

**ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA**

SCUOLA DI INGEGNERIA
- Sede di Forlì -

***CORSO DI LAUREA*
IN INGEGNERIA MECCANICA**

Classe: L-9

ELABORATO FINALE DI LAUREA
in Laboratorio di motori a combustione interna

**“MODELLAZIONE DI INDICI SINTETICI DI COMBUSTIONE PER
MOTORI ENDOTERMICI ALTERNATIVI CON ALGORITMI DATA-
DRIVEN DI ENSEMBLE LEARNING”**

Candidato:
Yasmine falco

Relatore:
Prof. Alessandro Brusa

Anno accademico 2023/2024

Ringraziamenti

Vorrei ringraziare di cuore il prof. Alessandro Brusa, che si è dimostrato lungo questo percorso di tesi una guida su cui poter contare e che mi ha permesso di approfondire e osservare da più vicino ciò di cui sono più appassionata.

Ringrazio i miei genitori che mi hanno supportato sin dall'inizio, senza mai dubitare delle mie capacità quando anche io sotto sotto lo facevo. Mi hanno fatto sentire un "ingegnere meccanico" (anche se per loro sarò per semplicità un meccanico – risata) sin dal primo giorno di università, e che mi ha dato la possibilità di poter esserlo oggi. Ringrazio, ovviamente, anche i miei fratelli se no si offendono. Dai su scherzavo.

Grazie nonna di aver creduto in me sempre e soprattutto durante questo percorso, le tue rassicurazioni e parole piene d'affetto mi hanno incoraggiato quando ne avevo più bisogno.

Gianluca, un grazie va anche a te che mi hai supportato e sopportato con tanta pazienza in questi ultimi due anni. Ti ringrazio dell'aiuto e dell'affetto che mi hai dato nei giorni in cui volevo solo sprofondare nel letto.

Ringrazio tutti gli amici che ho trovato durante questo corso di laurea, quando iniziai mai avrei immaginato di trovarne così tanti, avete reso le ore di lezione infinite molto più belle e divertenti. Mi mancheranno le nostre solite prime due file.

Infine, ringrazio me stessa. Non avete idea di quante volte ho dubitato delle mie capacità non sentendomi all'altezza di questo percorso, ma ho vinto queste insicurezze dando sempre il massimo. Ce l'abbiamo fatta.

INDICE

1. Introduzione alla modellazione mediante Ensemble learning	9
1.1. Modellazione data-driven	9
1.2. Ensemble learning	9
1.2.1. Principali metodologie di ensemble utilizzate	10
1.2.2. Altri metodi di Ensemble learning	14
1.2.3. Pro e Contro dell'Ensemble Learning	15
2. Analisi dei dati	16
2.1. Dati input modelli	16
2.2. Tipologie dati	19
2.2.1. Data Train.....	19
2.2.2. Data Test.....	21
3. MFB50	24
4. Script Matlab	30
5. Risultati	34
6. Conclusioni	42
Riferimenti e Bibliografia	43

ABSTRACT

L'obiettivo di questa tesi è esplorare l'applicazione dell'ensemble learning per la modellazione dei motori a combustione interna, con particolare attenzione alla previsione dell'indice di fase di combustione, MFB50. Il lavoro si basa su dati sperimentali ottenuti da test al banco di un motore Ferrari V8 GDI bi-turbo ad alte prestazioni, equipaggiato con sensori per la misurazione di parametri chiave come la pressione nei cilindri. Sono stati condotti diversi test, tra cui VVT (Variable Valve Timing) sweep, Lambda sweep e SA (spark advance) sweep, per garantire che il modello potesse operare efficacemente in condizioni dinamiche.

Sono stati utilizzati due tipi principali di dataset per la progettazione e l'addestramento dei modelli data-driven: registrazioni continue del motore e misurazioni stazionarie dei punti di funzionamento, con i dati mediati tra i cilindri 5, 6 e 7 (il cilindro 8 è stato escluso poiché presentava un comportamento leggermente differente). Questi dati includevano il target degli RPM, il carico relativo, l'anticipo dell'accensione e il rapporto aria/carburante (λ). Il numero di alberi decisionali, l'iperparametro principale del modello di ensemble learning, è stato ottimizzato tramite uno script dedicato. L'intero processo è stato condotto nell'ambiente MATLAB.

La particolarità dei modelli che verranno presentati in seguito è legata all'approccio utilizzato per l'addestramento. Invece di allenarli su dati continui, che sono spesso complessi e dispendiosi da ottenere, l'addestramento è stato eseguito su dati stazionari, i quali risultano molto più semplici da raccogliere e gestire. Questo approccio consente comunque di ottenere modelli capaci di predire con accuratezza dati continui, sfruttando una base di addestramento più facilmente accessibile e meno onerosa in termini di risorse. Alla fine, il valore predetto di MFB50 è stato confrontato con quello misurato, e il modello con il minor errore quadratico medio è risultato il più accurato.

Capitolo 1

1. Introduzione alla modellazione mediante Ensemble learning

1.1. Modellazione data-driven

La modellazione data-driven dei motori a combustione interna si basa sull'utilizzo di dati sperimentali per costruire modelli matematici capaci di descrivere il comportamento del motore. A differenza delle tecniche fisico-matematiche tradizionali, che richiedono una conoscenza approfondita della dinamica dei processi di combustione e del funzionamento meccanico, la modellazione data-driven si affida a metodi statistici e algoritmi di machine learning per apprendere direttamente dai dati raccolti. Questa tecnica è particolarmente utile per gestire la complessità e la non linearità dei fenomeni legati alla combustione, come la previsione degli indici di combustione e delle emissioni. I modelli data-driven possono migliorare l'efficienza del motore, prevedere con precisione il comportamento in diversi scenari operativi e contribuire allo sviluppo di sistemi di controllo più avanzati, con il vantaggio di adattarsi a nuovi dati senza dover modificare le leggi fisiche sottostanti.

1.2. Ensemble learning

Il machine learning è un ramo dell'intelligenza artificiale che permette ai computer di apprendere dai dati e migliorare le loro prestazioni su compiti specifici senza essere esplicitamente programmati. Utilizza algoritmi che riconoscono schemi all'interno di grandi quantità di dati, consentendo ai modelli di fare previsioni o prendere decisioni basate su queste informazioni. Le applicazioni spaziano dalla classificazione delle immagini, alla previsione delle tendenze, all'analisi del linguaggio naturale e molto altro.

Un sottoinsieme importante del machine learning è l'ensemble learning, che combina più modelli base (ad esempio, alberi decisionali) per migliorare l'accuratezza delle previsioni rispetto all'utilizzo di un singolo modello. Il principio chiave dell'ensemble learning è che un insieme di modelli può catturare meglio la variabilità nei dati, riducendo gli errori di generalizzazione. Tecniche comuni di ensemble includono Bagging (come nel metodo Random Forest), che addestra più modelli indipendenti e ne combina le previsioni, e Boosting, che costruisce sequenzialmente modelli correttivi, come nel caso di Least Squares Boosting. L'ensemble learning risulta particolarmente efficace perché permette di mitigare i limiti e i bias di singoli modelli, fornendo soluzioni più robuste e precise.

I metodi ensemble vengono spesso esemplificati utilizzando gli alberi decisionali, poiché questi algoritmi possono facilmente incorrere in problemi di over fitting (elevata varianza e bassa distorsione) se non adeguatamente ottimizzati. Allo stesso tempo, possono essere soggetti a under fitting (bassa varianza e alta distorsione) quando sono di dimensioni ridotte, come nel caso di un "decision stump", ossia un albero decisionale con un solo livello. Quando un algoritmo si adatta eccessivamente o insufficientemente ai dati di addestramento, non riesce a generalizzare efficacemente su nuovi set di dati. Per affrontare questo problema, si utilizzano tecniche ensemble che migliorano la capacità del modello di generalizzare a dati non visti.

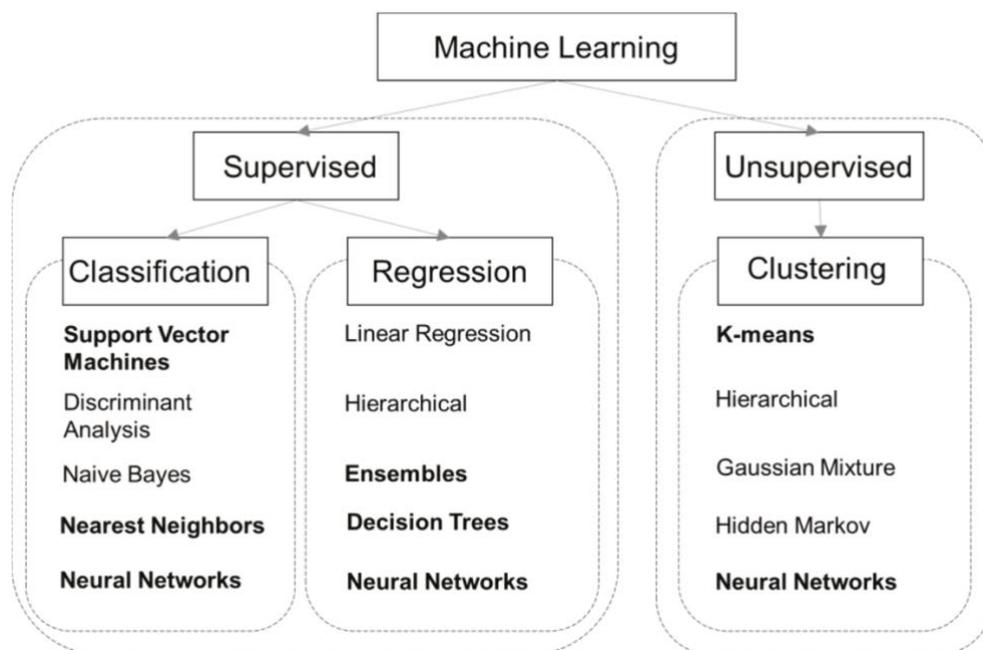


Figura 1-1: Tecniche di machine learning

1.2.1. Principali metodologie di ensemble utilizzate

Il Bagging e Boosting sono due tipologie principali di metodi di apprendimento d'insieme. La differenza principale tra questi metodi di apprendimento è il modo in cui i modelli vengono addestrati.

Nel Bagging, gli allievi deboli vengono addestrati in parallelo, invece nel Boosting apprendono in sequenza. Ciò significa che viene costruita una serie di modelli e, ad ogni nuova iterazione del modello, vengono aumentati i pesi dei dati erroneamente classificati nel modello precedente. Questa redistribuzione dei pesi aiuta l'algoritmo a identificare i parametri su cui deve focalizzarsi per migliorare le sue prestazioni.

