2.1	Dataset	7
2.2	Metodologia adottata	9
2.2.1	Matlab	9
2.2.2	Metodologia di svolgimento della challenge	10
3	Task 1	12
3.1	Generazione delle feature	12
3.1.1	Selezione della Frame Policy	12
3.1.2	Calcolo dello Spettro di Potenza	13
3.1.3	Feature diagnostiche nel dominio delle frequenze	17
3.1.4	Feature diagnostiche nel dominio del tempo	18
3.1.5	Selezione delle feature	18
3.2	Addestramento modello di classificazione binario	18
4	Task 2	21
4.1	Soluzioni individuate	21
4.2	Classificatori one-class in cascata	22
4.3	Classificatore one-class e binario	23
4.3.1	Addestramento del modello One-Class SVM	23
4.3.2	Addestramento del modello binario	24
5	Task 3	26
5.1	Generazione delle feature	26
5.2	Addestramento modello di classificazione multiclasse	27
6	Task 4	29
6.1	Generazione delle feature	29
6.2	Addestramento modelli	29

INDICE	3
7 Task 5	31
7.0.1 Generazione delle feature	31
7.0.2 Addestramento dei modelli	32
8 Testing	34
8.1 Task 1	35
8.2 Task 2	36
8.2.1 Classificatori one-class in cascata	36
8.2.2 Classificatore one-class e binario	38
8.2.3 Confronto risultati	40
8.3 Task 3	41
8.4 Task 4	42
8.5 Task 5	44
9 Metrica di valutazione	45

Capitolo 1

Stato dell'arte

I dati di telemetria che possono essere acquisiti in orbita sono ridotti a causa della limitazione nell'installazione di sensori e della capacità di downlink. Per ovviare a ciò, la Japan Aerospace Exploration Agency (JAXA) ha sviluppato un simulatore numerico per prevedere la risposta dinamica di un sistema di propulsione di un veicolo spaziale con elevata precisione per generare un set di dati che copra le condizioni normali e tutti gli scenari di guasto previsti nelle apparecchiature reali.

L'obiettivo della challenge è quello di migliorare la tecnologia PHM per i veicoli spaziali di prossima generazione rendendola in grado di diagnosticare condizioni normali, anomalie dovute a bolle, guasti alle elettrovalvole e casi anomali sconosciuti utilizzando i dati generati dal simulatore.

Alla fine di avere maggior chiarezza si vuole fornire una panoramica delle caratteristiche tecniche del simulatore e delle informazioni da esso generate.

Lo scenario di un esperimento di un sistema di propulsione è mostrato in Fig.1.1.

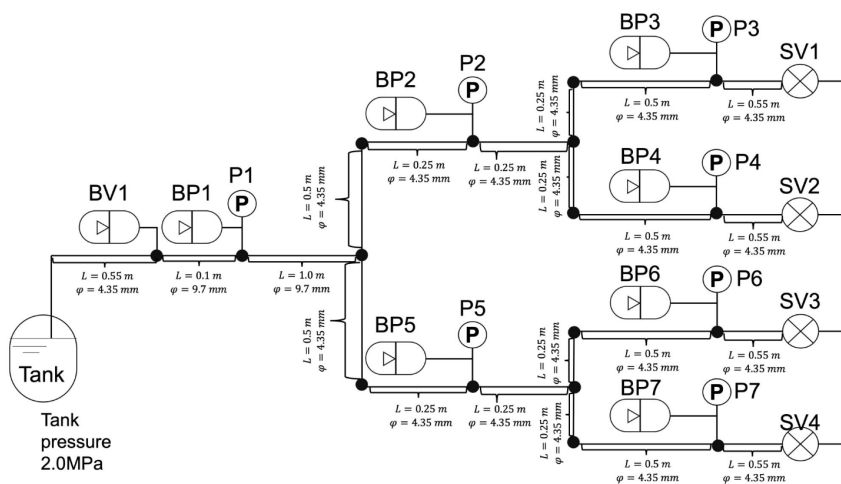


Figura 1.1: Schema del sistema di propulsione sperimentale

Il fluido di lavoro è acqua pressurizzata a 2 MPa e viene scaricato attraverso quattro elettrovalvole (SV1 – SV4), simulando così i propulsori. I sensori di pressione (P1 – P8) registrano i dati delle serie temporali a una frequenza di campionamento di 1 kHz per un intervallo temporale da 0 a 1200 ms. Durante l’apertura e la chiusura delle elettrovalvole, si osservano fluttuazioni di pressione causate da colpi d’aria seguiti da modalità acustiche all’interno del sistema di propulsione.

Da tale sistema è possibile generare un segnale rappresentante un tipico profilo di pressione. Tale profilo è raffigurato in Fig. 1.2.

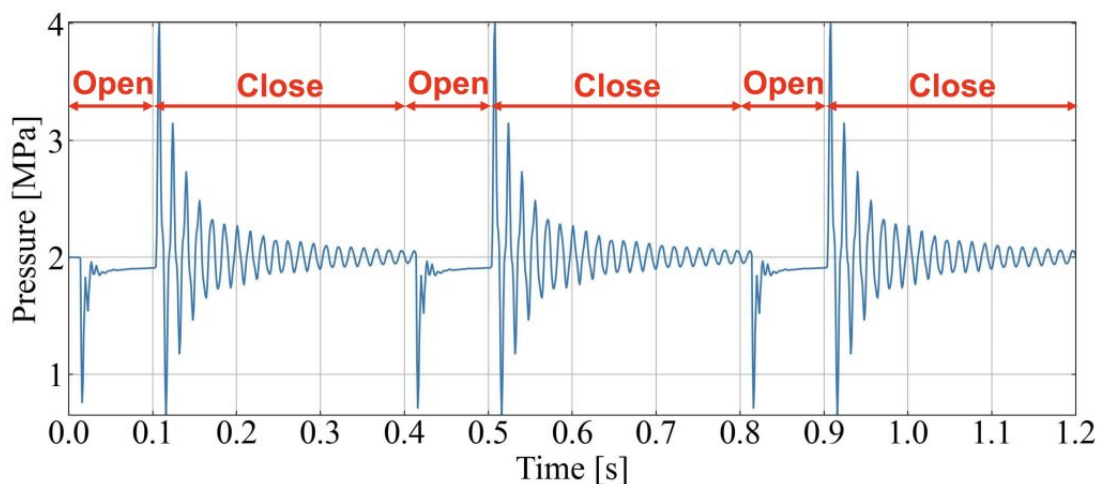


Figura 1.2: Profilo di pressione tipico estratto dal modello

L’elettrovalvola si apre a 100 ms e si chiude a 300 ms. Per tenere conto delle differenze individuali delle elettrovalvole che possono presentarsi nelle apparecchiature reali, il movimento della valvola presenta un’incertezza di 1 ms. I tempi di apertura e chiusura rimangono a 400 ms, anche con l’incertezza (ad esempio, apertura per 99,7 ms e chiusura per 300,3 ms). Questa sequenza viene eseguita tre volte consecutivamente, per un totale di misurazione di 1200 ms.

In questa competizione si considerano tre tipologie di anomalie:

- Anomalia dovuta a bolle: Durante il funzionamento reale di un veicolo spaziale, occasionalmente compaiono bolle d’aria nelle tubazioni. La presenza di bolle modifica la velocità del suono, causando lievi variazioni nelle fluttuazioni di pressione. È auspicabile rilevare la presenza di bolle e la loro posizione. Esistono otto possibili posizioni, da BV1 e da BP1 a BP7 come mostrato in Figura 1. La quantità di bolle presente nel sistema di propulsione è costante in tutti i casi per semplicità.
- Guasti alle elettrovalvole: Questa è una delle principali modalità di guasto nei sistemi di propulsione spaziali. È necessario individuare le elettrovalvole

guaste e il loro grado di apertura. Le elettrovalvole si aprono e chiudono con un rapporto di apertura rispettivamente del 100 % e dello 0%. In caso di guasto, le elettrovalvole si aprono con un grado compreso tra lo 0% e il 100%, causando una riduzione del volume di fluido che le attraversa.

- Anomalia sconosciuta: Nel funzionamento reale possono verificarsi anomalie o guasti completamente imprevisi e sconosciuti. È inoltre necessario distinguere le anomalie sconosciute senza confonderle con anomalie e guasti noti. Alcuni casi di anomalie o guasti sconosciuti sono inseriti nei dati di test. Anche la loro identificazione rientra tra i compiti di questa competizione.

L'obiettivo è quello di realizzare un sistema a cascata che svolga i seguenti passaggi :

- Determinare i valori normale e anormali del dataset di test;
- Dei valori anormali identificati stabilire in che tipologia di anomalia rientrano;
- Per le anomalie classificate come tipo bolla determinare in quale posizione queste sono avvenute.
- Per le anomalie classificate come errore di apertura delle elettrovalvole determinare quale delle 4 ha avuto problemi nella fase di apertura.
- Per la valvola con errore di apertura individuata determinare la percentuale di apertura della stessa.

La challenge mette a disposizione sia il dataset contenente tutti i dati necessari per procedere con le operazioni di estrazione delle feature, generazione dei modelli e dati per eseguire il testing. Sono inoltre specificate le metriche per valutare il risultato della challenge, le modalità di invio dei risultati e i tool consigliati per lo svolgimento della stessa.

Capitolo 2

Materiali e metodi

Nel seguente capitolo vengono si illustra il materiale a disposizione per lo svolgimento della challenge e gli strumenti e le metodologie adottate per quest'ultima. Nei capitoli successivi sarà poi possibile apprezzare i passaggi tecnici adottati nello specifico per ogni task da sviluppare per la challenge.

2.1 Dataset

Il dataset messo a disposizione per lo svolgimento della challenge è strutturato così come mostrato in Fig.2.1.

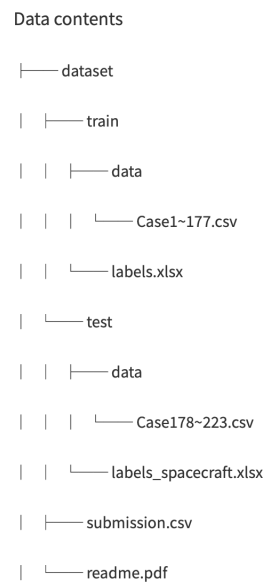


Figura 2.1: Struttura del dataset

Il dataset è composto da dati di addestramento e dati per il testing.

Più nello specifico si ha una suddivisione di questi ultimi che sono generati per 4 spacecraft. Ogni rilevazione effettuata su uno spacecraft viene identificata dalla dicitura "Case" seguita dal numero della rilevazione al fine di poterla individuare univocamente. Ogni Case corrisponde a uno dei veicoli e contiene i dati registrati dai sensori in un intervallo di 1200ms. All'interno di ogni case troviamo tre fasi di apertura e tre fasi di chiusura delle valvole. I dati per addestramento e testing sono divisi come di seguito.

Dati di addestramento

- Numero di dati normali per il Veicolo Spaziale-1: 35
- Numero di dati normali per il Veicolo Spaziale-2: 35
- Numero di dati normali per il Veicolo Spaziale-3: 35
- Numero di dati anomali per il Veicolo Spaziale-1: 24
- Numero di dati anomali per il Veicolo Spaziale-2: 24
- Numero di dati anomali per il Veicolo Spaziale-3: 24

I dati di addestramento vanno dal Case 1 al Case 177 e all'interno del file labels.xls vengono riportate tutte le informazioni e le specifiche per ogni caso e per ogni task.

Dati di testing

- Numero di dati per il Veicolo Spaziale-1: 23
- Numero di dati per il Veicolo Spaziale-4: 23

I dati di addestramento vanno dal Case 178 al Case 223 e all'interno del file labels.xls in questo caso vengono riportati esclusivamente i valori in riferimento a spacecraft e Case, non sono presenti etichette.

Le valvole solenoidi presentano differenze individuali che possono essere dovute al momento di apertura e chiusura, ai dati temporali acquisiti mediante i sensori e le componenti del sistema di propulsione del veicolo spaziale. Questo implica di fatto che ci sono differenze individuali nei veicoli spaziali.

Il dataset a nostra disposizione contiene i dati di 4 spacecraft, come già detto in precedenza. I risultati per tre di essi, dal N. 1 al N. 3, sono inclusi nei dati di addestramento, ma i dati di test sono composti dai dati per lo spacecraft N. 1 e per il N. 4. Il fatto che essi presentino delle differenze individuali unito al fatto che i dati dello spacecraft N. 4 non sono presenti per la fase di addestramento potrebbe comportare una riduzione delle prestazioni, specialmente nella rilevazione per lo spacecraft N. 4.

2.2 Metodologia adottata

Di seguito si fornisce una breve descrizione degli strumenti impiegati per affrontare la challenge.