I metodi di Bagging sono in genere utilizzati su allievi deboli che presentano elevata varianza e bassa distorsione, mentre i metodi di Boosting vengono sfruttati quando si osserva una bassa varianza e un'elevata distorsione.



Figura 1-2: Funzione Matlab per costruire un modello di ensemble learning

Bagging

Noto anche come aggregazione di bootstrap, è il metodo di apprendimento d'insieme comunemente utilizzato per ridurre la varianza in un set di dati rumorosi. Nel Bagging, un campione casuale di dati in un set di addestramento è selezionato con sostituzione, il che significa che i singoli punti di dati possono essere scelti più di una volta. Dopo che sono stati generati diversi campioni di dati, questi modelli vengono addestrati in modo indipendente e, a seconda del tipo di attività - ossia regressione o classificazione - la media o la maggioranza di queste previsioni produce una stima più accurata. È formato da tre passaggi di base:

- **Bootstrapping:** il Bagging utilizza efficacemente una tecnica di campionamento di bootstrapping per creare campioni eterogenei. Questo metodo di ricampionamento genera diversi sottoinsiemi di set di dati di addestramento selezionando punti dati in modo casuale e con sostituzione. Questo significa che, ogni volta che si seleziona un punto dati dal set di dati di addestramento, puoi selezionare la stessa istanza più volte. Come risultato, un valore/istanza viene ripetuto due volte (o più) in un campione.
- **Addestramento parallelo:** questi campioni di bootstrap vengono quindi addestrati in modo indipendente e in parallelo tra loro utilizzando allievi deboli o di base.
- **Aggregazione:** infine, a seconda dell'attività (cioè regressione o classificazione), vengono ricavate una media o una maggioranza delle previsioni per calcolare una stima

più accurata. Nel caso della regressione, viene calcolata una media di tutti gli output previsti dai classificatori individuali; tale operazione è nota come "soft voting".

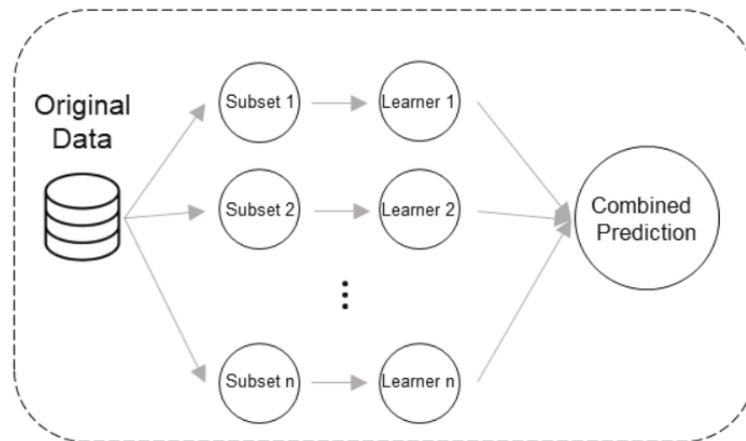


Figura 1-3: Schema grafico sul funzionamento del Bagging

▪ Random forest

Il metodo Random Forest estende il Bagging combinando la casualità delle caratteristiche e la tecnica di bootstrap per creare una foresta di alberi decisionali non correlati. La chiave di questa tecnica risiede nella selezione casuale di un sottoinsieme di caratteristiche, che garantisce una bassa correlazione tra gli alberi. Questa differenza è fondamentale rispetto agli alberi decisionali tradizionali, che considerano tutte le possibili suddivisioni. Gli algoritmi Random Forest, invece, scelgono solo una parte delle caratteristiche, rendendo il modello più robusto e meno incline all'overfitting.

Tre iperparametri principali devono essere configurati prima dell'addestramento: la dimensione dei nodi, il numero di alberi e il numero di caratteristiche campionate. Una volta impostati, il classificatore Random Forest può essere utilizzato per problemi sia di classificazione che di regressione. Nel nostro caso, il modello affronta una regressione.

Ogni albero nella foresta viene costruito su un campione di dati prelevato con sostituzione dal set di addestramento, noto come esempio di bootstrap. Un terzo di questi dati viene escluso per essere utilizzato come campione "out-of-bag" (oob), che serve per validare il modello. La casualità delle caratteristiche tramite Bagging aumenta ulteriormente la diversità tra gli alberi, riducendo la correlazione tra di essi e migliorando la generalizzazione.

Nel caso di un problema di regressione, la previsione finale è ottenuta calcolando la media delle predizioni degli alberi. Il campione oob viene poi usato per la convalida incrociata, verificando l'accuratezza del modello senza la necessità di un set di test separato.

I Random Forest sono particolarmente efficaci nel gestire dati con molte variabili e rumore, rendendoli ideali per scenari complessi e ad alta dimensionalità.

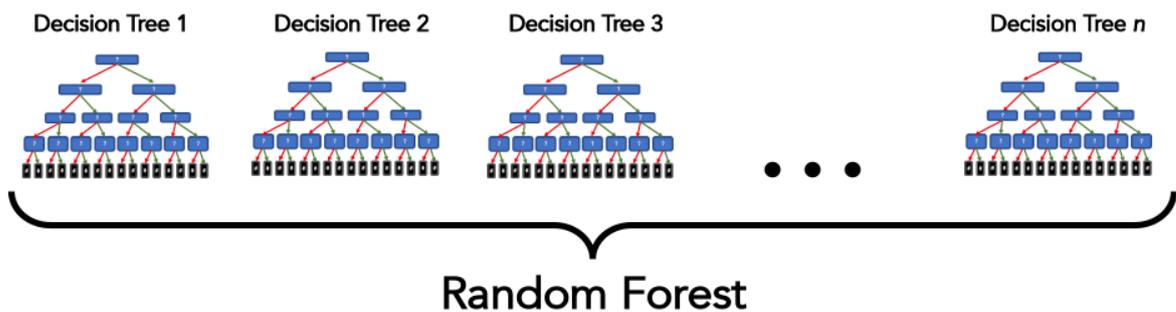


Figura 1-4: Schema grafico sul funzionamento del Random Forest

Boosting

È un metodo di apprendimento d'insieme che combina una serie di "allievi" deboli in un "allievo" forte per ridurre al minimo gli errori di addestramento.

Nel processo di boosting, si crea inizialmente un sottoinsieme casuale dal dataset originale, assegnando a tutti i punti dati pesi uguali. Un modello base viene costruito su questo sottoinsieme e viene utilizzato per fare previsioni sull'intero dataset. Successivamente, gli errori vengono calcolati confrontando i valori reali con quelli previsti dal modello. Le osservazioni che sono state classificate in modo errato ricevono pesi più elevati. A questo punto, viene creato un secondo modello, che prova a correggere gli errori commessi dal primo. Questo processo si ripete: ogni nuovo modello cerca di correggere gli errori del modello precedente. Alla fine, il modello finale, chiamato "strong learner", è una media pesata di tutti i modelli precedenti, noti come "weak learners". In questo modo, l'algoritmo di boosting combina diversi weak learners per formare uno strong learner. Sebbene i singoli modelli non siano in grado di fornire buone prestazioni su tutto il dataset, funzionano bene su specifiche parti, contribuendo ciascuno a migliorare le prestazioni complessive dell'ensemble.

Uno dei metodi utilizzati nello specifico è l'**LSBoost** (Least-Squares Boosting) che è una variante che si applica agli ensemble di regressione. Ad ogni passo, l'ensemble si adatta alla differenza tra la risposta osservata e la previsione aggregata di tutti i modelli precedenti. L'obiettivo di LSBoost è minimizzare l'errore quadratico medio nel processo di adattamento.

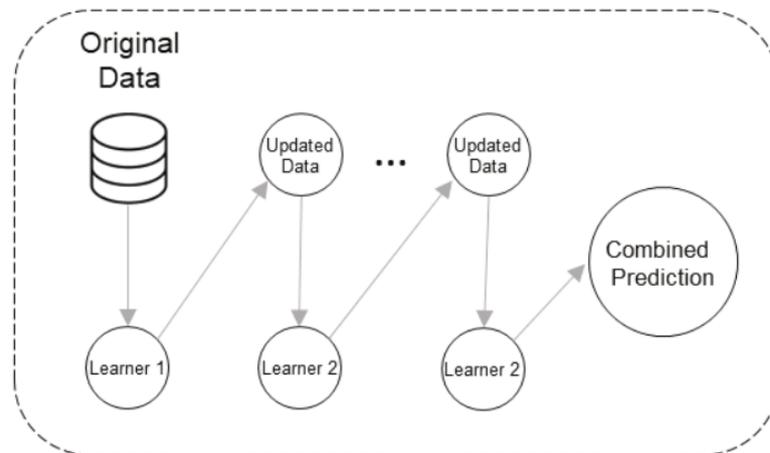


Figura 1-5: Schema grafico sul funzionamento del Boosting

1.2.2. Altri metodi di Ensemble learning

Stacking

Lo Stacking, o Stacked Generalization, è una tecnica che combina diversi modelli di apprendimento attraverso un metamodello di livello superiore. I modelli di base effettuano le loro predizioni, che vengono poi utilizzate come input dal metamodello. Quest'ultimo cerca di migliorare ulteriormente le previsioni combinando i risultati ottenuti dai modelli di base. Un vantaggio di questa tecnica è la possibilità di combinare modelli diversi, come alberi decisionali, modelli di regressione e reti neurali, per sfruttare i rispettivi punti di forza. Tuttavia, uno svantaggio è che lo Stacking richiede una fase aggiuntiva di ottimizzazione e addestramento per il metamodello, aumentando così la complessità computazionale complessiva.

Voting

Il metodo del Voting si basa sull'utilizzo di diversi modelli di base, dove la decisione finale viene presa attraverso un processo di votazione. Esistono due tipi principali di Voting: il Voting Maggioritario, in cui ogni modello fornisce una predizione e la classe che riceve più voti diventa la predizione finale, e il Voting Ponderato, dove ogni modello ha un peso specifico e la decisione finale si basa sulla somma dei voti ponderati. Questo metodo ha il vantaggio di essere semplice e veloce da implementare, tuttavia, non sempre garantisce un miglioramento delle prestazioni se i modelli di base non sono adeguatamente calibrati.

1.2.3. Pro e Contro dell'Ensemble Learning

Vantaggi

- i. **Maggiore Accuratezza:** L'Ensemble Learning, combinando modelli diversi, tende a fornire una maggiore accuratezza rispetto ai singoli modelli.
- ii. **Riduzione dell'Overfitting:** Il Bagging, ad esempio, può ridurre significativamente il rischio di overfitting, soprattutto su modelli instabili come gli alberi decisionali.
- iii. **Maggiore Robustezza:** Essendo basato su una pluralità di modelli, l'ensemble può essere più robusto ai dati rumorosi o a set di dati difficili.
- iv. **Versatilità:** Può essere applicato a modelli diversi (alberi decisionali, regressori lineari, modelli non lineari, ecc.) e adattato a vari tipi di dati.

Svantaggi

- i. **Complessità Computazionale:** L'addestramento di più modelli può richiedere molte risorse computazionali e tempo.
- ii. **Difficoltà di Interpretazione:** La combinazione di più modelli può rendere difficile interpretare le decisioni finali, rendendo il sistema una "black box".
- iii. **Suscettibilità all'Overfitting (Boosting):** In alcuni casi, specialmente con il Boosting, se i modelli base sono troppo complessi, l'ensemble può overfittare i dati.
- iv. **Maggiore Manutenzione:** La gestione di diversi modelli implica una maggiore attenzione nella fase di aggiornamento e manutenzione del sistema.

Capitolo 2

2. Analisi dei dati

2.1. Dati di input

Le grandezze che incidono maggiormente sul ciclo di pressione, e quindi sull'MFB50, si dividono in due famiglie. La prima racchiude tutte le grandezze di natura fluidodinamica, mentre la seconda quelle di natura chimica. Le grandezze di natura fluidodinamica incidono sul moto della miscela e sulla massa che viene racchiusa nel cilindro. Le grandezze di natura chimica, invece, interessano strettamente il processo di combustione. Le grandezze che influenzano maggiormente la fluidodinamica della miscela sono il coefficiente di riempimento del cilindro e la velocità di rotazione dell'albero motore. Esse producono effetti anche sulla combustione sviluppata nel cilindro, ma tali effetti sono riconducibili al moto della carica.

- Il target degli RPM (giri motore) viene definito dall'operatore attraverso il software del banco prova e trasmesso al freno a correnti parassite (freno dinamometrico), il quale è accoppiato al motore e opportunamente controllato. Il freno è in grado di applicare una coppia resistente modulabile, così da riprodurre le condizioni operative a cui il componente in prova deve essere sottoposto. La velocità di rotazione dell'albero motore influisce sul moto della carica entrante nel cilindro. All'aumentare dei giri motore, infatti, si genera una maggiore turbolenza nel cilindro, il che favorisce la stabilità della combustione. In particolare, un aumento del moto turbolento con un asse perpendicolare rispetto a quello del cilindro accelera la velocità di combustione, migliorando l'efficienza. Di conseguenza, la pressione massima sviluppata all'interno del cilindro aumenta, a parità di aria intrappolata, anticipo e rapporto di miscela (λ). Durante il funzionamento del motore, la velocità e la posizione angolare sono costantemente monitorate grazie all'utilizzo di sensori induttivi o sensori ad effetto Hall.
- Il carico di riempimento r_l (simile a un'efficienza volumetrica) è definito dal rapporto tra la massa di fluido che effettivamente entra dal cilindro e la massa di fluido che teoricamente potrebbe entrare in condizioni stazionarie alla pressione e alla temperatura di riferimento.

$$rl = \frac{m_r}{m_t}$$

La pressione e la temperatura scelte come riferimento possono essere quelle di collettore oppure quelle relative all'ambiente esterno. Nel primo caso l' rl prende il nome di coefficiente di riempimento relativo, mentre, nel secondo, coefficiente di riempimento assoluto. Il presente elaborato prende in considerazione solamente il coefficiente di riempimento relativo.

Si discosta quindi dall'unità a causa di tutti quei fenomeni che comportano una variazione della massa aspirata, sia in termini di incremento che di decremento, rispetto alla massa d'aria che entrerebbe in condizioni stazionarie.

La massa intrappolata per ciclo incide sulla pressione sviluppata nel cilindro. Infatti, un elevato rl implica una maggiore pressione media di ciclo. La Figura 2-1 mostra due cicli a rl assoluto differenti. La curva blu, segnale di pressione di un ciclo a 180 di rl , ha tutte e quattro le fasi di ciclo a pressione superiore rispetto alla curva rossa, rl pari a 120. L'effetto su aspirazione e compressione è dovuto semplicemente all'aumento di massa intrappolata a pari volume, mentre l'espansione presenta un picco nettamente superiore perché aumenta la massa di combustibile iniettabile e di conseguenza l'energia rilasciata.

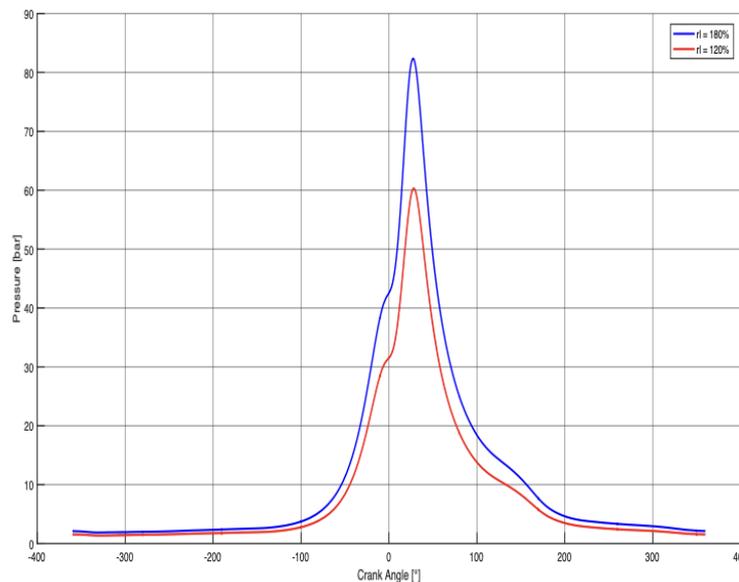


Figura 2-1: Segnale pressione in camera per due cicli a rl differenti

L' rl è sempre noto durante il funzionamento del motore. In centralina, infatti, è presente un modello che permette di stimare il coefficiente di riempimento nota la pressione del collettore di aspirazione e la velocità di rotazione dell'albero. Tale modello viene

calibrato in fase di sviluppo motore e permette di conoscere la massa realmente aspirata per ogni ciclo.

Dal punto di vista chimico le grandezze di maggiore interesse sono l'anticipo di accensione e il rapporto di miscela, λ .

- SA (spark advance), ovvero l'anticipo d'accensione, indica l'angolo di anticipo del pistone rispetto al PMS deciso dall'accensione e dove l'impianto d'accensione fa scoccare la scintilla alla candela. Il posizionamento angolare della fase di combustione, rispetto al movimento dello stantuffo, è uno dei parametri che influenzano maggiormente il valore della coppia erogata da un motore ad accensione comandata. Poiché il suddetto posizionamento dipende dall'istante (angolo di manovella) di accensione (anticipo di accensione rispetto al PMS), quest'ultimo parametro va controllato opportunamente per ottenere la massima prestazione dal motore, in ogni condizione operativa. Generalmente è necessario aumentare l'anticipo all'aumentare del numero di giri, se si ha l'obiettivo di mantenere il baricentro della combustione in posizione ottimale.
- Lambda, il così detto rapporto di miscela, è un indice normalizzato rispetto al rapporto stechiometrico, definito come:

$$\lambda = \frac{AFR}{(AFR)_{st}}$$

Dove AFR indica "air fuel ratio" (ovvero rapporto aria/combustibile) e $(AFR)_{st}$ è il medesimo ma nelle condizioni di stechiometria (ovvero quando le proporzioni tra i reagenti sono tali che nessuno di essi è in eccesso, cioè non ci saranno reagenti residui tra i prodotti della reazione)

Ovviamente $\lambda=1$ in condizioni di stechiometria, $\lambda<1$ in condizioni di eccesso di combustibile (miscela grassa) e $\lambda>1$ in condizioni di eccesso d'aria (miscela magra).

Gli effetti prodotti da queste due grandezze sulla combustione sono dipendenti tra loro e possono essere identificati al variare dell'MFB50. Questo perché anticipo e λ determinano rispettivamente l'inizio e la velocità della combustione. Quindi la stessa fase di combustione può essere ottenuta variando in maniere differentemente le due grandezze. L'effetto sulla combustione determinato da un determinato valore di fase è il medesimo qualsiasi sia la coppia di valori anticipo- λ attuata, per questo motivo studiare

l'andamento della pressione al variare dell'MFB50 permette di considerare tutte le coppie anticipo- λ attuabili.

2.2. Tipologie dati

L'acquisizione dei dati in una sala prove motori è un processo fondamentale per la valutazione delle prestazioni dei motori a combustione interna e per la creazione di modelli data-driven. La sala prove è equipaggiata con strumenti avanzati per monitorare una serie di parametri operativi del motore in tempo reale.

I dati sperimentali disponibili per questo lavoro di tesi possono essere suddivisi in due gruppi: registrazioni continue del motore e punti di funzionamento del motore in regime stazionario e mediati.

2.2.1. Data Train

I dati utilizzati per addestrare il modello provengono da registrazioni dei punti di funzionamento del motore in regime stazionario, relative a diversi parametri chiave mediati tra i cilindri 5, 6 e 7 (il cilindro 8 è stato escluso poiché presentava un comportamento leggermente differente).

I test stazionari, come il VVT sweep, il lambda sweep e lo spark sweep, vengono eseguiti su motori a combustione interna per analizzare il loro comportamento in condizioni controllate, al fine di ottimizzarne l'efficienza e le prestazioni. In ciascun sweep, un singolo parametro del motore viene variato sistematicamente, mantenendo costanti tutti gli altri, per comprendere meglio il suo impatto sulle prestazioni complessive del motore. Per ogni tipologia di dati si riporta una breve spiegazione sul funzionamento e il grafico che rappresenta i punti motore (Coppia-Giri) utilizzati durante l'acquisizione dei dati training:

- **VVT Sweep (Variable Valve Timing)**

Il VVT sweep si concentra sulla regolazione del timing delle valvole (apertura e chiusura) del motore, che può essere variabile nei moderni motori a combustione interna per ottimizzare le prestazioni in base alle condizioni di carico e regime. Durante il test, si esegue una variazione continua o a intervalli del timing delle valvole, mentre il motore è mantenuto a un regime e a un carico costanti.