2.2.1 Matlab

MATLAB è un ambiente di calcolo numerico e per la programmazione e lo sviluppo di algoritmi, sviluppato da MathWorks. È ampiamente utilizzato in contesti ingegneristici e scientifici e viene impiegato per un'ampia gamma di applicazioni dall'analisi dei dati allo sviluppo di algoritmi. Può essere utilizzato anche per la modellazione e la simulazione di sistemi dinamici ma anche per task di machine learning. Il software inoltre mette a disposizione per gli utenti un'ampia gamma di tool applicativi. Per il lavoro da noi svolto gli applicativi messi a disposizione dalla piattaforma che abbiamo utilizzato sono i seguenti:

Diagnostic Feature Designer

Strumento che consente di estrarre caratteristiche diagnostiche dai dati di sensori. Le caratteristiche diagnostiche sono caratteristiche dei dati che possono essere utilizzate per identificare anomalie o malfunzionamenti nel sistema. Lo strumento ricevendo in input un dataset consente di svolgere diverse funzioni tra cui la generazione dello spettro mediante autoregressive model. Inoltre grazie a questo strumento è possibile estrarre le feature dai segnali applicando dei parametri come ad esempio delle framepolicy per la suddivisione del segnale. Il tool, oltre a consentire la generazione delle feature, sia nel dominio del tempo che dello spettro, consente anche la loro classificazione mediante diverse modalità di ranking anche in funzione del tipo di task di classificazione per cui i dati verranno elaborati. Inoltre il programma consente di estrapolare una funzione sotto forma di codice che può essere impiegata direttamente dall'ambiente di lavoro matlab per applicare tutte le operazioni di elaborazione, estrazione delle feature e classificazione di queste ultime. Questo consente di estrarre feature da dataset diversi seguendo sempre le stesse configurazioni senza necessariamente dover interagire tutte le volte con il tool ma eseguendo semplicemente la funzione su un set di dati.

Classification Learner

Strumento che, a partire dalle feature estratte da un dataset, consente di addestrare e validare diversi modelli di classificazione. L'addestramento può essere personalizzato inoltre andando a variare, tramite GUI, i parametri di configurazione. Inoltre lo strumento mette a disposizione diverse funzionalità come ad esempio la possibilità di confrontare quale modello ha performance migliori per

uno specifico task in funzione delle metriche di valutazione. È inoltre possibile riservare, in fase di caricamento delle feature, le modalità di gestione del dataset e se conservare una parte di quest'ultimo per il testing. Inoltre è possibile visualizzare confusion matrix e altri grafici utili allo scopo di valutare i modelli generati. Lo strumento, oltre a consentire l'esportazione del modello, consente anche di esportare la funzione matlab per la generazione di quest'ultimo che potrà quindi essere utilizzata direttamente all'interno dell'ambiente di lavoro per addestrare lo stesso modello con gli stessi parametri ma su un feature diverse senza dover necessariamente interagire con il tool apposito.

Regression Learner

Strumento analogo al Classification Learner ma che consente di addestrare modelli per task di regressione e non di classificazione. In questo caso le differenze principali con il precedente sono il tipo di metriche utilizzate per valutare i modelli e la tipologia di grafici generati dallo strumento.

2.2.2 Metodologia di svolgimento della challenge

Seguendo le linee guida della challenge e identificati gli strumenti necessari, la realizzazione del lavoro ha seguito una ben definita successione di step.

Innanzitutto è stato caricato e elaborato il dataset per renderlo utilizzabile con il Diagnostic Feature Designer.

Tale operazione è stata effettuata attraverso un'apposito script, DatasetStrutturato.m, in grado di ristrutturare il dataset messo a disposizione andando a creare all'interno del workspace di matlab una struttura apposita. Tale struttura viene creata unendo i dati provenienti dal file delle label con i dati dei rispettivi Case. Pertanto, per ogni case, lo script crea una riga di una tabella contenente nella prima colonna la timetable con le rilevazioni per quello specifico case, e sulle colonne successive le etichette per i corrispondenti task che fanno riferimento a quel case così come mostrato in Fig.2.2.

Capitolo 3

Task 1

Il Task 1 mira a distinguere la natura dei dati, categorizzandoli come comportamento *normale* o *anormale*.

All'interno del dataset i dati sono inizialmente classificati in tre categorie: *normale*, *anomalia* e *guasto*. Tuttavia, per semplificare il processo di classificazione in un contesto binario, sono state manipolate le etichette dei dati come:

- etichetta 0: assegnata ai dati classificati come *normali*;
- etichetta 1: assegnata ai dati classificati come *anomalia* o *guasto*.

3.1 Generazione delle feature

Dopo aver preparato i dati per l'analisi, sono stati utilizzati gli strumenti forniti dall'app Diagnostic Feature Designer per estrarre informazioni sia dall'andamento nel tempo (dominio temporale) che dalla distribuzione delle frequenze (dominio delle frequenze) dei segnali. Inizialmente è stata valutata l'opzione di analizzare l'intero segnale, ma successivamente è stata adottata una strategia di suddivisione del segnale in frame. Questa decisione è stata presa in seguito a valutazioni che hanno evidenziato risultati migliori con l'approccio basato su frame. Di conseguenza, nell'elaborato si discuterà esclusivamente dell'analisi condotta sui segnali suddivisi in frame.

3.1.1 Selezione della Frame Policy

La frame policy è un elemento fondamentale nel processo di analisi del segnale in MATLAB, specialmente quando si utilizzano tecniche basate su frame, come l'estrazione di caratteristiche e la successiva classificazione. In dettaglio, la frame policy definisce come il segnale viene segmentato in frame, ovvero in porzioni di dati più piccole che vengono analizzate individualmente. Pertanto, la scelta di una frame policy appropriata è cruciale per ottenere risultati accurati e significativi.

I parametri chiave della Frame Policy sono:

- *Frame Size (FS)*, il numero di campioni che compongono ciascun frame e determina la risoluzione temporale dell'analisi. Frame più grandi catturano più informazioni temporali, ma possono essere meno sensibili a brevi eventi o cambiamenti rapidi nel segnale.
- *Frame Rate (FR)* indica la frequenza con cui i frame vengono estratti dal segnale. Un Frame Rate elevato permette di analizzare il segnale con maggiore dettaglio temporale, ma può aumentare la complessità computazionale.

Uno dei principali fattori considerati nella selezione della frame policy è la natura intrinseca del segnale, inclusi fattori come la sua frequenza dominante e la presenza di eventi transitori. Questi aspetti, infatti, influenzano direttamente le scelte relative alla dimensione del frame (FS) e alla frequenza di campionamento dei frame (FR).

Nel contesto della challenge, si è optato per un $FS=FR=0.4s$, suddividendo così il segnale in 3 finestre. Questa scelta è stata motivata dal fatto che la misurazione totale del segnale ha una durata di 1200 millisecondi e comprende tre cicli completi di apertura e chiusura delle valvole. Di conseguenza, la suddivisione in tre finestre è sembrata la soluzione più appropriata e coerente con le caratteristiche del segnale a disposizione.

3.1.2 Calcolo dello Spettro di Potenza

Nell'ambito dell'analisi dei segnali per la valutazione delle prestazioni di un sistema, uno strumento prezioso è la generazione dello spettro di potenza. Lo spettro di potenza, infatti, fornisce informazioni essenziali sulla distribuzione dell'energia in frequenza e sulle risonanze presenti all'interno del sistema. Poiché le variazioni nelle prestazioni si riflettono spesso nella forma spettrale, l'esame dello spettro offre una ricca fonte di dati per l'analisi.

Per ottenere lo spettro di potenza, è stato impiegato un modello parametrico noto come modello AutoRegressive (AR) di ordine 4, questo significa che la stima si basa su 4 campioni del segnale. In generale, il modello AR si basa sull'ipotesi che il valore attuale di un segnale possa essere espresso come una combinazione lineare dei suoi valori passati, con l'aggiunta di un termine di errore casuale.

Una volta stimato il modello AR dai dati del segnale, è possibile calcolare la funzione di trasferimento del modello. Da questa funzione di trasferimenti, su può quindi derivare lo spettro di potenza.

Nel caso in esame, sono stati utilizzati 7 segnali di pressione provenienti da sensori posizionati in punti di misurazione specifici, P1-P7. L'analisi degli spettri di potenza ottenuti per ciascun segnale ha permesso di comprendere meglio il comportamento dei sistemi in esame. Di seguito sono mostrati i 7 spettri di potenza relativi al Task 1.