- **Lambda sweep**

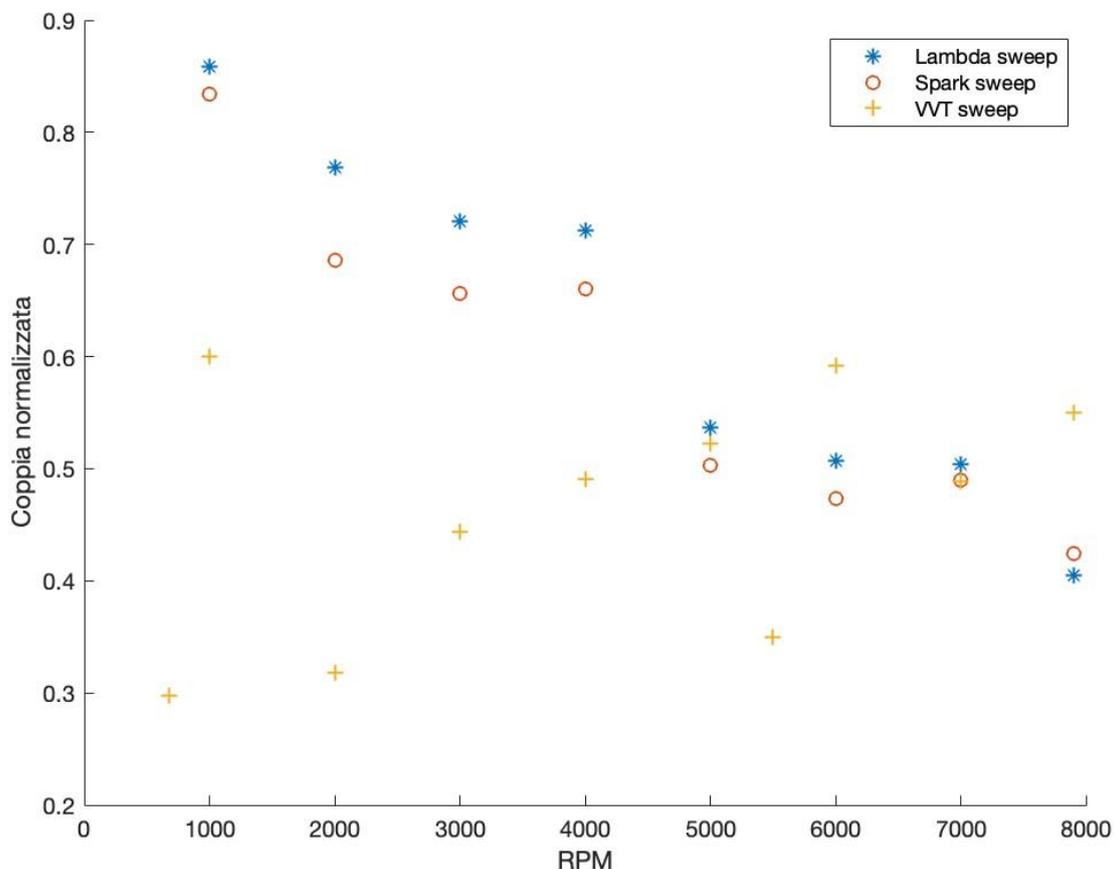
Il lambda sweep varia il rapporto aria-carburante (indicato come lambda) per esplorare il comportamento del motore sotto diverse condizioni di miscela. Il lambda è il rapporto tra l'aria effettiva aspirata e quella necessaria per una combustione stechiometrica ideale (lambda = 1 significa rapporto stechiometrico). Durante il test, si cambia gradualmente il rapporto aria-carburante (rendendolo più ricco o più magro) mantenendo costanti il carico e la velocità del motore. Le prove di variazione del lambda vengono utilizzate per calibrare la mappa di variazione di coppia in funzione di lambda e la mappa di variazione di anticipo ottimo in funzione di lambda.

Quello che ci si aspetta è che per valori di lambda di 0.9 l'angolo di anticipo ottimo sia minore rispetto a quello calibrato a lambda 1.1, perché per miscele ricche la velocità di combustione è maggiore e, per mantenere l'angolo MFB50 centrato attorno a valori di 8°/12° dopo il punto morto superiore, la combustione deve partire qualche istante dopo. Continuando ad ingrassare la miscela, la velocità di combustione viene rallentata nuovamente ma risulta comunque maggiore rispetto a quella di miscele magre. Il valore di anticipo deve essere invece aumentato per miscele magre, la cui velocità di combustione è minore rispetto a quella a lambda 1.1. La coppia erogata risulta invece massima per valori di miscela leggermente ricca, poiché a causa delle alte temperature dovute alla combustione che avviene in condizioni stechiometriche, sono presenti molecole di ossigeno dovute alla dissociazione dei prodotti di combustione e risulta quindi possibile bruciare più combustibile rispetto alla quantità stabilita dalla reazione chimica di combustione.

- **Spark sweep**

Il spark sweep riguarda la variazione dell'anticipo dell'accensione (SA, Spark Advance), cioè il momento in cui la candela genera la scintilla per avviare la combustione del carburante. Durante questo test, l'anticipo viene aumentato o diminuito mantenendo costanti gli altri parametri (carico e RPM), per verificare l'effetto di diverse impostazioni sul motore. La combustione presenta efficienza massima in corrispondenza di un preciso valore di angolo di accensione, situato prima del punto morto superiore e tale per cui l'angolo MFB50, che individua il punto in cui la metà del combustibile iniettato è stato bruciato, si trova tra gli 8 e i 12 gradi dopo il punto morto superiore.

In seguito, è riportato un grafico che mostra per ogni tipologia di dati training, i punti di funzionamento motore coppia – giri. In questo caso la coppia è normalizzata per il valore massimo.



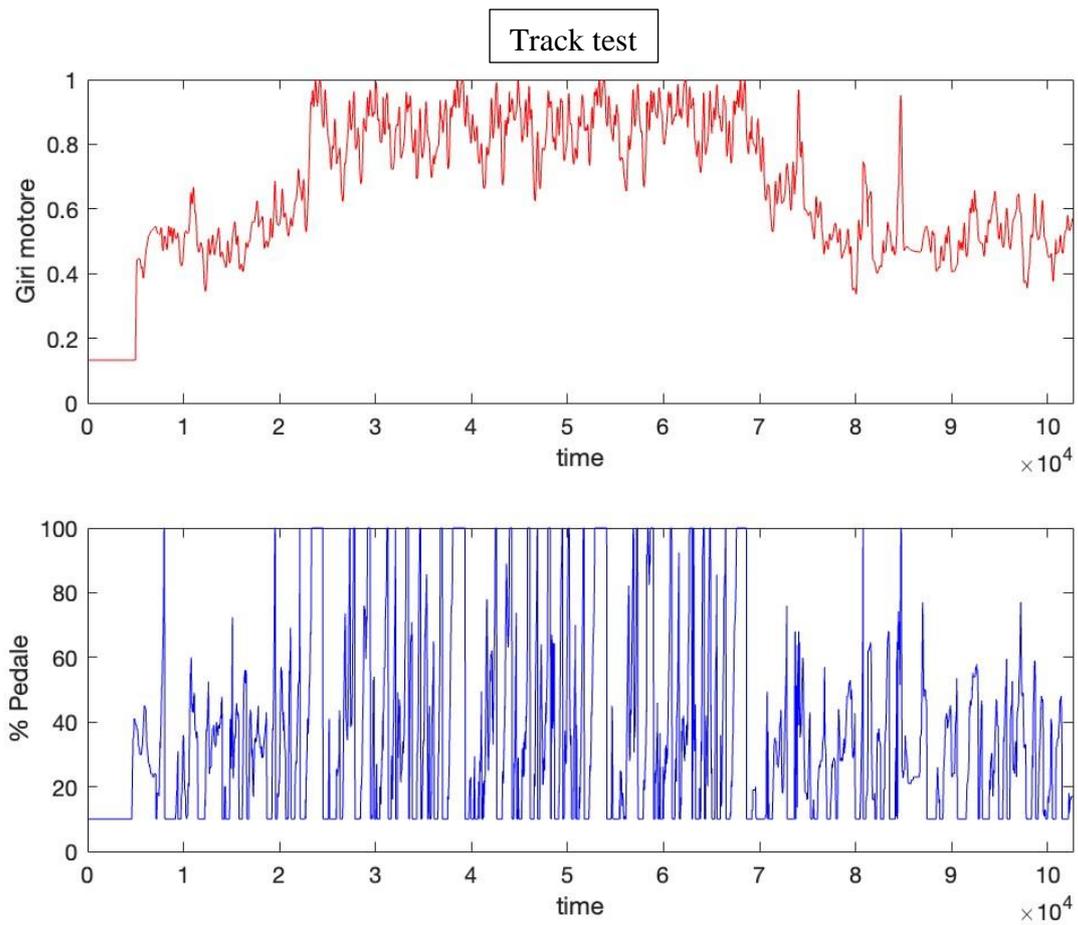
2.2.2. Data Test

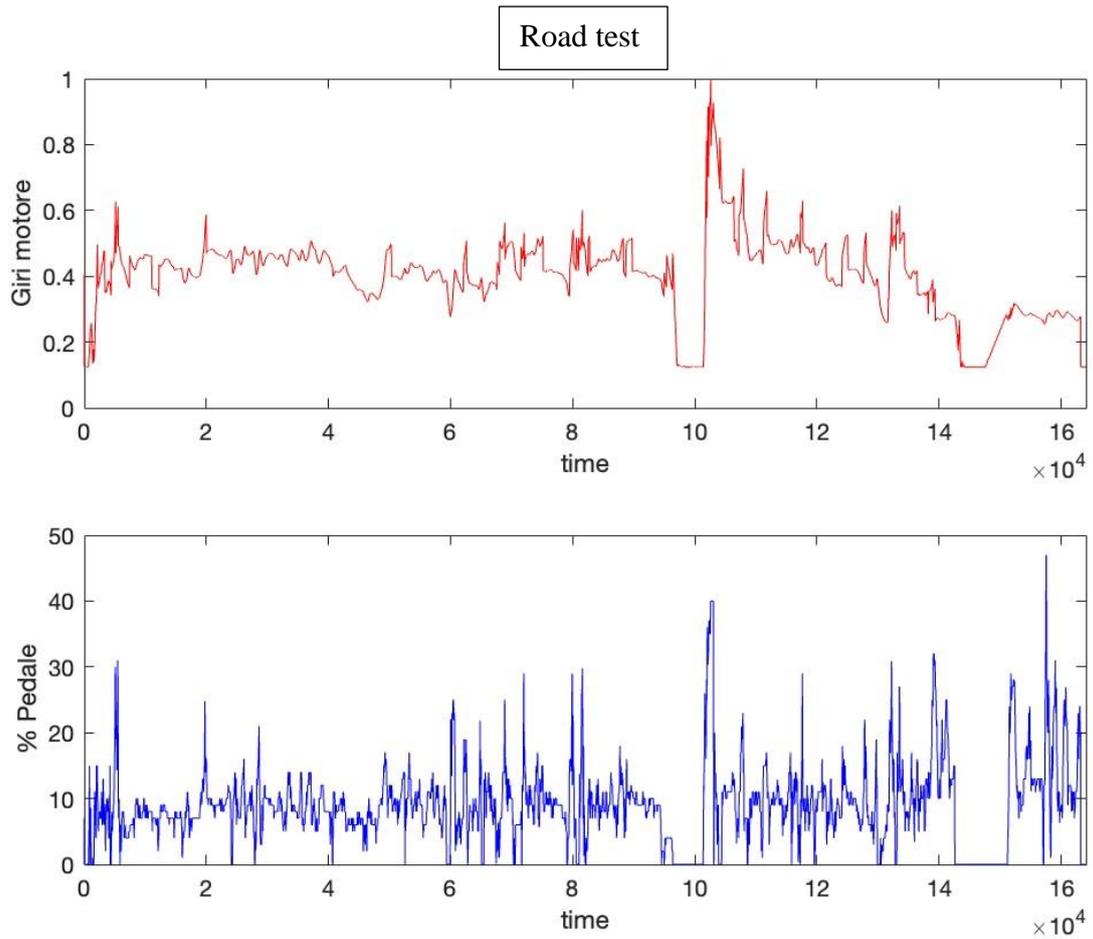
I dati impiegati nella fase di testing dei modelli di Ensemble Learning sono di natura continua e rappresentano nello specifico alcuni cicli di guida effettuati sul motore. Questi dati riflettono il comportamento dinamico del motore durante il suo funzionamento, includendo una serie di variabili che cambiano in modo continuo nel tempo. Tali variabili possono comprendere parametri come velocità del motore, coppia, consumo di carburante e altre grandezze legate alle prestazioni e all'efficienza del motore in condizioni di guida diverse. Utilizzando dati continui, si riesce a testare l'efficacia dei modelli nell'adattarsi a scenari complessi e nel prevedere con precisione il comportamento del motore in vari contesti operativi.

Durante la fase di testing sono stati usati due cicli guida, uno rispecchia un ciclo di guida quotidiano ("road test"), mentre il secondo un ciclo di guida più spinto eseguito in pista ("track test") allo scopo di avere un modello più efficiente in tutte le condizioni.

I grafici sottostanti mostrano la relazione tra i giri motore e la posizione del pedale dell'acceleratore durante una sessione di guida.

Quando il pedale dell'acceleratore viene premuto (grafico inferiore del subplot), ci sono variazioni nei giri motore (grafico superiore del subplot), anche se non sempre direttamente proporzionali, probabilmente a causa dell'inerzia del motore o di altri fattori dinamici.





Questo tipo di grafico è tipico di prove RDE (Real Driving Emissions), in cui si monitorano vari parametri durante la guida reale per valutare le prestazioni e le emissioni del motore.

Capitolo 3

3. MFB50

La frazione di massa bruciata (MFB) in ogni singolo ciclo del motore è una quantità normalizzata con una scala da 0 a 1, che descrive il processo di rilascio di energia chimica in funzione dell'angolo di rotazione dell'albero motore.

MFB50 (Mass Fraction Burned 50%) è un parametro di combustione che indica l'angolo di rotazione dell'albero motore al quale il 50% della massa totale del combustibile è bruciato in un cilindro del motore. È fondamentale per comprendere e ottimizzare il processo di combustione, fornendo informazioni sul tempismo e l'efficienza della combustione del carburante. L'analisi dell'MFB50 aiuta gli ingegneri a regolare il timing di accensione, migliorare l'efficienza del carburante e ridurre le emissioni.

Come si vede in Figura 3-1 al diminuire dell'MFB50 aumenta il lavoro prodotto dal ciclo qui identificato attraverso l'IMEP, definita come il lavoro del ciclo termodinamico indicato per unità di cilindrata.

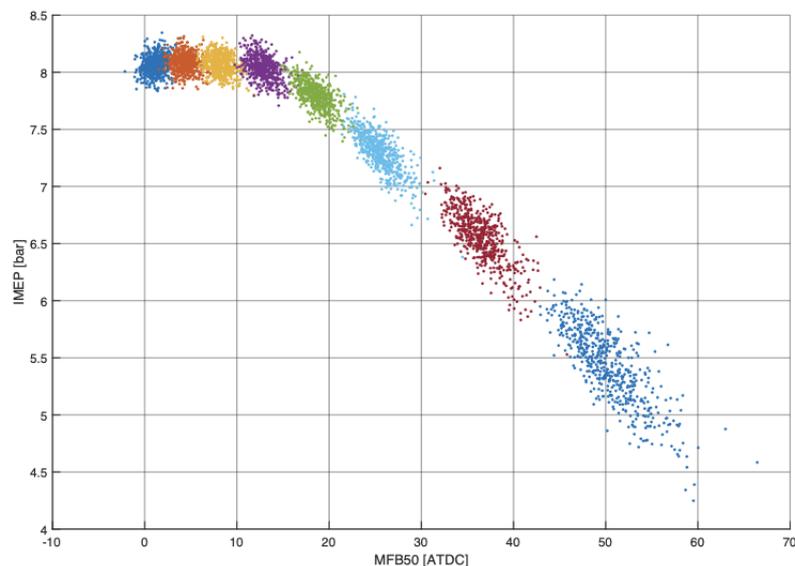


Figura 3-1: Andamento della pressione media indicata, **IMEP**, al variare dell'**MFB50**

Il fenomeno mostrato in Figura 3-1 è dovuto all'aumento di efficienza prodotto da una fase di combustione più spostata verso la compressione. Infatti, in tali condizioni viene liberata più energia a parità di combustibile iniettato nel cilindro. Il punto ottimo di efficienza si ha per un valore di MFB50 compreso tra 8 e 10 °ATDC (gradi dopo il punto morto di fine compressione).

Al diminuire di MFB50, la combustione si sposta verso la compressione. In questo modo la miscela inizia a liberare calore, quindi ad aumentare la propria temperatura e la propria pressione, mentre il pistone sta ancora eseguendo la corsa di compressione. Il verificarsi di questo fenomeno è ottimo dal punto di vista della reazione, perché aiuta la miscela a raggiungere le condizioni di accendibilità. Infatti, la presenza di combustibile e comburente all'interno di un volume predefinito non è condizione sufficiente affinché avvenga la combustione. Essa richiede che i due reagenti stiano tra di loro a rapporto stechiometrico, siano ben miscelati e che si trovino a una temperatura superiore alla temperatura di attivazione della reazione. Una volta prodotta la scintilla, la miscela immediatamente adiacente alla candela supera la temperatura di attivazione, dando il via alla reazione di combustione. Dalla zona adiacente alla candela si libera, in tal modo, un fronte di fiamma che propaga la reazione in tutto il volume in cui è presente la miscela. Se durante la propagazione del fronte di fiamma il pistone continuasse a comprimere i reagenti, essi aumenterebbero la propria temperatura e la combustione risulterebbe più omogenea in tutto il volume, provocando un aumento dell'energia prodotta. Al contrario se il fronte di fiamma iniziasse la propria propagazione durante la fase di espansione, esso incontrerebbe gas sempre più freddi che farebbero sempre più fatica a partecipare alla reazione. Nel primo caso però aumenta il lavoro di compressione che, per anticipi di accensione molto elevati, incide sensibilmente sul lavoro prodotto. La Figura 3-2 mostra chiaramente l'effetto dell'anticipo di accensione sulla pressione in camera.

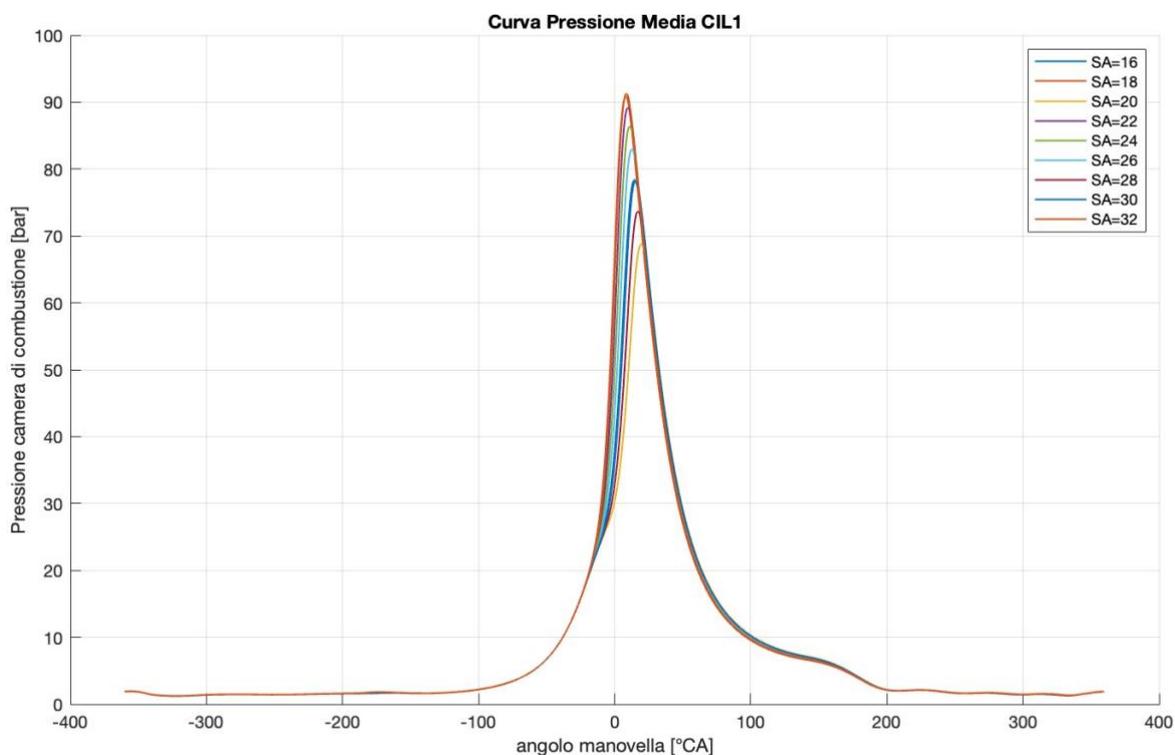


Figura 3-2: Segnali di **pressione** in camera al variare dell'**SA**

Il grafico mostra come cambia l'andamento della pressione in camera al variare di MFB50.