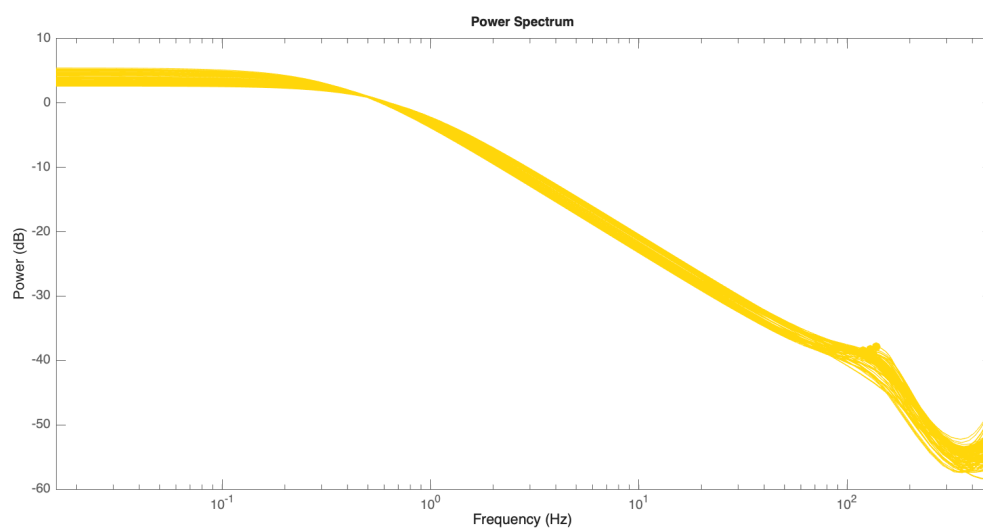


Figura 3.1: Spettro di potenza per il segnale P1

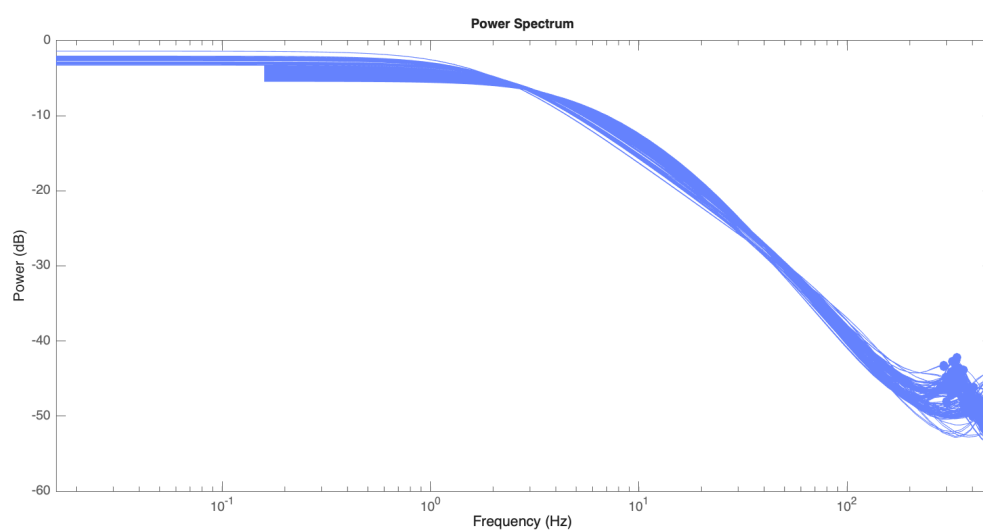


Figura 3.2: Spettro di potenza per il segnale P2

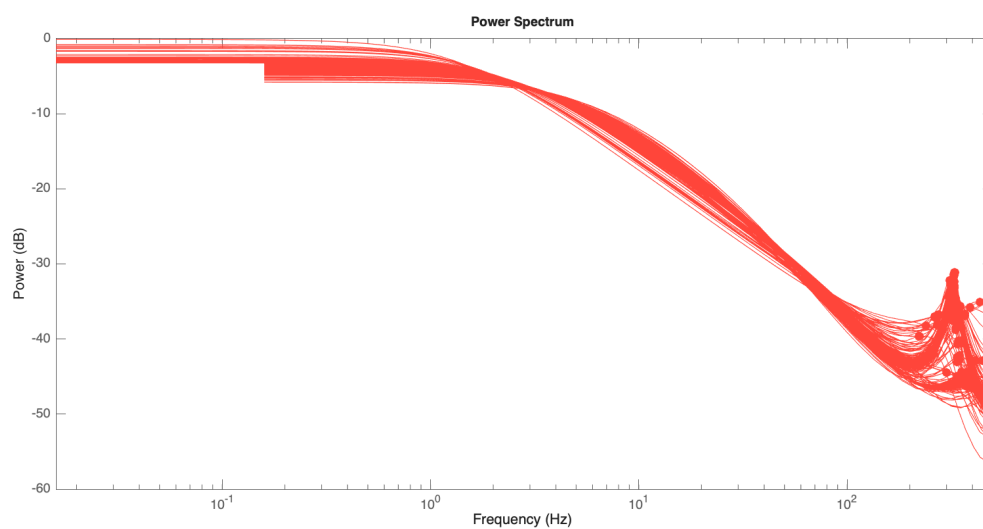


Figura 3.3: Spettro di potenza per il segnale P3

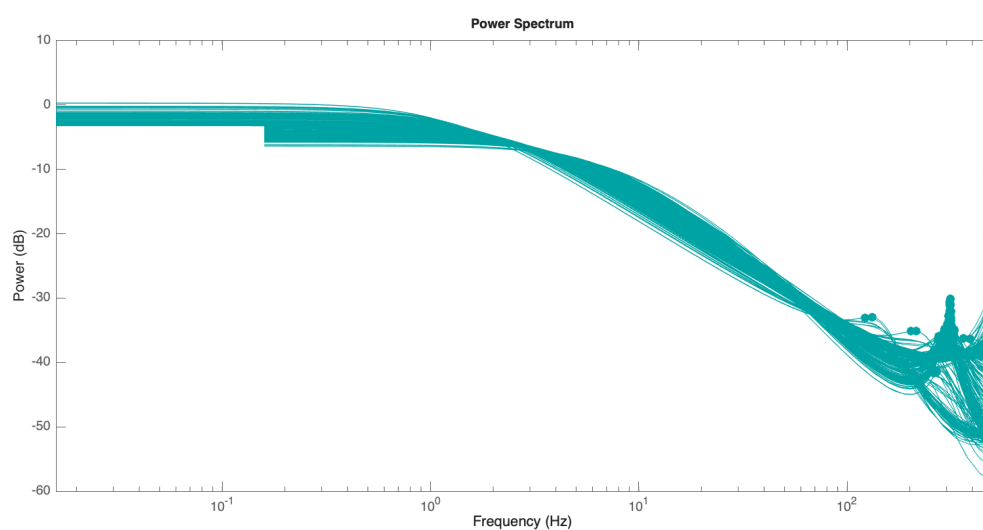


Figura 3.4: Spettro di potenza per il segnale P4

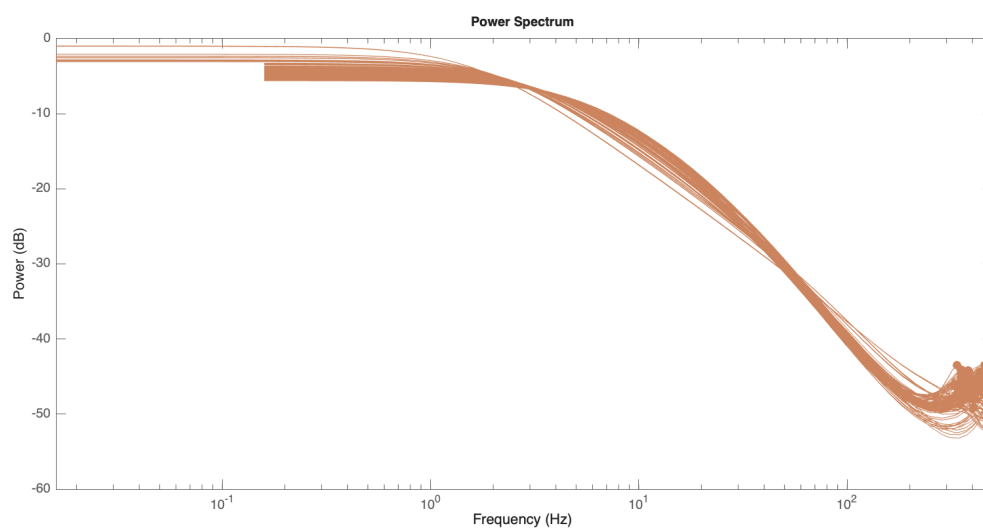


Figura 3.5: Spettro di potenza per il segnale P5

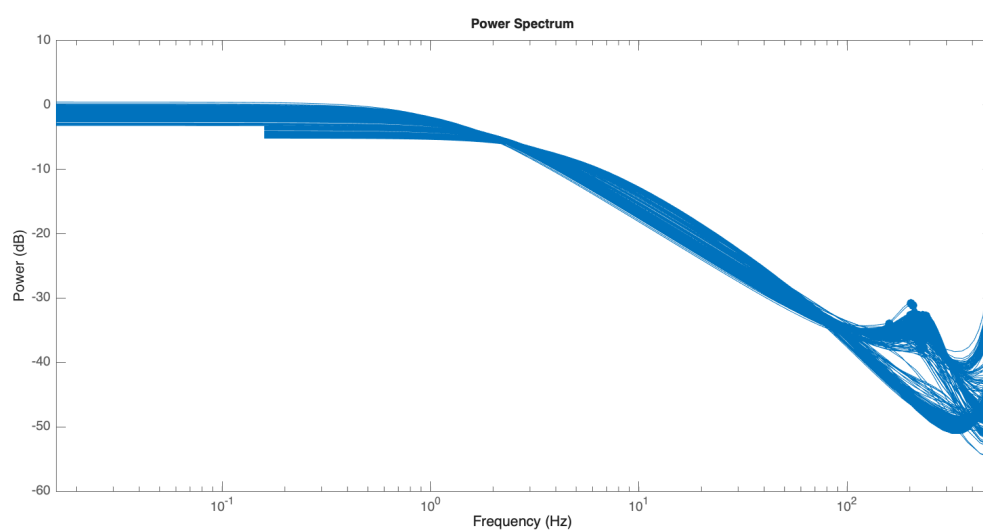


Figura 3.6: Spettro di potenza per il segnale P6

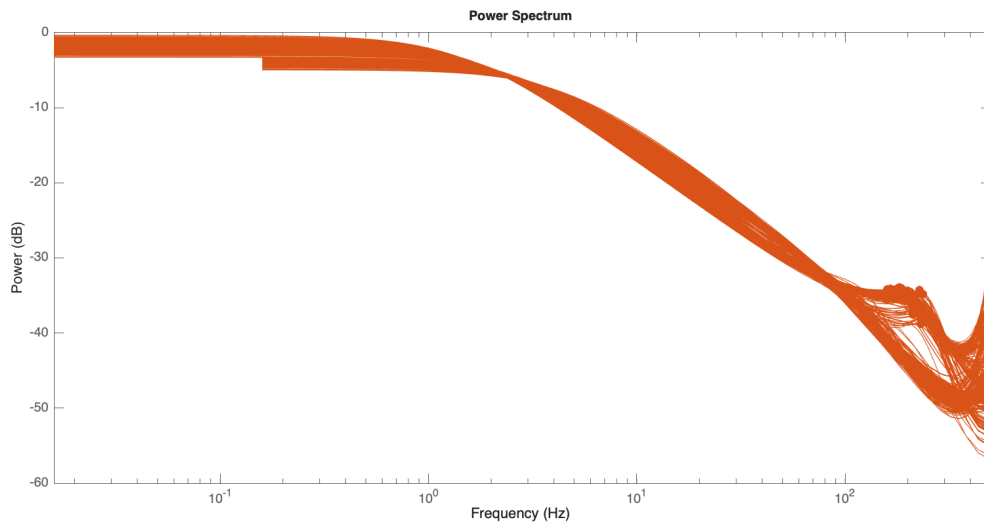


Figura 3.7: Spettro di potenza per il segnale P7

L'analisi di questi spettri di potenza ha mostrato che la maggior parte dell'informazione interessante per lo studio è contenuta nel range di frequenze 0-10 Hz. In particolare, fino a circa 10 Hz, lo spettro risulta abbastanza stabile, mentre a frequenze più alte si osservano picchi significativi dovuti a rumore e disturbi, specialmente nelle vicinanze dei 100 Hz.

Per evitare che il rumore presente nella parte alta dello spettro possa influenzare negativamente l'interpretazione dei risultati, è stata selezionata una banda di frequenza più ristretta con una buona marginalità, escludendo le frequenze superiori a 10 Hz.

Tuttavia, nei task successivi, l'analisi sugli spettri di potenza verrà omessa poiché i segnali di pressione rilevanti per il task in questione sono già stati generati.

3.1.3 Feature diagnostiche nel dominio delle frequenze

Un metodo comune ed efficace che il Diagnostic Feature Designer adopera per generare le feature diagnostiche nel dominio delle frequenze è l'utilizzo dello spettro di potenza, illustrato precedentemente. Tale approccio rappresenta un potente strumento per l'analisi dei segnali provenienti da sistemi complessi al fine di identificare anomalie, guasti o altri comportamenti indesiderati.

In pratica, le feature estratte dal dominio delle frequenze forniscono informazioni preziose sulla distribuzione dell'energia in frequenza e sul comportamento dinamico del sistema oggetto di analisi.

3.1.4 Feature diagnostiche nel dominio del tempo

L'app Diagnostic Feature Designer mette a disposizione la funzionalità *Signal Features* per la generazione di feature diagnostiche nel dominio del tempo, particolarmente utili per l'analisi di segnali stocastici. Queste feature, che includono parametri statistici tipici come media, varianza, curtosi e skewness, forniscono una rappresentazione sintetica del comportamento medio del segnale nel tempo.

In sintesi, l'analisi di tali feature permette di identificare eventuali cambiamenti o anomalie nel segnale, che possono essere indicativi di variazioni nello stato di salute del sistema sotto esame.

3.1.5 Selezione delle feature

Data la grande quantità di feature generate sia nel dominio della frequenza che nel dominio del tempo, è stato essenziale eseguire una selezione delle caratteristiche più rilevanti prima di procedere con la classificazione. Tale processo di selezione delle feature è stato realizzato utilizzando metriche di ranking che valutano l'importanza di ciascuna feature nel contesto della classificazione.

Per il caso in esame, sono state adottate metriche di ranking supervisionato, tra cui il T-test e la ROC. Il T-test determina se le differenze tra le medie di due gruppi etichettati sono statisticamente significative, mentre la curva ROC valuta le prestazioni dei modelli di classificazione binaria.

Dopo aver valutato i risultati ottenuti con entrambe le metriche, si è deciso di utilizzare la curva ROC per la selezione delle feature. In base a questa metrica, sono state selezionate le prime 30 feature più significative per la classificazione, evidenziate in un riquadro rosso nella Figura 3.8, rappresentando un sottoinsieme ottimale per il successivo addestramento del modello di classificazione.

Una volta aver selezionato le feature, queste possono essere esportate sia nel Classification Learner che nel workspace per analisi ulteriori.

3.2 Addestramento modello di classificazione binario

Una volta selezionate le feature più rilevanti, esse sono state impiegate per addestrare diversi modelli di classificazione disponibili nell'app Classification Learner e valutate le prestazioni migliori ottenute utilizzando la metrica di validazione adottata.

Inizialmente, sono state utilizzate due tecniche di validazione: l'holdout a 20% e la k-fold cross-validation a 5 fold. Tuttavia, a seguito dell'analisi dei risultati, si è scelto di scartare l'holdout e utilizzare esclusivamente la k-fold cross-validation per i migliori risultati ottenuti. Questa tecnica di validazione consente di effettuare più valutazioni del classificatore attraverso k cicli, permettendo così di

Feature	T-Test	ROC
Case_sigstats_5/Std	9.6306	0.2342
Case_sigstats_5/ShapeFactor	9.7424	0.2333
Case_sigstats/Std	9.1342	0.2239
Case_sigstats/ShapeFactor	9.2167	0.2230
Case_sigstats/RMS	9.4738	0.2215
Case_sigstats_3/SNR	8.0942	0.2061
Case_sigstats_5/RMS	10.0544	0.1978
Case_sigstats_1/Std	8.4698	0.1959
Case_sigstats_7/Kurtosis	8.1182	0.1947
Case_sigstats_1/ShapeFactor	8.4815	0.1916
Case_sigstats_3/Mean	7.5338	0.1909
Case_sigstats_1/RMS	8.7284	0.1900
Case_sigstats_6/Kurtosis	6.8973	0.1838
Case_ps_4_spec/BandPower	8.5314	0.1804
Case_sigstats_7/THD	7.3930	0.1774
Case_sigstats_1/SINAD	6.4689	0.1737
Case_sigstats_6/Mean	5.8992	0.1703
Case_ps_1_spec/BandPower	9.1967	0.1701

Feature	T-Test	ROC
Case_sigstats_6/ShapeFactor	6.6241	0.1670
Case_sigstats_5/SINAD	6.5329	0.1639
Case_sigstats_6/Std	6.6370	0.1634
Case_sigstats_7/Std	7.5160	0.1627
Case_sigstats_5/THD	7.4971	0.1614
Case_sigstats_7/ShapeFactor	7.4462	0.1612
Case_sigstats_5/SNR	6.1884	0.1580
Case_sigstats_1/SNR	7.1665	0.1544
Case_sigstats/PeakValue	6.7535	0.1542
Case_sigstats/ImpulseFactor	6.7387	0.1534
Case_sigstats/ClearanceFa...	6.8028	0.1534
Case_sigstats/CrestFactor	6.6466	0.1531
Case_sigstats_7/RMS	6.4029	0.1507
Case_ps_2_spec/BandPower	7.4037	0.1297
Case_sigstats_7/CrestFactor	4.2424	0.1233
Case_sigstats_3/SINAD	5.2879	0.1154
Case_sigstats_4/SNR	4.6623	0.1128
Case_sigstats_3/ShapeFactor	3.8350	0.1101

Figura 3.8: Selezione feature task 1

mediare e bilanciare l'influenza di eventuali bias o casi fortunati nella suddivisione del dataset.