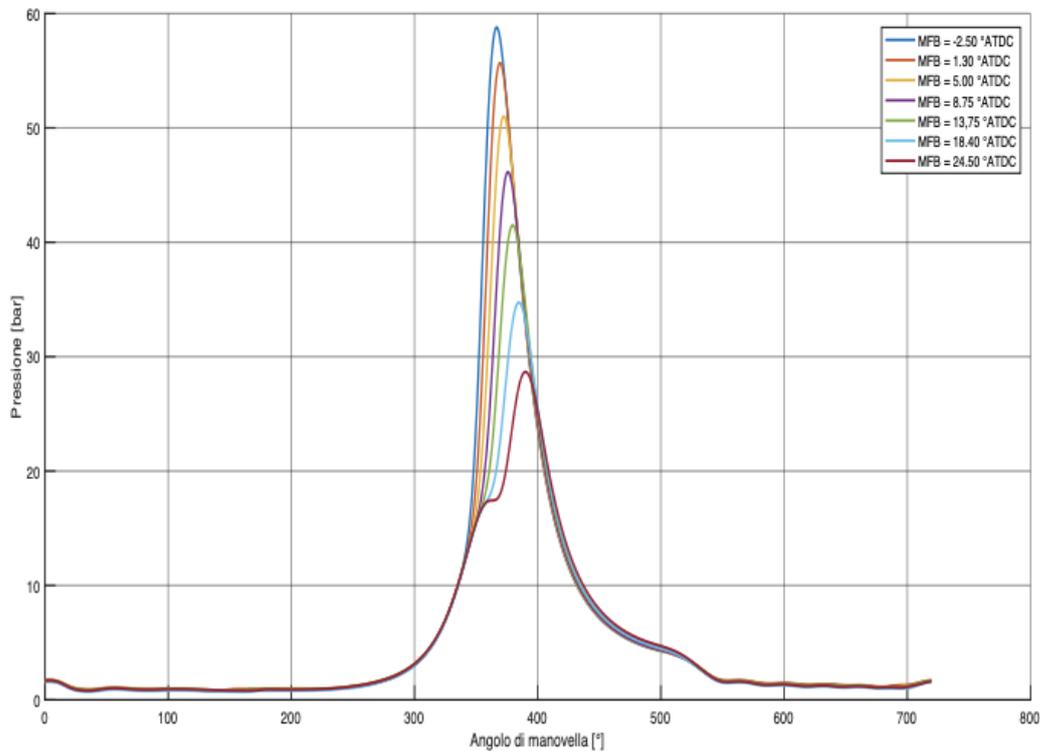


Figura 3-3: Segnali di pressione in camera al variare di **MFB50**

Per MFB50 bassi (nel caso in figura il valore minimo è pari a -2.50° ATDC) il segnale di pressione cambia repentinamente pendenza nell'istante angolare in cui viene prodotta la scintilla. In questo modo si raggiungono valori di pressione molto elevati prima che la miscela raggiunga un livello di espansione tale da invertire l'andamento della pressione stessa. Al contrario si vede che una combustione ritardata produce un gradiente di pressione inferiore che non permette di generare pressioni importanti. Man mano che la combustione viene ritardata si riesce a distinguere sempre più chiaramente il picco di pressione dovuto alla compressione da quello dovuto alla combustione. Fino a raggiungere lo stato in cui la combustione è talmente spostata verso l'espansione che la miscela non riesce a raggiungere le condizioni di accendibilità neanche grazie all'ausilio della scintilla. Un altro aspetto interessante che si può evincere dalla Figura 3-3 riguarda il valore per cui si ha il valore massimo di P_{MAX} sviluppata in camera. Come detto in precedenza il massimo lavoro si ha per MFB50 pari a 8° ATDC, questo potrebbe indurre a dire che, essendo il lavoro pari al prodotto tra la pressione e la variazione di volume, il punto di massimo del lavoro coincida con il punto di massimo della P_{MAX}. Questo, invece, non è vero. La pressione massima in camera continua ad aumentare anche per MFB50 minori di 8° ATDC. Questo perché la sovrapposizione degli effetti di combustione e compressione produce un aumento di P_{MAX} ma comporta un lavoro di compressione maggiore. La Figura 3-4 mostra il fenomeno appena descritto. Come si vede dalla curva in figura, la P_{MAX} aumenta fino a raggiungere un massimo. Dopo il massimo la curva inizia a decrescere perché la

combustione risulta talmente anticipata da non permettere al pistone di compiere una compressione completa. Invece per valori di MFB50 molto elevati la miscela non riesce ad accendersi e la pressione massima rimane quella sviluppata a fine compressione.

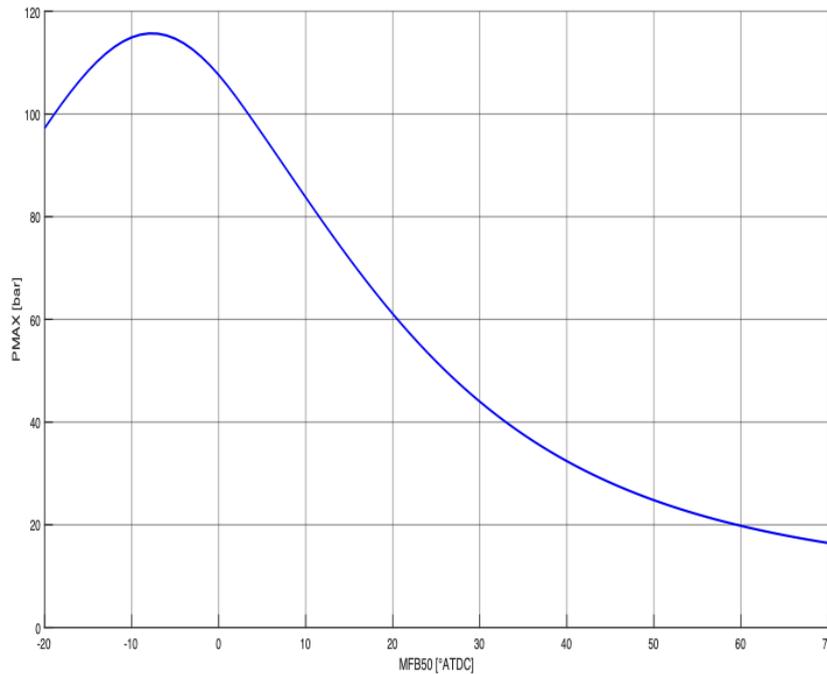


Figura 3-4: Andamento P_{MAX} al variare di $MFB50$

In fase di analisi dei dati sperimentali acquisiti al banco, MFB50 viene calcolato partendo dalla definizione di calore lordo sviluppato in camera:

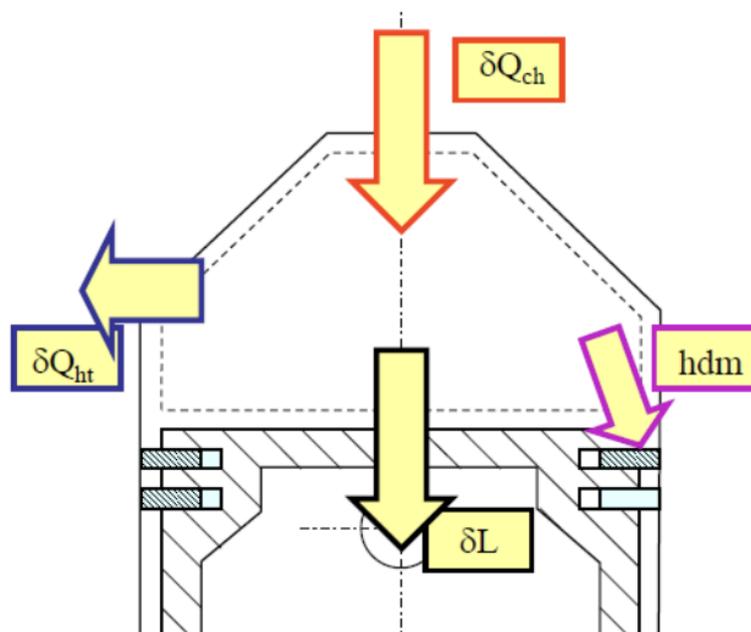


Figura 3-5: Modello di rilascio di calore in camera di combustione

$$\delta Q_{ch} = dU_s + \delta Q_{ht} + dL + h dm \quad (1.1)$$

Dove:

- dQ_{ch} = calore lordo liberato durante la combustione
- dU_s = variazione di energia interna del sistema aperto considerato
- δQ_{ht} = calore scambiato con le pareti
- dL = lavoro compiuto dai gas sul pistone
- $h dm$ = entalpia associata alle perdite di gas attraverso le fasce elastiche

Nell'ipotesi che il gas sia perfetto e che la temperatura sia uniforme in tutto il volume l'energia interna risulta pari a:

$$U_s = m * u(T) \quad (1.2)$$

il cui differenziale diventa:

$$dU_s = m * c_v * dT + u * dm \quad (1.3)$$

Sostituendo l'espressione (1.3) nella (1.4):

$$\delta Q_{ch} = m * c_v * dT + \delta Q_{ht} + dL + (h - u)dm \quad (1.4)$$

Il rilascio netto di calore viene definito dall'espressione:

$$\delta Q_n = \delta Q_{ch} - \delta Q_{ht} - (h - u)dm \quad (1.5)$$

Sostituendo la (1.4) all'interno della (1.5) si arriva all'espressione:

$$\delta Q_n = m * c_v * dT + dL \quad (1.6)$$

Ipotizzando di essere nelle condizioni di gas perfetto vale la relazione:

$$dT = \frac{1}{mR} (p dV + V dp) \quad (1.7)$$

Sostituendo nella (1.6) la (1.7) insieme all'espressione del lavoro $dL = p dV$, il calore netto diventa:

$$\delta Q_n = \left(\frac{c_v}{R}\right) V dp + \left(\frac{c_v}{R} + 1\right) p dp \quad (1.8)$$

Introducendo nell'espressione (1.8) la relazione di Mayer, $R = c_p - c_v$, e il coefficiente $\gamma = \frac{c_p}{c_v}$ si arriva all'espressione del **ROHR – rate of heat release**:

$$ROHR = \frac{\delta Q_n}{\delta \theta} = \frac{1}{\gamma-1} V \frac{dp}{d\theta} + \frac{\gamma}{\gamma-1} p \frac{dV}{d\theta} \quad (1.9)$$

Analizzando l'espressione (1.9) si evince che noto l'andamento della pressione e del volume durante il ciclo è possibile calcolare il calore netto sviluppato per unità di angolo.

Una volta calcolato il ROHR, per dato ciclo di pressione, è possibile calcolare il calore netto semplicemente integrando l'espressione (1.9). Infine, il calore netto così calcolato viene normalizzato rispetto al proprio massimo, il risultato che si ottiene è la frazione di massa di combustibile bruciata al variare dell'angolo di manovella, MFB. Da quest'ultima informazione si trova l'MFB50 come l'angolo in cui la funzione MFB è pari a 0.5.

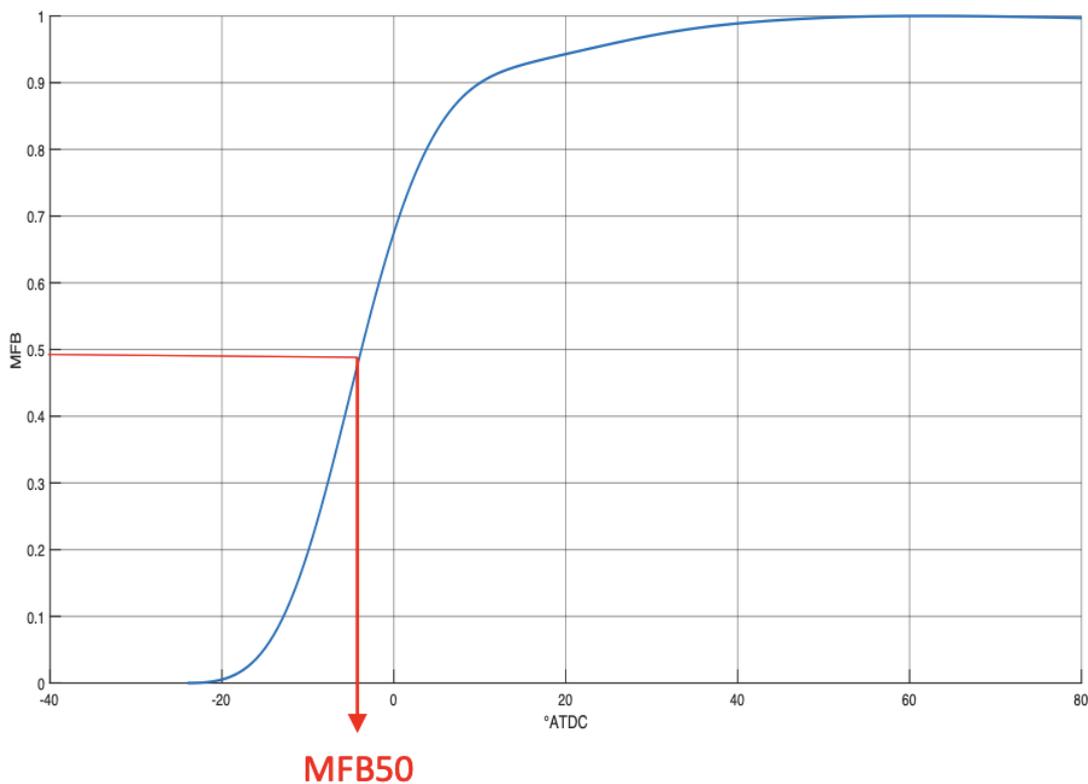


Figura 3-6: Andamento **MFB** al variare dell'angolo di manovella

Capitolo 4

4. Script Matlab

Lo sviluppo dello script Matlab dei modelli è stato diviso in quattro macro-fasi: acquisizione dati di training, definizione e training dei modelli di ensemble learning con ottimizzazione del numero di alberi, acquisizione dei dati di testing e fase di testing sui cicli di guida con i modelli ottimizzati.

- Dati di training

Nella parte iniziale dello script si selezionano i file relativi alle prove in stazionario e si acquisiscono tutti i dati utili al training (RPM, rl, SA, MFB50); si uniscono infine in un'unica variabile, detta "Alldata".

```
%% Acquisizione dati da fogli excell

% LAMBDA SWEEP DATA

[fName_banco, folder_banco] = uigetfile('*.xlsx', "MultiSelect", "on", "Select bench files");
raw = readtable([folder_banco fName_banco]);
data = raw(3:end,:);

for i = 1:92
    x = data(i,194:196);
    MFB50(i,1) = mean(x,2);
end

% SPARK SWEEP DATA

[fName_banco2, folder_banco2] = uigetfile('*.xlsx', "MultiSelect", "on", "Select bench files");
raw2 = readtable([folder_banco2 fName_banco2]);
data2 = raw2(3:end,:);

for i = 1:128
    x2 = data2(i,194:196);
    MFB50(92+i,1) = mean(x2,2);
end

% VVT SWEEP DATA

[fName_banco3, folder_banco3] = uigetfile('*.xlsx', "MultiSelect", "on", "Select bench files");
raw3 = readtable([folder_banco3 fName_banco3]);
data3 = raw3(3:end,:);

for i = 1:350
    x3 = data3(i,194:196);
    MFB50(220+i,1) = mean(x3,2);
end

% ALL DATA TABLE (TRAIN SET)

RPM = [data(:,69);data2(:,69);data3(:,69)];
rl = [data(:,124);data2(:,124);data3(:,124)];
SA = [data(:,157);data2(:,157);data3(:,157)];
lambda = [data(:,99);data2(:,99);data3(:,99)];

Alldata = [RPM,rl,SA,lambda,MFB50];
Alldata.Properties.VariableNames(5) = {'MFB50'};
```

- Definizione e training modelli

In questa fase si implementa un ciclo for, variando a ogni iterazione il numero di alberi decisionali, calcolando con il modello l'errore quadratico medio tra il valore reale di MFB50 e quello predetto. Successivamente si realizza un grafico per identificare più chiaramente il numero di alberi che dia delle predizioni abbastanza corrette evitando l'overfit dello stesso sui dati di training.

Infine, viene realizzato un grafico che visualizza la dispersione dei dati predetti rispetto a quelli reali.

Nello script sottostante è stato sviluppato il modello di regressione mediante la tecnica di Boosting (in particolare LSBoost) mediante la funzione Matlab apposita chiamata “fitrensemble” [7].

```
%% LSBoost - fitrensemble

numTrees = 1:1:100;
for i = 1:length(numTrees)
    t = templateTree('Surrogate','on','MaxNumSplits',1);
    Mdl_1 = fitrensemble(Alldata,'MFB50','NumLearningCycles',numTrees(i),...
        'Learners',t,'LearnRate',0.1);
    pmdl = predict(Mdl_1,Alldata);
    RMSE_1(i) = rmse(pmdl,Alldata.MFB50);
end

f = fit(numTrees',RMSE_1','exp2'); % grafico su cui fare la scelta del numero di alberi
plot(f,numTrees,RMSE_1);
xlabel('Number of Grown Trees');
ylabel('Root Mean Squared Error');
title("Boosting");

j1=55; % scelgo il numero di alberi minimo che mi consente di ottenere un buon RMSE evitando overfitting
Mdl_lopt = fitrensemble(Alldata,'MFB50','NumLearningCycles',numTrees(j1),'Learners',t,'LearnRate',0.1);
RMSE_opt1 = feval(f,numTrees(j1));

figure;

line(Alldata.MFB50,pmdl,'DisplayName','Observations','MarkerFaceColor',...
    [0 0.447 0.741],'MarkerSize',8,'Marker','.', 'LineStyle','none','Color',...
    [0 0.447 0.741]);

line(Alldata.MFB50,Alldata.MFB50,'DisplayName','Perfect prediction','LineWidth', 1.3);

ylabel('PREDICTED MFB50');
xlabel('MFB50 MEASURED');
title("Boosting");
```

Il corrispettivo modello con la tecnica di Bagging è stato creato mediante la funzione Matlab “TreeBagger” [8].

```
%% BAGGING - TREEBAGGER

numTrees = 1:1:100;
for i = 1:length(numTrees)
    c = TreeBagger(numTrees(i),Alldata,'MFB50','Method','Regression','MinLeafSize',1);
    pc = predict(c,Alldata);
    RMSE_2(i) = rmse(pc,Alldata.MFB50);
end

f = fit(numTrees',RMSE_2','exp2');
plot(f,numTrees,RMSE_2);
xlabel('Number of Grown Trees');
ylabel('Root Mean Squared Error');
title("Bagging");

j2 = 10;
c_opt = TreeBagger(numTrees(j2),Alldata,'MFB50','Method','Regression','MinLeafSize',1);
RMSE_opt2 = feval(f,numTrees(j2));

figure;

line(Alldata.MFB50,pc,'DisplayName','Observations','MarkerFaceColor',[0 0.447 0.741], 'MarkerSize',8, 'Marker', '.', ...
    'LineStyle','none','Color',[0 0.447 0.741]);

line(Alldata.MFB50,Alldata.MFB50,'DisplayName','Perfect prediction','LineWidth',1.3);

ylabel('PREDICTED MFB50');
xlabel('MFB50 MEASURED');
title("Bagging");
```

Nel modello di Random Forest viene usata la “[Bayesian optimization](#)” [9]. L’ottimizzazione Bayesiana è una strategia di progettazione sequenziale per l’ottimizzazione globale di funzioni black-box che non assume alcuna forma funzionale.

```
%% RANDOM FOREST using quantile error and Bayesian Optimization

maxMinLS = 20;
minLS = optimizableVariable('minLS',[1,maxMinLS],'Type','integer');
numPTS = optimizableVariable('numPTS',[1,size(Alldata,2)-1],'Type','integer');
hyperparametersRF = [minLS; numPTS];
results = bayesopt(@(params)oobErrRF(params,Alldata),hyperparametersRF,'AcquisitionFunctionName','expected-improvement-plus','Verbose',0);
best00BErr = results.MinObjective;
bestHyperparameters = results.XAtMinObjective;

numTrees = 1:1:100;
for i = 1:length(numTrees)
    RFmdl = TreeBagger(numTrees(i),Alldata,'MFB50','Method','Regression','MinLeafSize',bestHyperparameters.minLS,'NumPredictorstoSample', ...
        bestHyperparameters.numPTS);
    pRFmdl = predict(RFmdl,Alldata);
    RMSE_3(i) = rmse(pRFmdl,Alldata.MFB50);
end

f = fit(numTrees,'RMSE_3','exp2');
plot(f,numTrees,RMSE_3);
xlabel('Number of Grown Trees');
ylabel('Root Mean Squared Error');
title("Random forest");

j3=20;
RFmdl_opt = TreeBagger(numTrees(j3),Alldata,'MFB50','Method','Regression','MinLeafSize',bestHyperparameters.minLS,'NumPredictorstoSample', ...
    bestHyperparameters.numPTS);
RMSE_opt3 = feval(f,numTrees(j3));

figure;

line(Alldata.MFB50,pRFmdl,'DisplayName','Observations','MarkerFaceColor',[0 0.447 0.741],'MarkerSize',8,'Marker','.','LineStyle','none','Color', ...
    [0 0.447 0.741]);

line(Alldata.MFB50,Alldata.MFB50,'DisplayName','Perfect prediction','LineWidth',1.3);

ylabel('PREDICTED MFB50');
xlabel('MFB50 MEASURED');
title("Random forest");
```

- Dati di testing

L’acquisizione dei dati di testing è analoga a quella usata per i dati di training.

```
%% TEST SET TABLE TRACK or ROAD

[fName_banco4, folder_banco4] = uigetfile('*.*xlsx','MultiSelect',"on","Select bench files");
raw4 = readtable([folder_banco4 fName_banco4]);
data4 = raw4(3:end,:);

if strcmp(fName_banco4, 'TRACK_2016_04_06_02_02_13474_000000_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00.xlsx') == 1
    for i = 1:size(data4)
        x4 = data4(i,193:195);
        MFB50t(i,1) = mean(x4,2);
    end
    RPMt = [data4(:,67)];
    rlt = [data4(:,123)];
    SAT = [data4(:,156)];
    lambdat = [data4(:,98)];
else
    for i = 1:164047
        x4 = data4(i,193:195);
        MFB50t(i,1) = mean(x4,2);
    end
    RPMt = [data4(:,67)];
    rlt = [data4(:,122)];
    SAT = [data4(:,155)];
    lambdat = [data4(:,97)];
end

dataTest = [RPMt,rlt,SAT,lambdat,MFB50t];
dataTest.Properties.VariableNames(5) = {'MFB50'};
```

- Fase di Testing

In quest'ultima fase vengono testati i tre modelli con i dati dei cicli guida, con i calcoli del rispettivo errore quadratico medio.