In particolare, la k-fold cross-validation opera nel seguente modo:

1. Il dataset di partenza viene suddiviso in k insiemi.
2. Si eseguono k iterazioni con le seguenti operazioni:
 - Si seleziona uno dei k insiemi per la validazione, assicurandosi che non sia stato già utilizzato nelle iterazioni precedenti.
 - I restanti k-1 insiemi vengono utilizzati per addestrare il classificatore.
 - Il classificatore addestrato viene utilizzato per fare previsioni sull'insieme selezionato.
 - I risultati delle previsioni vengono salvati.
3. I risultati delle previsioni ottenute durante le k iterazioni vengono confrontati con le etichette reali dei dati per valutare le prestazioni del classificatore.

Per il task 1, tra le varie opzioni disponibili, il modello che ha mostrato le migliori performance si basa sull'algoritmo *Ensemble Bagged Trees*. Questa tecnica di apprendimento ensemble è utilizzata per la classificazione supervisionata e combina diverse istanze di alberi decisionali, noti come "bagged trees", per migliorare le prestazioni complessive del modello.

La matrice di confusione, presentata nella Fig.3.9, mostra un'accuratezza complessiva del 94.2% sui dati di validazione. Tale risultato evidenzia che il classificatore è in grado di identificare quasi tutti i casi di normale funzionamento, ma potrebbe non riconoscere correttamente tutti i casi di guasto.

Sebbene l'obiettivo ideale sia evitare qualsiasi errore di classificazione, è preferibile che il modello classifichi erroneamente un caso di guasto come normale piuttosto che il contrario. Questo perché l'identificazione di un dato come anomalo può innescare una serie di costose procedure di manutenzione.

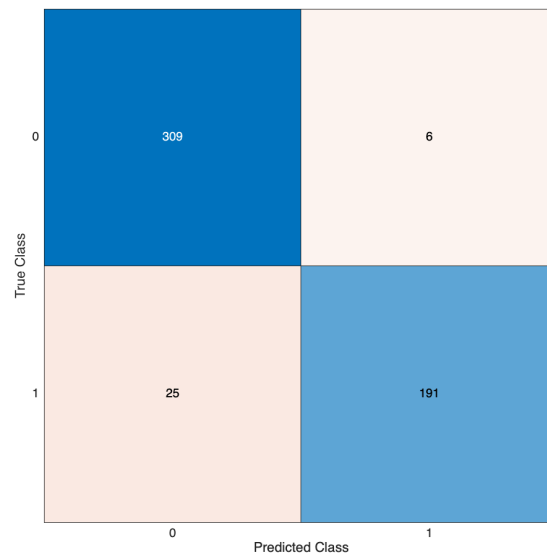


Figura 3.9: Matrice di confusione per il task 1 sui dati di validazione

Capitolo 4

Task 2

Il task 2 ha l'obiettivo di identificare la causa del guasto per i casi anormali individuati nel task 1. Le classi di guasto da distinguere sono le seguenti:

- unknown fault, anomalie o guasti imprevisi e non identificati in precedenza; a tale classe è associata l'etichetta '1'.
- bubble contamination, ovvero la presenza di bolle d'aria nei tubi del sistema di propulsione; a tale classe è associata l'etichetta '2'.
- solenoid valve fault, cioè il malfunzionamento di valvole; a tale classe è associata l'etichetta '3'.

4.1 Soluzioni individuate

La mancanza di dati di addestramento per la classe "unknown fault" rappresenta un ostacolo significativo per il task 2. I metodi di machine learning standard, che si affidano all'apprendimento da dati pre-etichettati, non sono adatti a questo scenario. Sono necessarie soluzioni alternative per classificare correttamente i casi di guasto sconosciuto. Per tali motivi sono state pensate diverse soluzioni; in particolare sono stati analizzati vantaggi e svantaggi di 3 delle tecniche individuate:

1. **Inserimento di dati sporchi nel dataset:** Aggiungere artificialmente dati "sporchi" al dataset di training che simulano la presenza di anomalie sconosciute. Questi dati sporchi possono essere generati in vari modi, come ad esempio aggiungendo rumore casuale ai segnali o alterando le caratteristiche in base a regole predefinite. Tale soluzione può migliorare la capacità del modello di generalizzare a casi sconosciuti. Lo svantaggio è che richiede la creazione manuale di dati sporchi, che può essere un processo dispendioso in termini di tempo e risorse; inoltre, la qualità dei dati sporchi generati influenza pesantemente le prestazioni del modello.

2. **Classificatori one-class in cascata:** Utilizzare una serie di classificatori one-class in cascata per identificare i diversi tipi di guasto. Il primo classificatore distingue tra "guasto 2" e il resto dei guasti. Il secondo classificatore distingue tra "guasto 3" e il resto dei guasti (guasti sconosciuti). Tale soluzione presenta una struttura semplice e intuitiva e può essere efficiente dal punto di vista computazionale. Lo svantaggio principale è che l'errore di un classificatore può propagarsi ai successivi, causando una riduzione complessiva dell'accuratezza.
3. **Classificatore one-class e binario:** Utilizzare un classificatore "one-class" per identificare le anomalie. Un classificatore binario viene poi utilizzato per distinguere tra guasto 2 e guasto 3 per i dati identificati come anomali. Tale soluzione può essere efficace per identificare le anomalie sia conosciute che sconosciute. Inoltre, la struttura è relativamente semplice con due modelli. La criticità sta nella scelta dell'algoritmo "one-class" che risulta fondamentale per il rilevamento accurato delle anomalie.

Sebbene la prima tecnica possa migliorare la capacità del modello di generalizzare a casi sconosciuti, è stata scartata a causa della significativa richiesta di risorse per la creazione manuale di dati sporchi di alta qualità. Inoltre, la qualità dei dati sporchi generati influenza pesantemente le prestazioni del modello. Alla luce di tale analisi, è stato deciso di implementare e confrontare i risultati delle ultime due soluzioni.

4.2 Classificatori one-class in cascata

L'approccio si basa sull'utilizzo di una serie di classificatori one-class disposti in cascata, ciascuno addestrato per distinguere una specifica classe di guasto dalle restanti. Questo schema a due livelli permette di sfruttare le caratteristiche distintive di ogni classe di guasto in modo efficiente. Gli step eseguiti per il training di tali classificatori sono i seguenti:

1. *Ristrutturazione del dataset di training:* Sono state rimosse le condizioni normali dal dataset di training.
2. *Generazione delle feature:* Sono state estratte le feature tramite l'app Diagnostic Feature Design.
3. *Addestramento dei modelli:* Per ogni classe di guasto viene addestrato un modello Linear Discriminant Analysis (LDA). Tale modello permette di distinguere la classe di guasto in questione da tutte le altre classi di guasto presenti nel dataset di training attraverso la separazione dello spazio delle feature con un iperpiano. Per valutare i modelli addestrati, è stata utilizzata la cross-validation a 5 fold. Quindi, per ogni modello sono stati memorizzati i risultati e calcolata l'accuracy mostrata nella Fig.4.1.

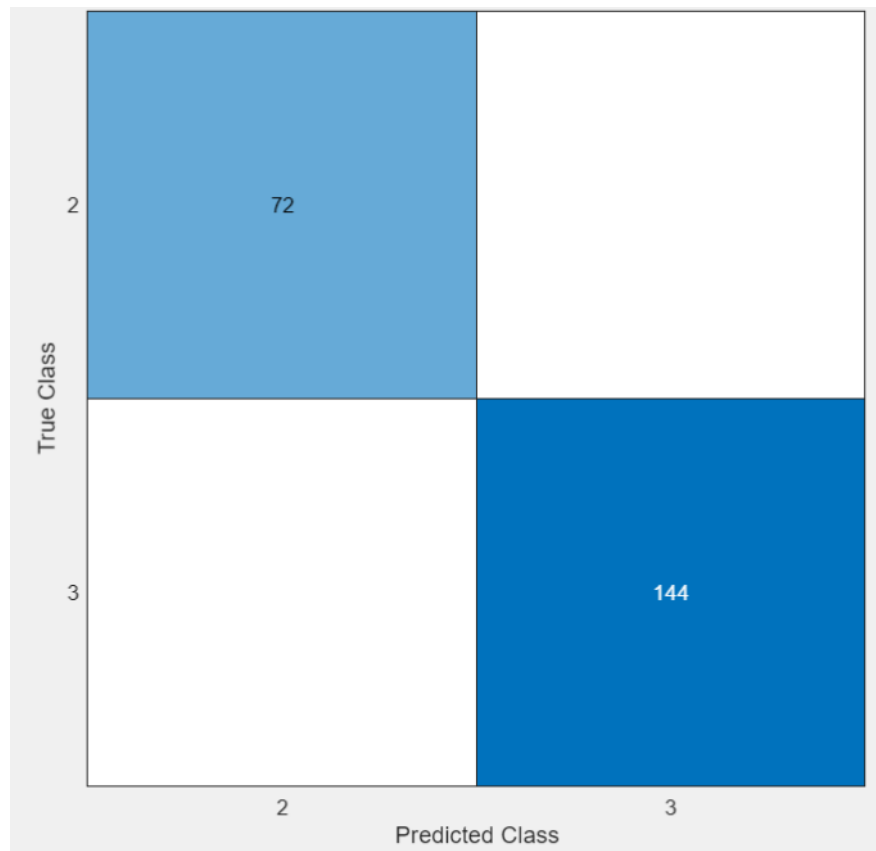


Figura 4.1: Matrice di confusione del modello one-class in cascata

4.3 Classificatore one-class e binario

Il classificatore one-class seguito da quello binario rappresenta un approccio efficace per la classificazione dei guasti in presenza di dati di training limitati per le anomalie sconosciute. Questa strategia combina due modelli distinti: un classificatore one-class per identificare le anomalie generiche e un classificatore binario per distinguere tra i tipi di guasto noti (guasto 2 e guasto 3).

4.3.1 Addestramento del modello One-Class SVM

Gli step eseguiti per l'addestramento del primo modello sono i seguenti:

1. *Ristrutturazione del dataset di training*: Sono state rimosse le condizioni normali dal dataset di training, poiché il classificatore one-class si concentra sull'identificazione delle anomalie. In seguito è stata assegnata un'etichetta "0" a tutte le istanze restanti, indipendentemente dal tipo di guasto.

2. *Generazione delle feature*: Sono state estratte le feature tramite l'app Diagnostic Feature Design. La scelta delle feature appropriate è cruciale per l'accuratezza del modello. In particolare, è stato eseguito un ranking delle features secondo il criterio monotonicity. La monotonicità descrive la relazione tra due variabili, indicando se il loro andamento è crescente, decrescente o costante. Nel ranking delle feature, la monotonicità è utile per identificare le feature più informative per la discriminazione tra dati normali e anomali.
3. *Stratificazione del dataset*: Il dataset di training ristrutturato è stato diviso in un set di training (80%) e un set di test (20%). Tale suddivisione è stata eseguita mantenendo la proporzione tra guasti noti e anomalie nel dataset originale.
4. *Addestramento del modello*: È stato utilizzato un algoritmo di Support Vector Machine (SVM) one-class per addestrare il modello in grado di identificare le anomalie generiche. L'algoritmo SVM one-class impara a definire un confine nello spazio delle feature che racchiude i dati normali (in questo caso i guasti conosciuti) e separa le anomalie (ovvero i guasti sconosciuti).

4.3.2 Addestramento del modello binario

Gli step eseguiti per l'addestramento del secondo modello sono i seguenti:

1. *Ristrutturazione del dataset di training*: Sono state rimosse le condizioni normali dal dataset di training, quindi è stato creato un nuovo training set contenente solo le istanze etichettate come "guasto 2" e "guasto 3".
2. *Generazione delle feature*: Sono state estratte le feature tramite l'app Diagnostic Feature Design.
3. *Addestramento del modello*: È stata utilizzata l'app Classification Learner di MATLAB per addestrare un modello binario in grado di distinguere tra "guasto 2" e "guasto 3". L'accuratezza ottenuta per il validation set utilizzando il modello Quadratic Discriminant, che è risultato il migliore, è riportata in Fig.4.2.

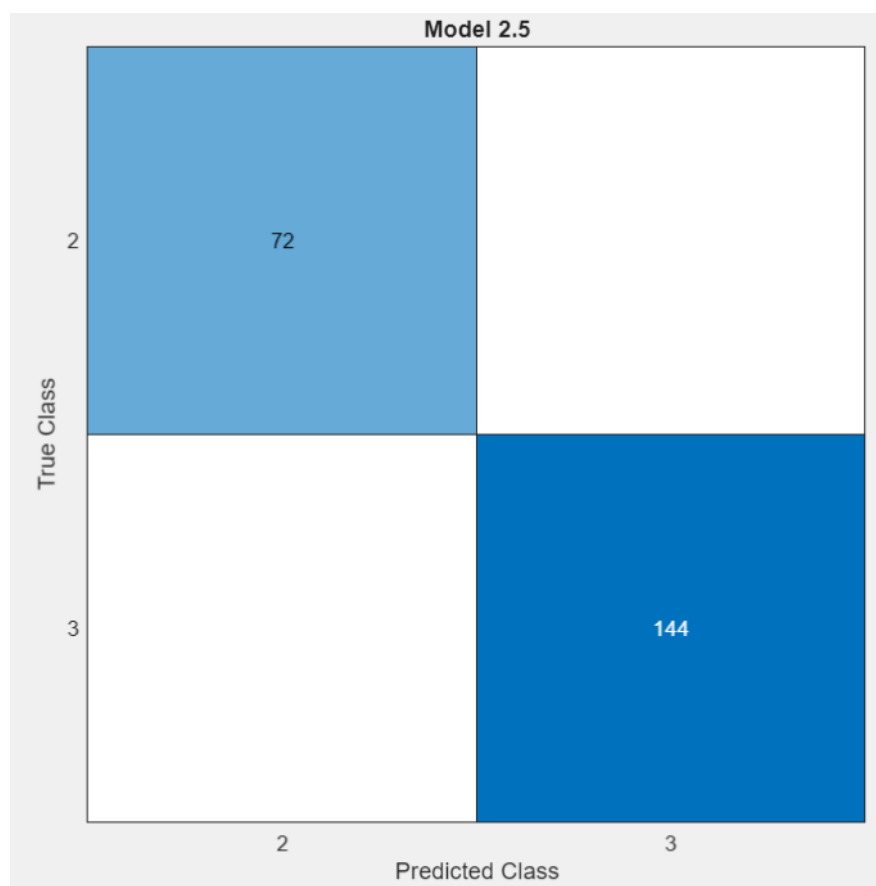


Figura 4.2: Matrice di confusione del modello binario

Capitolo 5

Task 3

Il task 3 mira a determinare la posizione specifica della bolla per tutti i dati identificati nel task 2 come contaminati dalla presenza di bolle. Ogni posizione della bolla è associata a un'etichetta numerica specifica:

- *BP1* (etichetta numerica 1)
- *BP2* (etichetta numerica 2)
- *BP3* (etichetta numerica 3)
- *BP4* (etichetta numerica 4)
- *BP5* (etichetta numerica 5)
- *BP6* (etichetta numerica 6)
- *BP7* (etichetta numerica 7)
- *BV1* (etichetta numerica 8)

5.1 Generazione delle feature

Una volta che i dati sono stati preparati attraverso il pre-processing per il task 2, si è proceduto con la generazione delle feature sia nel dominio del tempo che nel dominio delle frequenze utilizzando l'app Diagnostic Feature Designer.

Anche per questo task sono stati generati gli spettri dei segnali utilizzando il modello parametrico Autoregressive (AR) con l'ordine del modello pari a 4.

Seguendo la metodologia simile a quella utilizzata per i task precedenti, le feature nel dominio del tempo sono state calcolate tramite la funzionalità Signal Features messa a disposizione dall'app che estrae le caratteristiche statistiche di base dai segnali. Le feature nel dominio delle frequenze, invece, sono state

ottenute attraverso la funzionalità Spectral Features che analizza le proprietà spettrali dei segnali.

Poiché il task coinvolge una classificazione multiclasse, il metodo di ranking delle feature utilizzato è stato l'ANOVA mostrato nella Fig.5.1 selezionando le prime 30 evidenziate nel riquadro rosso.

Feature	One-way ANOVA
Case_sigstats/THD	203.9829
Case_sigstats/SINAD	150.2300
Case_sigstats/SNR	107.6598
Case_ps_11_spec/BandPower	68.9455
Case_sigstats_5/Skewness	59.7321
Case_ps_11_spec/PeakFreq1	47.9399
Case_ps_7_spec/BandPower	46.5030
Case_sigstats_6/CrestFactor	43.5807
Case_sigstats_6/ImpulseFa...	41.7569
Case_sigstats_6/PeakValue	41.3125
Case_sigstats_6/Kurtosis	40.2830
Case_sigstats_6/Clearance...	38.4172
Case_sigstats/RMS	35.0778
Case_sigstats_5/Kurtosis	31.8856
Case_sigstats/Skewness	29.4589
Case_sigstats_6/Skewness	26.5178
Case_sigstats_6/THD	22.2169
Case_sigstats_4/RMS	21.5510

Feature	One-way ANOVA
Case_sigstats_1/RMS	21.3604
Case_sigstats_4/Skewness	19.9996
Case_sigstats_2/SNR	19.0929
Case_sigstats_1/SNR	18.9114
Case_sigstats_4/THD	17.2137
Case_ps_9_spec/BandPower	16.2794
Case_sigstats/Kurtosis	15.8905
Case_ps_8_spec/BandPower	15.2198
Case_ps_11_spec/PeakAmp1	14.9597
Case_sigstats_1/SINAD	13.9764
Case_sigstats_3/Mean	12.2880
Case_sigstats_6/RMS	11.0266
Case_sigstats_4/SINAD	10.3738
Case_sigstats_4/SNR	8.6871
Case_sigstats_1/Skewness	8.4275
Case_sigstats_5/Mean	8.0484
Case_sigstats_3/SNR	7.7764
Case_sigstats_2/Skewness	6.7522

Figura 5.1: Selezione feature task 3

5.2 Addestramento modello di classificazione multiclasse

In primo luogo, si è proceduto alla creazione di un "nuovo" dataset di training escludendo le istanze del comportamento normale e includendo solo i dati etichettati come contaminati dalla presenza di bolle. Tale decisione è stata motivata dalla natura a cascata del problema di classificazione, come specificato dalle direttive della challenge. In questo task, infatti, l'obiettivo del classificatore è identificare la corretta posizione della contaminazione da bolle.

Successivamente, le feature sono state importate nell'app Classification Learner e, seguendo l'approccio adottato per gli altri task, è stata utilizzata la metrica di validazione basata sulla K-Cross Fold Validation a 5 fold. Sono stati addestrati diversi modelli disponibili nel toolbox dell'app.

Per il task 3, il modello che ha dimostrato le migliori performance si basa ancora una volta sull'algoritmo *Ensemble Bagged Trees*. Questo modello ha ottenuto

un'accuratezza complessiva del 95.8% sui dati di validazione, come evidenziato nella matrice di confusione presentata nella Fig.5.2. Sebbene il modello sia in grado di classificare correttamente la maggior parte delle classi, sembra presentare alcune difficoltà con alcune di esse.

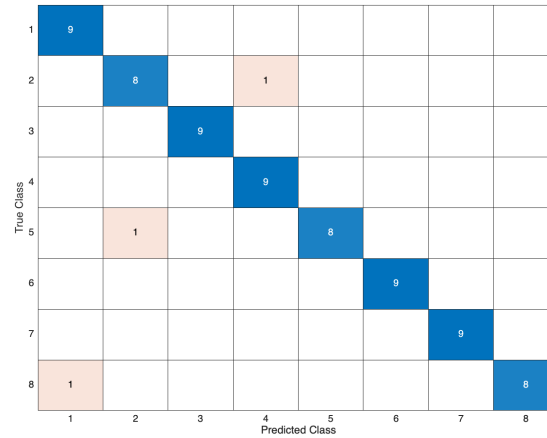


Figura 5.2: Matrice di confusione per il task 3 sui dati di validazione

Capitolo 6

Task 4

Sulla base delle informazioni raccolte nel Task 2, che hanno evidenziato la presenza di un guasto su una valvola, il Task 4 ha l'obiettivo di identificare con precisione la valvola specifica responsabile del malfunzionamento.

All'interno del sistema operano quattro valvole, ognuna con una propria etichetta numerica per un'identificazione precisa:

- SV1, etichettata con il numero 1.
- SV2, etichettata con il numero 2.
- SV3, etichettata con il numero 3.
- SV4, etichettata con il numero 4.

6.1 Generazione delle feature

Attraverso l'app messa a disposizione da MATLAB, Diagnostic Feature Designer, sono state generate le feature a partire dal dataset strutturato.

In particolare, come per gli altri task, a seguito di diverse prove eseguite per garantire il miglior risultato, è stato scelto di applicare una frame policy con un frame rate e frame size a 0.4 secondi.

Quindi, dopo aver generato lo spettro del segnale utilizzando il modello AutoRegressive, sono state generate le feature sia nel dominio del tempo che nel dominio delle frequenze.

Infine, è stato eseguito il ranking delle feature con il metodo One-Way ANOVA in quanto il task riguarda un problema di classificazione multiclasse.

6.2 Addestramento modelli

Le feature generate sono state esportate nel Classification Learner al fine di addestrare e confrontare i diversi modelli di classificazione.

Sono state eseguite molteplici prove selezionando inizialmente 20 feature poi 30 feature; inoltre, sono stati addestrati i diversi modelli variando il metodo di suddivisione del dataset in training set e validation set; in particolare, sono state provate l'Holdout e la Cross-Validation.

La configurazione migliore è stata trovata selezionando 30 feature ed utilizzando la Cross-Validation a 5 fold.

L'accuratezza ottenuta per il validation set utilizzando il modello Ensemble Subspace KNN, che è risultato il migliore, è riportata in Fig.6.1.

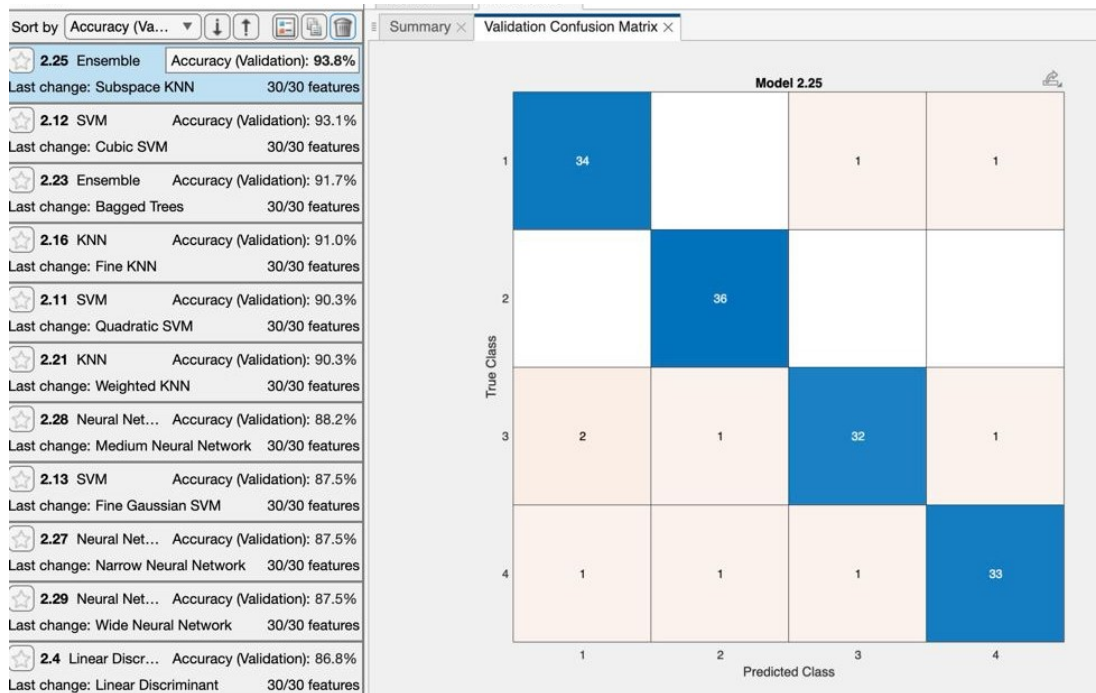


Figura 6.1: Matrice di confusione del modello per Task 4

Capitolo 7

Task 5

Il task 5 si occupa di determinare, per quei segnali individuati come anomalia di apertura della valvola, la percentuale di apertura di quest'ultima. Rispetto alle casistiche espresse in precedenza in questo caso siamo di fronte a un problema di regressione.

7.0.1 Generazione delle feature

Per quanto riguarda la parte di generazione delle feature sono state ripetute le procedure analoghe svolte nei casi precedenti. Pertanto, dopo aver preparato i dati tramite il pre-processing per il task in questione, si è proceduto con la generazione delle feature sia nel dominio del tempo che nel dominio delle frequenze utilizzando l'app Diagnostic Feature Designer. Anche per questo task la framepolicy è stata mantenuta a 0.4 e sono stati generati gli spettri dei segnali utilizzando il modello parametrico AutoRegressive (AR) con l'ordine del modello pari a 4. Le feature nel dominio del tempo sono state calcolate tramite la funzionalità Signal Features messa a disposizione dall'app che estrae le caratteristiche statistiche di base dai segnali. Mentre le feature nel dominio delle frequenze sono state ottenute attraverso la funzionalità Spectral Features.

Rispetto alle casistiche usate in precedenza, in questo caso sono il ranking delle feature è stato effettuato su più metodi:

- ANOVA
- MONOTONICITY
- TRENDABILITY
- PROGNOSABILITY

Per ognuna delle modalità è stata esportata sia la funzione di generazione delle feature, utile per le fasi successive di testing, che la tabella contenente le feature estratte dai dati di allenamento.

7.0.2 Addestramento dei modelli

Per ogni metodologia di ranking delle feature, le tabelle ottenute sono state caricate nel tool Regression Learner Messo a disposizione da Matlab. Successivamente sono stati generati i modelli e analizzate le prestazioni di questi ultimi. Al fine di migliorare la performance si è deciso di rimuovere dalle feature tutti i dati relativi alle valvole completamente aperte in quanto, per gli obiettivi del task 5, questa condizione, cioè un'apertura corretta al 100%, non viene mai analizzata dal classificatore che in questa fase si vuole realizzare. Questo perché il task 5 deve occuparsi di determinare la percentuale di apertura delle valvole che non si sono aperte correttamente e che quindi di certo non avranno un'apertura pari al 100%. Eliminare pertanto la condizione di apertura al 100% consente di migliorare la precisione di predizione in quanto il modello si concentra esclusivamente sulla tipologia di dati che riceverà. Sulla base di queste considerazioni sono stati generati i modelli per ogni metodo di ranking delle feature esclusivamente per i dati con un'apertura diversa dal 100% e si è riscontrato un notevole aumento delle performance. Sulla base dei risultati in fase di validation si è deciso di mantenere il modello ottenuto mediante le feature il cui ranking è stato effettuato con il metodo trendability con i dati ridotti. Sia nel caso con tutti i dati, sia nel caso con i dati ridotti infatti questo metodo ha prodotto il miglior punteggio in fase di validation. Nelle in Fig.7.1 e Fig.7.2 è possibile vedere i risultati ottenuti dai modelli generati usando le feature estratte con il ranking mediante il metodo Trendability con i dati integrali e con i dati ridotti.



Figura 7.1: Risultati validation ranking di tipo Trendability con feature integrali.

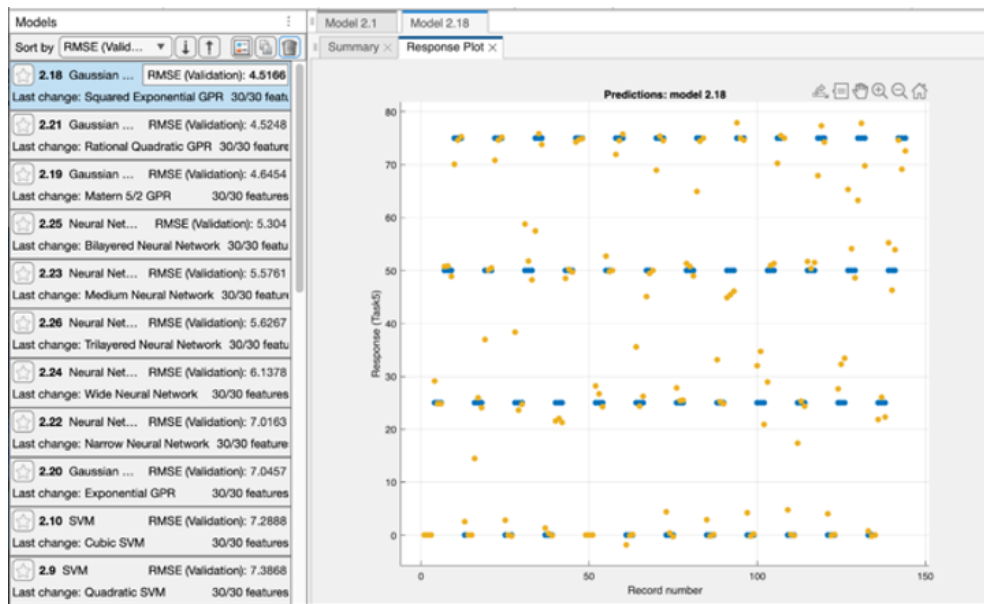


Figura 7.2: Risultati validation ranking di tipo Trendability con feature ridotte.

In entrambi i casi il modello è un Gaussian Regression Process. La metrica impiegata in fase di validation è la RMSE che, nel caso delle feature integrali ha un valore di 6,13, mentre nel caso delle feature ridotte ha un valore di 4,51. Da questi valori è evidente che la riduzione delle feature comporta un notevole miglioramento del modello.

Capitolo 8

Testing

In tale sezione, vengono esaminati i risultati ottenuti per ciascun task attraverso l'utilizzo di modelli addestrati sui dati di training e successivamente testati su dati di testing.

Il primo passo cruciale è stato organizzare il dataset di test in modo tale da garantire una valutazione accurata delle soluzioni proposte.

In generale, la metodologia di testing adottata per tutti i task segue i seguenti passaggi:

- *Generazione delle features*: è stata garantita l'identità delle features estratte durante la fase di validazione e utilizzate per addestrare i modelli di classificazione. Per mantenere uniformità tra le features utilizzate durante l'addestramento e quelle impiegate nel testing, sono state sfruttate le funzioni di generazione delle features esportate dal Diagnostic Feature Designer. Quest'approccio assicura coerenza nel processo di feature extraction.
- *Caricamento del modello addestrato*: il modello di classificazione precedentemente addestrato è stato caricato, pronto per essere utilizzato nel processo di testing.
- *Predizione delle classi*: le classi dei dati di test sono state predette applicando una frame policy, cioè suddividendo il segnale in frame più piccoli e classificando ciascun frame individualmente. Per determinare la classe del segnale nel suo insieme, sono state combinate le predizioni di tutti i frame utilizzando un algoritmo di maggioranza 2/3. In particolare, questo algoritmo considera una predizione come *anomala* solo se almeno due terzi dei frame temporali restituiscono un risultato *anomalo*, seguendo i seguenti passaggi:
 - Definizione del threshold di maggioranza: viene definito un valore di soglia pari a 2, il che significa che la previsione viene etichettata come *anomala* solo se almeno due terzi delle finestre temporali ritornano un risultato di *anomalo*.

- Calcolo del conteggio di maggioranza: viene calcolato il numero minimo di finestre temporali necessario per considerare la previsione come *anomala*. Questo valore è calcolato come un terzo del numero totale di finestre temporali moltiplicato per il valore di soglia di maggioranza. Questo assicura che almeno due terzi delle finestre temporali devono predire l'etichetta *anomala* per considerare l'intero esperimento come *anomalo*.
- *Calcolo dell'accuratezza e generazione della matrice di confusione*: i risultati sui dati di test, ad eccezione del task 1, vengono discussi riportando l'accuratezza complessiva e due matrici di confusione. La prima è calcolata sulla base del formato di consegna dei risultati richiesto dalla challenge, mentre la seconda prende in considerazione solo i dati effettivamente classificati dai modelli addestrati. Infatti, la suddivisione in task imposta dalla competizione è, di fatto, una classificazione in cascata, in cui i risultati dei task precedenti influenzano quelli dei task successivi.

8.1 Task 1

Per il testing del task 1, è stato impiegato il modello basato sull'algoritmo di classificazione *Ensemble Bagged Trees*, il quale è stato configurato in base ai risultati della fase di validazione.

La matrice di confusione, presentata nella Fig.8.1, evidenzia che il modello è in grado di correttamente identificare tutti i casi di normale funzionamento. Tuttavia, si registrano degli errori nella previsione della classe per alcuni casi di funzionamento anomalo.

L'accuratezza complessiva ottenuta, calcolata dalla matrice di confusione, risulta essere del 89.13%.

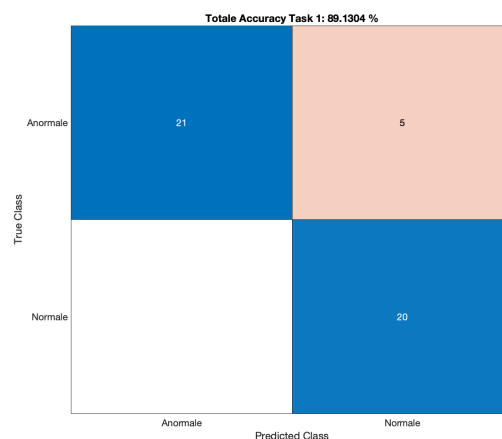


Figura 8.1: Matrice di confusione per il task 1 sui dati di test

8.2 Task 2

Il testing è stato eseguito per entrambi i metodi implementati al fine di identificare e selezionare la tecnica che garantisca risultati migliori.

8.2.1 Classificatori one-class in cascata

Il testing del metodo caratterizzato dai classificatori one-class in cascata è stato eseguito in diversi step:

1. Sono stati selezionati i dati di test che il task 1 ha classificato come potenziali guasti.
2. Sono state generate le feature del dataset senza casi di normale funzionamento con la funzione specifica del task estratta dal Diagnostic Feature Designer.
3. Si applica il modello LDA addestrato per la classe di guasto 2 per classificare i dati di test selezionati; quindi, ai dati classificati come "Guasto 2" viene assegnata l'etichetta 2 nella matrice delle previsioni finali.
4. I dati etichettati con il numero 2 (bubble contamination) vengono rimossi dal dataset di test in modo da non considerarli nelle successive fasi di classificazione.
5. Si applica il modello LDA addestrato per la classe di guasto 3 per classificare i dati di test rimanenti; quindi, ai dati classificati come "Guasto 3" viene assegnata l'etichetta 3 nella matrice delle previsioni finali.
6. I dati etichettati con il numero 3 (valve fault) vengono rimossi dal dataset di test.
7. I dati di test rimanenti vengono classificati come guasti sconosciuti, quindi etichettati con il numero 1.

L'accuratezza ottenuta con tale tecnica, considerando solo i dati classificati nel task 1 come casi anormali, è rappresentata dalla matrice di confusione in Fig.8.2. Combinando i risultati del primo task e del secondo task, è possibile ottenere la matrice di confusione totale mostrata in Fig. 8.3.

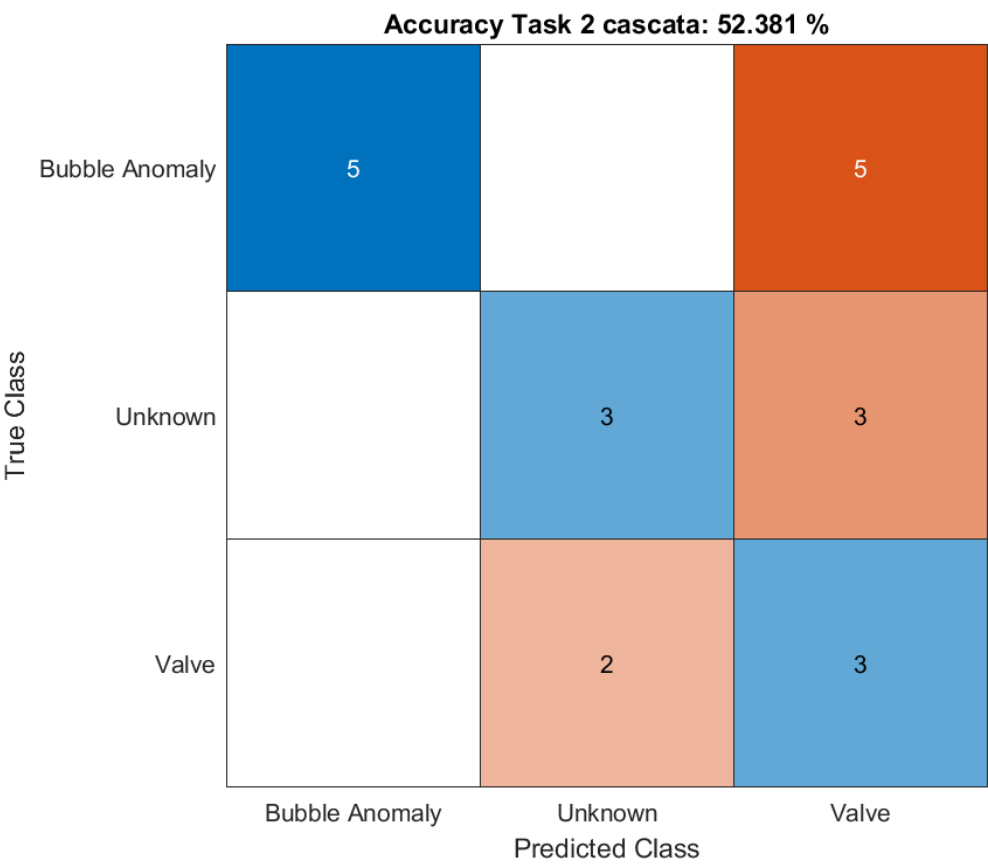


Figura 8.2: Accuracy Task 2 metodo classificatori in cascata

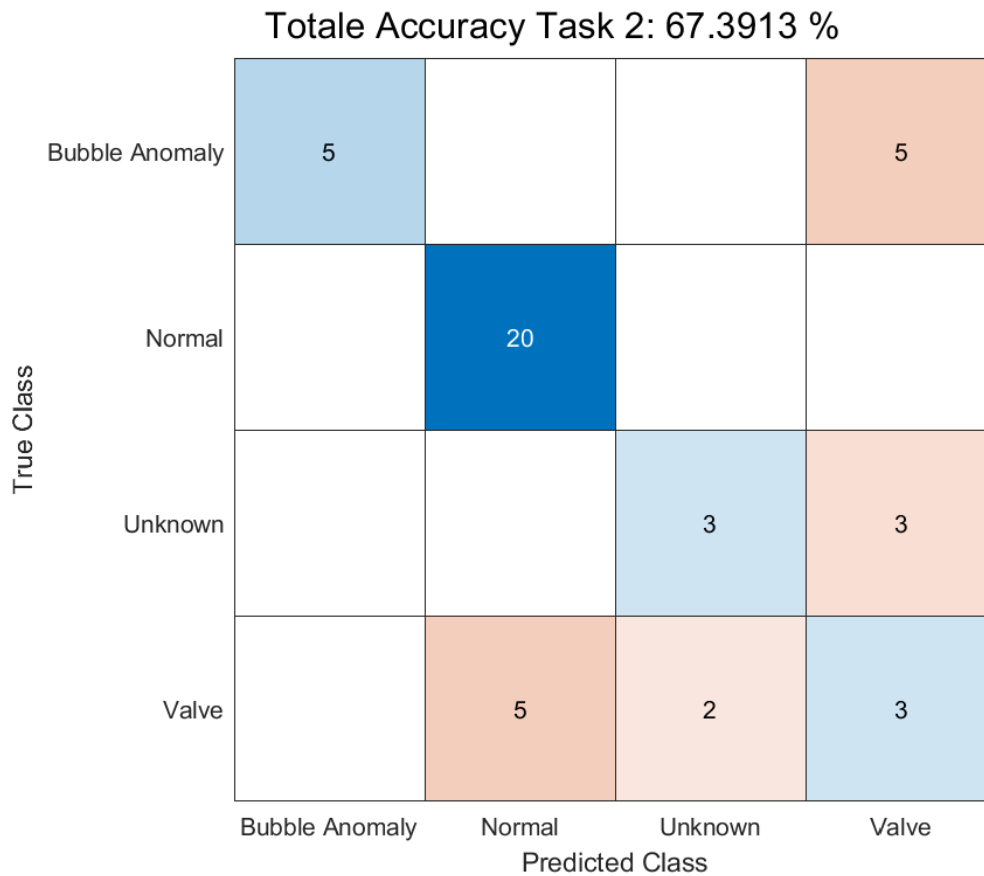


Figura 8.3: Accuracy Totale Task 2 metodo classificatori in cascata

8.2.2 Classificatore one-class e binario

Il testing del metodo caratterizzato dalla combinazione dei classificatori one-class e binario in cascata è stato eseguito in pochi step:

1. Si identificano i dati di test per i quali il task 1 ha predetto un potenziale guasto.
2. Si generano le feature del dataset senza casi di normale funzionamento con la funzione specifica del modello one-class estratta dal Diagnostic Feature Designer.
3. Si applica il modello One-Class SVM precedentemente addestrato per riconoscere eventuali guasti non visti in fase di training.
4. Si generano le feature del dataset senza casi di normale funzionamento e guasti sconosciuti con la funzione specifica del modello binario estratta dal Diagnostic Feature Designer.

5. Si applica il modello binario pre-addestrato sui dati di test classificati come "guasti consociuti" per classificare i dati in guasti di bolle (etichetta '2') o guasti alle valvole (etichetta '3').

L'accuratezza ottenuta con tale tecnica, considerando solo i dati classificati nel task 1 come casi anormali, è rappresentata dalla matrice di confusione in Fig.8.4. Combinando i risultati del primo task e del secondo task, è possibile ottenere la matrice di confusione totale mostrata in Fig. 8.5. In questo caso, l'accuratezza diminuisce a causa dei 5 casi classificati erroneamente dal primo classificatore come casi di normale funzionamento.

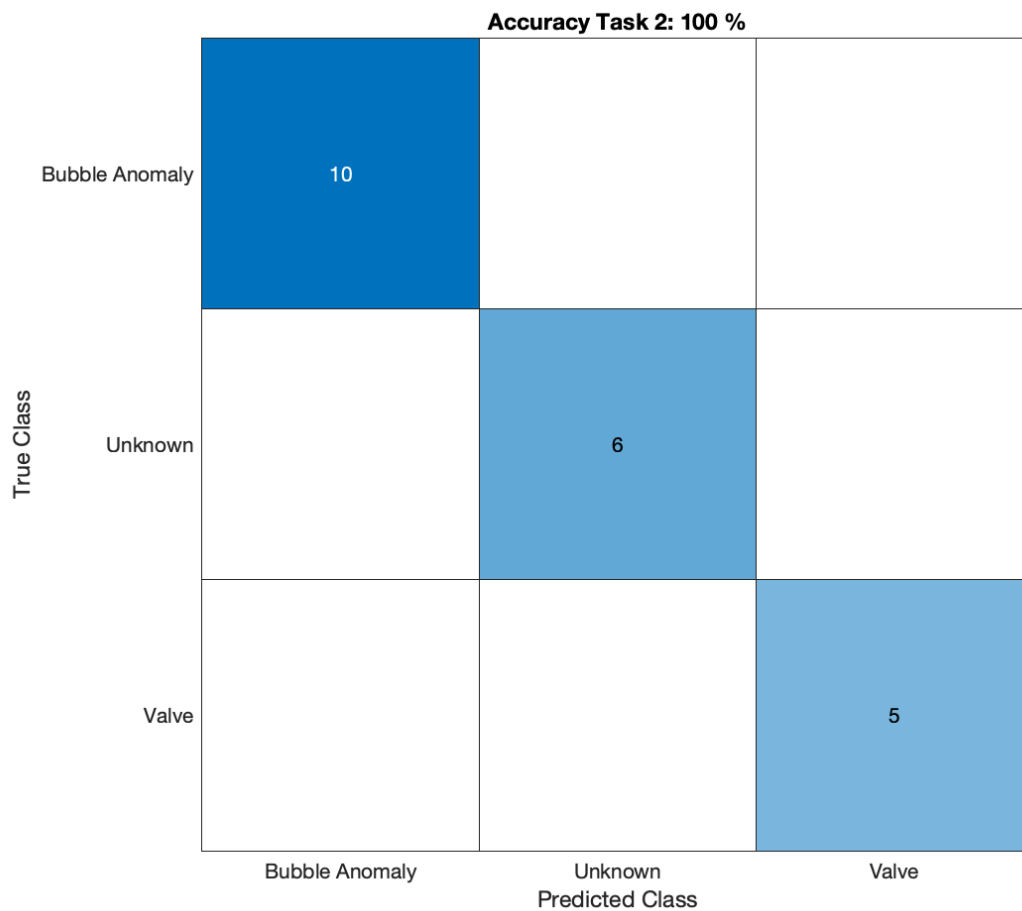


Figura 8.4: Accuracy Task 2 metodo One-Class+Binario

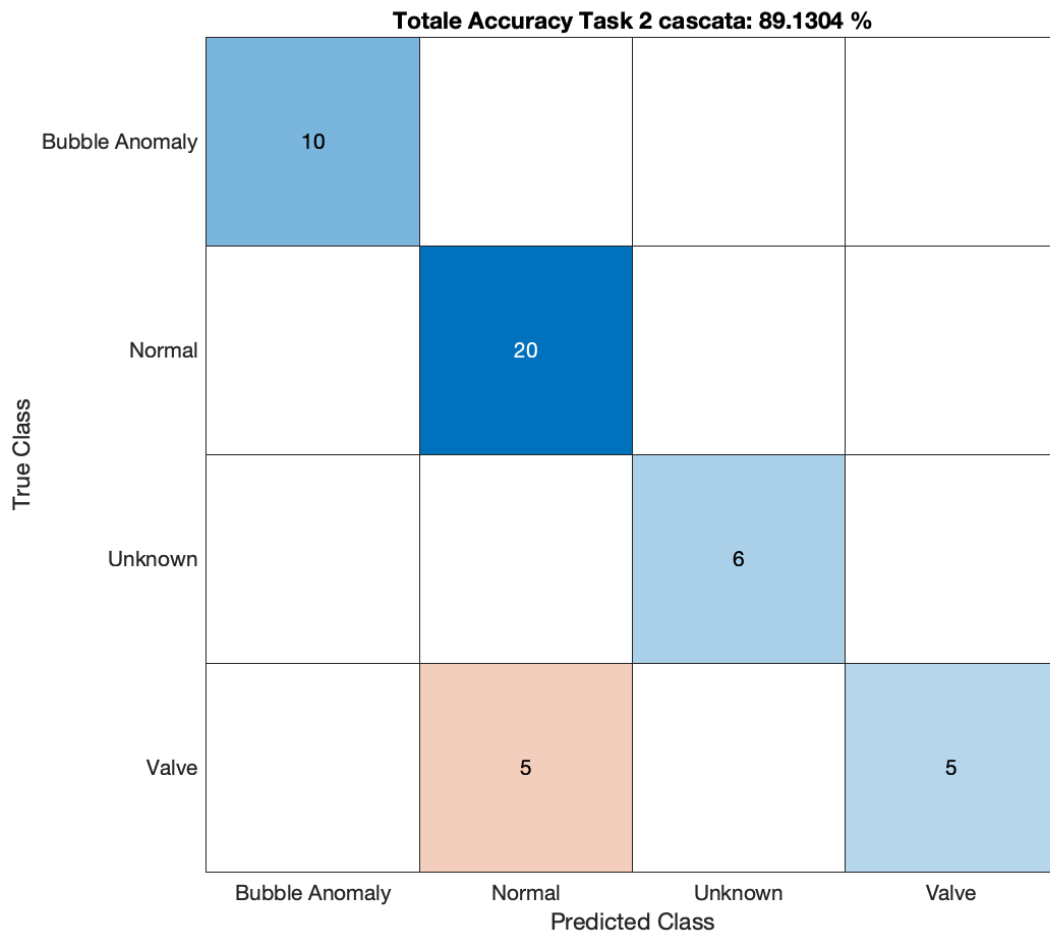


Figura 8.5: Accuracy Totale Task 2 metodo One-Class+Binario

8.2.3 Confronto risultati

Entrambe le soluzioni presentate per l'identificazione dei guasti hanno i loro punti di forza e di debolezza. Tuttavia, la seconda soluzione, che combina un classificatore one-class con un classificatore binario, emerge come scelta preferibile per diversi motivi:

- *Maggior robustezza alle anomalie sconosciute:* La prima soluzione, basata su classificatori one-class in cascata, soffre di una propagazione dell'errore. Un errore commesso in un classificatore iniziale può influenzare negativamente i successivi, riducendo l'accuratezza complessiva nella classificazione dei guasti sconosciuti. La seconda soluzione, invece, utilizza un classificatore one-class per identificare genericamente le anomalie, indipendentemente dal tipo specifico. Questo approccio è più robusto nel distinguere tra dati normali e anomali, anche in presenza di guasti sconosciuti non presenti nei dati di training.

- *Migliore discriminazione tra i tipi di guasto conosciuti*: Una volta identificata un'anomalia, la seconda soluzione impiega un classificatore binario per distinguere tra i due tipi di guasto conosciuti ("Guasto 2" e "Guasto 3"). Questo approccio mirato permette di ottenere una classificazione più precisa dei guasti specifici, riducendo la probabilità di errori di classificazione.
- *Struttura più semplice e flessibile*: La seconda soluzione, con solo due modelli (one-class e binario), vanta una struttura più semplice e intuitiva rispetto alla prima, che richiede una cascata di classificatori. Questa semplicità facilita la comprensione del processo di classificazione e la sua implementazione. Inoltre, la struttura modulare permette di sostituire più facilmente uno dei due modelli con un algoritmo alternativo, se necessario, per migliorare le prestazioni o adattarsi a nuove esigenze.
- *Risultati visibilmente migliori*: La combinazione di un classificatore one-class e uno binario dimostra la sua efficacia nel rilevare sia le anomalie conosciute che quelle sconosciute con un'accuracy del 100%.

8.3 Task 3

Per il testing del task 3, è stato impiegato il modello basato sull'algoritmo di classificazione *Ensemble Bagged Trees*, il quale è stato configurato in base ai risultati della fase di validazione.

Per valutare i risultati sui dati di test, vengono discusse due matrici di confusione. La prima è stata calcolata seguendo il formato di consegna richiesto dalla challenge, mentre la seconda considera solo i dati classificati dai modelli addestrati e presentati nelle sezioni precedenti. Questo perché la competizione prevede una classificazione in cascata, dove i risultati dei task precedenti influenzano quelli dei task successivi.

Nel task 3, solo i dati classificati come anomalie nelle bolle nel task 2 vengono considerati per la classificazione nelle diverse classi di anomalie. Tutti gli altri casi, inclusi quelli classificati come normali nel task 1 e quelli classificati come guasti alle valvole, sono etichettati come classe 0, ossia *altri* nella matrice di confusione.

Le matrici di confusione ottenute sono presentate nella Fig.8.6. In entrambi i casi, l'accuratezza è quasi ottima, con solo un errore che classifica erroneamente un dato come appartenente alla classe *BV1* anziché *BP1*. Si noti che queste due posizioni delle bolle sono in serie.

L'accuratezza complessiva, ottenuta seguendo le direttive impostate dalla challenge, risulta essere del 97.82

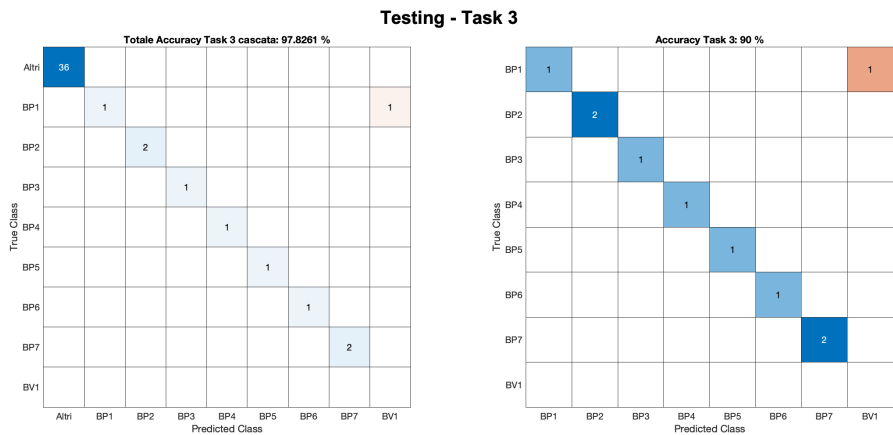


Figura 8.6: Matrice di confusione per il task 3 sui dati di test

8.4 Task 4

Il testing del task 4 è stato eseguito a partire dai dati di test classificati dal task 2 come casi di guasto alle valvole (etichetta '3').

Quindi, sono state generate le feature di tali dati utilizzando la funzione specifica generata in fase di addestramento dall'app Diagnostic Feature Designer.

È stato eseguito il modello multiclasse pre-addestrato per classificare il posizionamento del guasto tra le quattro valvole.

Per la valutazione dei risultati sui dati di test, come eseguito in precedenza per il task 3, sono state generate due matrici di confusione. La prima, in Fig.8.7, mostra l'accuratezza del modello addestrato a riconoscere il posizionamento del guasto tra le quattro valvole. La seconda, in Fig.8.8, è stata calcolata seguendo il formato di consegna richiesto dalla challenge; poichè il classificatore è addestrato sui soli casi di guasti, tutti gli altri casi, inclusi quelli classificati come normali nel task 1 e quelli classificati come anomalie di bolle, sono etichettati come classe 0, ossia altri nella matrice di confusione.

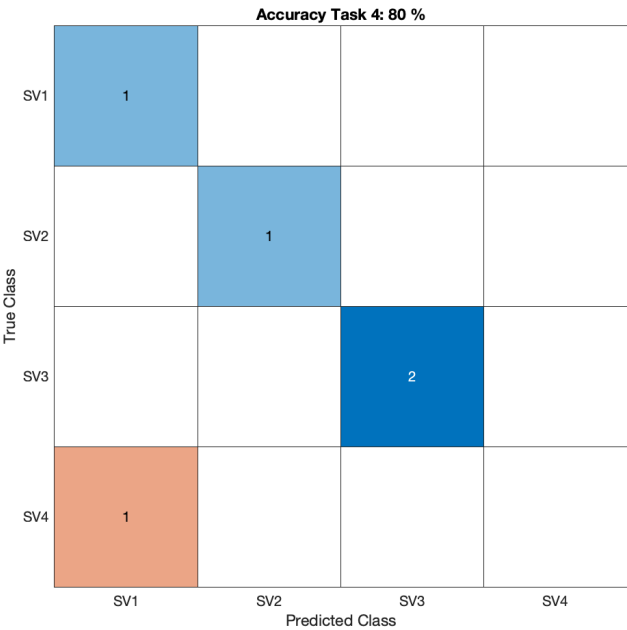


Figura 8.7: Accuracy Task 4

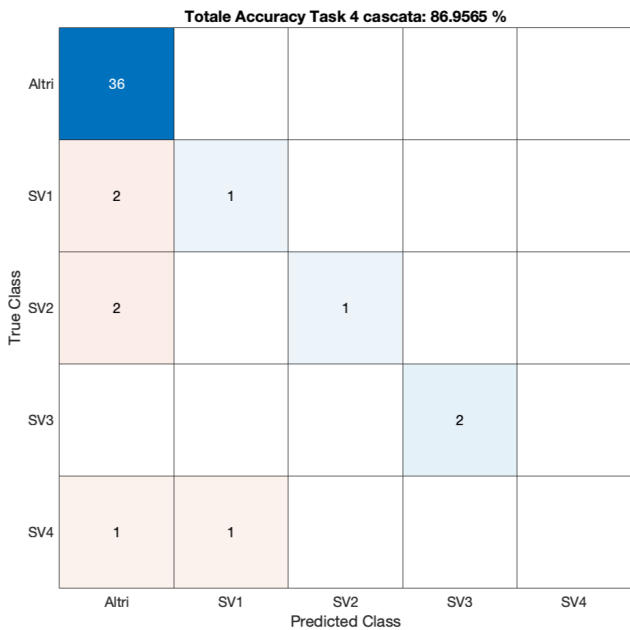


Figura 8.8: Accuracy Totale Task 4

8.5 Task 5

Il testing del task 5 è stato eseguito a partire dai dati di test classificati dal task 2 come casi di guasto alle valvole (etichetta '3') al fine di valutare la percentuale di apertura delle valvole.

Quindi, sono state generate le feature di tali dati utilizzando la funzione estratta dall'app Diagnostic Feature Designer per il rankig di tipo Trendability.

È stato eseguito il modello di regressione generato in precedenza.

Infine, è stata calcolata l'accuratezza del modello che viene mostrata dal grafico in Fig. 8.9.

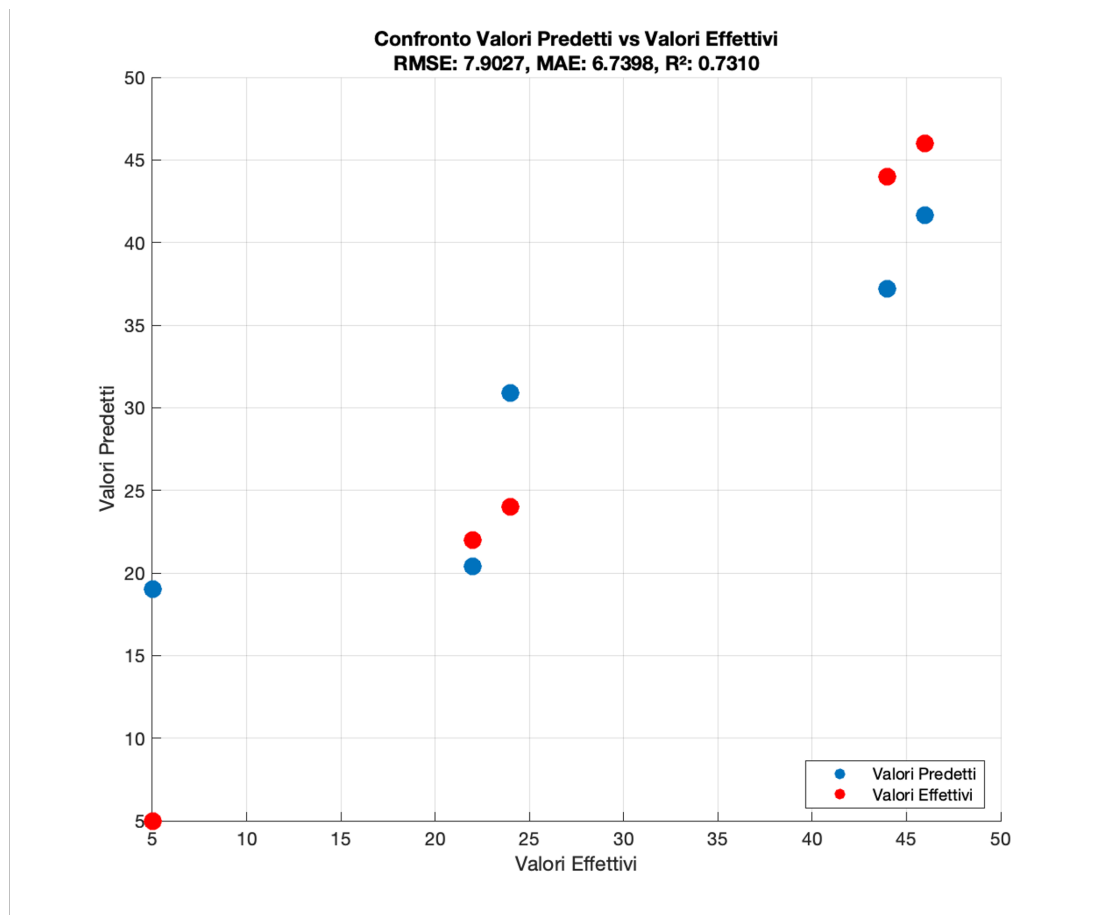


Figura 8.9: Grafico rappresentante lo scostamento tra valori reali e predetti nel task 5 con le rispettive metriche.

Oltre alla RMSE (Errore Quadratico Medio), metrica usata anche in fase di validation del modello, vengono riportati anche il MAE (Errore Assoluto Medio) e il coefficiente di determinazione R^2 .

Capitolo 9

Metrica di valutazione

Per valutare le prestazioni del sistema di diagnosi è stato implementato un metodo conforme alle specifiche della challenge. In particolare, il metodo si basa su diverse metriche che valutano la corretta classificazione e identificazione di diverse anomalie. In sintesi, la metrica valuta i seguenti aspetti:

1. Task 1 (Classificazione Normale/Anormale): la corretta classificazione viene premiata con *10 punti*;
2. Task 2 (Classificazione Anormale): per la corretta identificazione della condizione specifica di guasto (anomalia contaminazione bolle/guasto elettrovalvola/guasto sconosciuto) nei dati correttamente classificati come anormali nel Task 1 assegna *10 punti*;
3. Task 3 (Identificazione della posizione della bolla): per la corretta identificazione della posizione della bolla nei dati correttamente classificati come contaminati da bolla nel Task 2, sono assegnati *10 punti*;
4. Task 4 (Identificazione del guasto sulla valvola): per la corretta identificazione della posizione della valvola guasta nei dati correttamente classificati come guasti sulla valvola nel Task 2, sono assegnati *10 punti*;
5. Task 5 (Previsione grado di apertura della valvola guasta): per la corretta previsione del grado di apertura della valvola guasta nei dati correttamente classificati come guasti sulla valvola nel Task 2, il punteggio è calcolato come $\max(|\text{valore reale} - \text{previsione}| + 20, 0)$. Questa formula incentiva l'accuratezza entro una tolleranza di 20 unità. Le previsioni entro questa tolleranza ricevono un punteggio positivo, che diminuisce man mano che l'errore aumenta, fino a un massimo di 20. Le previsioni con errori superiori a 20 unità ricevono un punteggio di 0, indicando una performance insoddisfacente.

Per il veicolo spaziale-4, i punteggi ottenuti sono raddoppiati, in considerazione della maggiore difficoltà.

Il punteggio totale massimo ottenibile è 1680, suddiviso in base alla complessità di ogni task:

- Task 1: 690 punti
- Task 2: 390 punti
- Task 3: 150 punti
- Task 4: 150 punti
- Task 5: 300 punti

Il sistema di diagnosi implementato ha ottenuto un punteggio totale di 1192.62, con la seguente distribuzione:

- 610 punti su 690 ottenuti nel Task 1;
- 310 punti su 390 ottenuti nel Task 2;
- 130 punti su 150 ottenuti nel Task 3;
- 50 punti su 150 ottenuti nel Task 4;
- 92.62 punti su 300 ottenuti nel Task 5.

Il punteggio finale è stato calcolato rapportando il punteggio ottenuto al punteggio massimo, ottenendo così una valutazione complessiva del 70.99%. È evidente che un peso significativo nella valutazione finale è stato dato dalle performance del modello di classificazione del task 4 e dal modello di regressione del task 5, dove sono stati ottenuti meno della metà dei punti totali previsti. Nel Task 1, infatti, il sistema ha commesso errori solo nei 5 casi di anomalia sulle valvole, il che ha avuto un impatto negativo anche sui successivi task relativi a tale anomalia.

Elenco delle figure

1.1	Schema del sistema di propulsione sperimentale	4
1.2	Profilo di pressione tipico estratto dal modello	5
2.1	Struttura del dataset	7
2.2	Struttura del dataset	11
3.1	Spettro di potenza per il segnale P1	14
3.2	Spettro di potenza per il segnale P2	14
3.3	Spettro di potenza per il segnale P3	15
3.4	Spettro di potenza per il segnale P4	15
3.5	Spettro di potenza per il segnale P5	16
3.6	Spettro di potenza per il segnale P6	16
3.7	Spettro di potenza per il segnale P7	17
3.8	Selezione feature task 1	19
3.9	Matrice di confusione per il task 1 sui dati di validazione	20
4.1	Matrice di confusione del modello one-class in cascata	23
4.2	Matrice di confusione del modello binario	25
5.1	Selezione feature task 3	27
5.2	Matrice di confusione per il task 3 sui dati di validazione	28
6.1	Matrice di confusione del modello per Task 4	30
7.1	Risultati validation ranking di tipo Trendability con feature integrali.	32
7.2	Risultati validation ranking di tipo Trendability con feature ridotte.	33
8.1	Matrice di confusione per il task 1 sui dati di test	35
8.2	Accuracy Task 2 metodo classificatori in cascata	37
8.3	Accuracy Totale Task 2 metodo classificatori in cascata	38
8.4	Accuracy Task 2 metodo One-Class+Binario	39
8.5	Accuracy Totale Task 2 metodo One-Class+Binario	40
8.6	Matrice di confusione per il task 3 sui dati di test	42
8.7	Accuracy Task 4	43
8.8	Accuracy Totale Task 4	43

8.9	Grafico rappresentante lo scostamento tra valori reali e predetti nel task 5 con le rispettive metriche.	44
-----	---	----