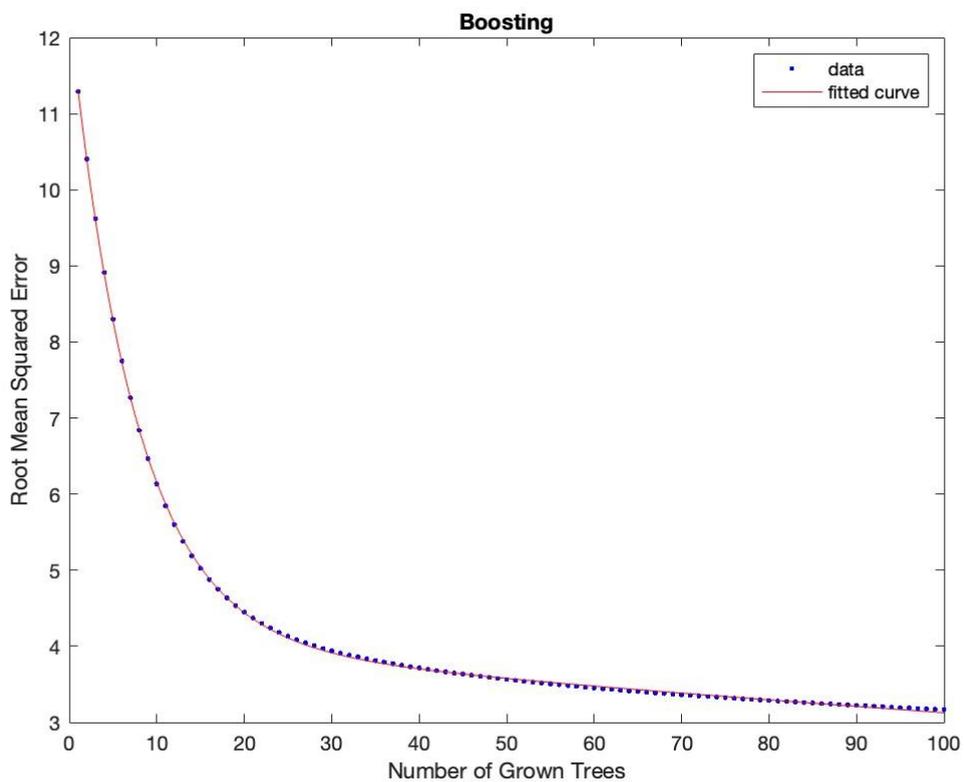
<pre>%% RMSE WITH TEST DATA</pre>	
<pre>%% model 1 (BOOSTING) t = templateTree('Surrogate','on','MaxNumSplits',1); pmdl_opt = predict(Mdl_1opt,dataTest); RMSE_1opt = rmse(pmdl_opt,dataTest.MFB50); time = 1:1:size(data4); figure; hold on plot(time,pmdl_opt,'-r','DisplayName','MFB50 predicted'); plot(time,dataTest.MFB50,'-b','DisplayName','MFB50 measured'); xlabel('time'); ylabel('MFB50 [CA°]'); title("Boosting"); hold off</pre>	
<pre>%% model 2 (BAGGING) pc_opt= predict(c_opt,dataTest); RMSE_copt=rmse(pc_opt,dataTest.MFB50); time = 1:1:size(data4); figure; hold on plot(time,pc_opt,'-r','DisplayName','MFB50 predicted'); plot(time,dataTest.MFB50,'-b','DisplayName','MFB50 measured'); xlabel('time'); ylabel('MFB50 [CA°]'); title("Bagging"); hold off</pre>	
<pre>%% model 3 (RF) pRFmdl_opt = predict(RFmdl_opt,dataTest); RMSE_3opt = rmse(pRFmdl_opt,dataTest.MFB50); time = 1:1:size(data4); figure; hold on plot(time,pRFmdl_opt,'-r','DisplayName','MFB50 predicted'); plot(time,dataTest.MFB50,'-b','DisplayName','MFB50 measured'); xlabel('time'); ylabel('MFB50 [CA°]'); title("Random forest"); hold off</pre>	

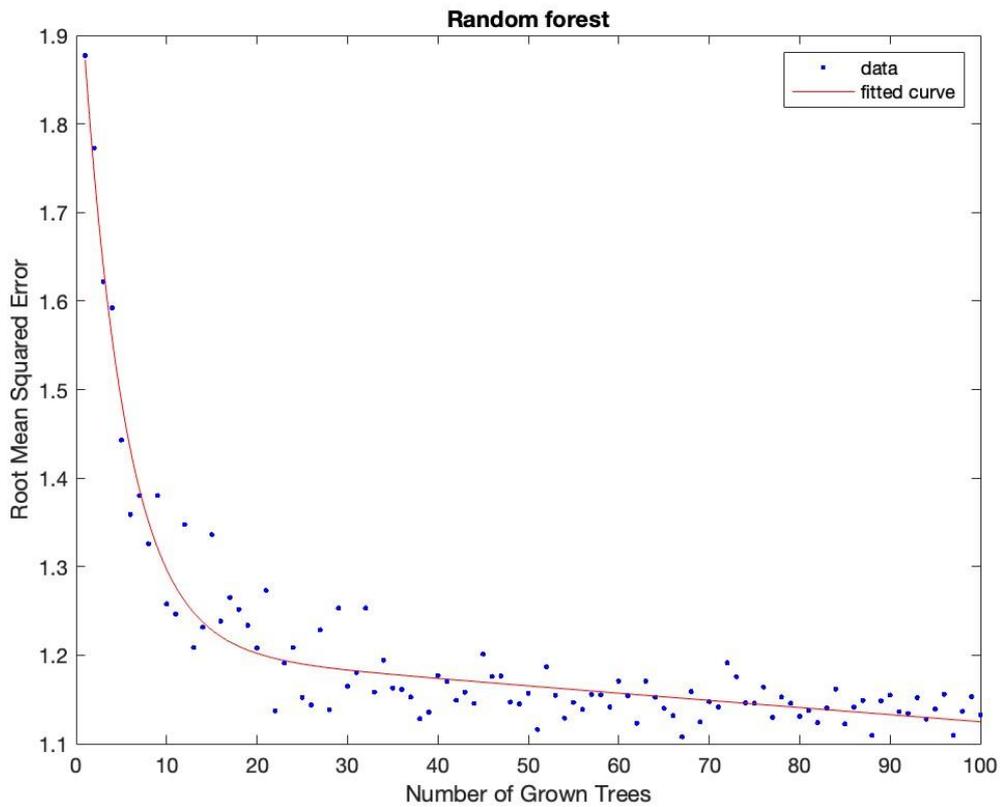
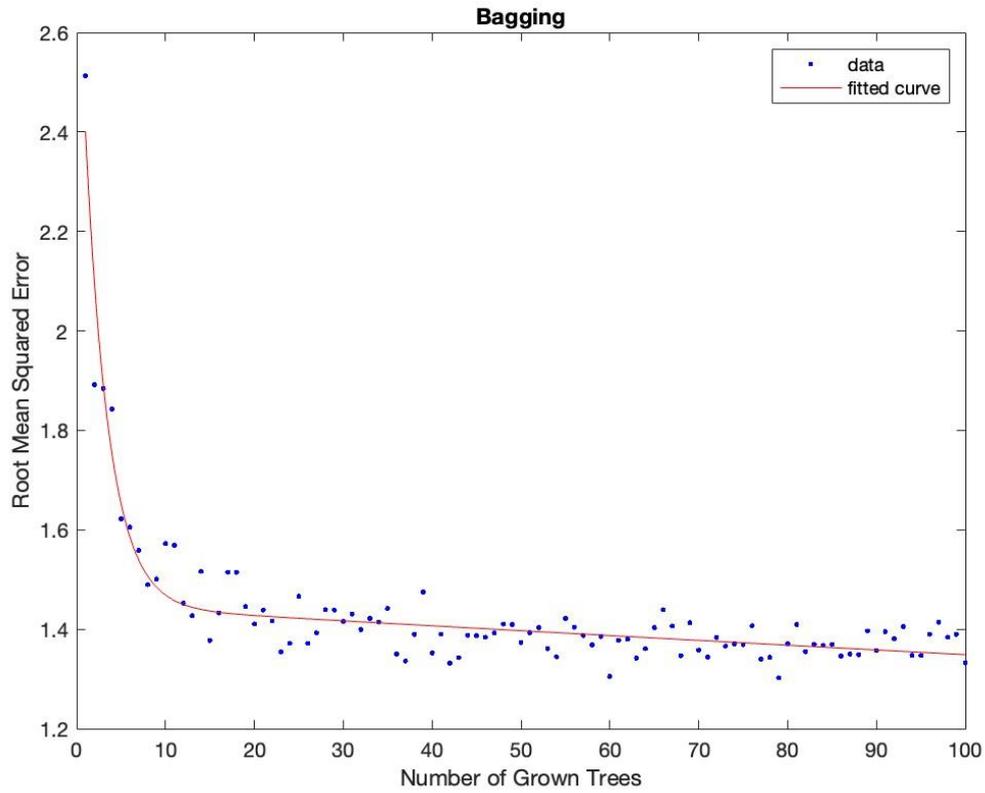
Capitolo 5

5. Risultati

5.1. Training

Per ottenere modelli il più possibile precisi e robusti, è stata effettuata un'ottimizzazione dei dati di training incentrata sull'iperparametro principale, ossia il numero di alberi decisionali per ogni modello. Sono stati creati tre tipi di modelli di ensemble: uno basato sul Boosting e due basati sul Bagging, uno in particolare sul Random Forest. L'ottimizzazione è stata condotta variando il numero di alberi decisionali a ogni iterazione e calcolando l'errore quadratico medio (RMSE) tra il valore predetto dal modello e il valore reale.

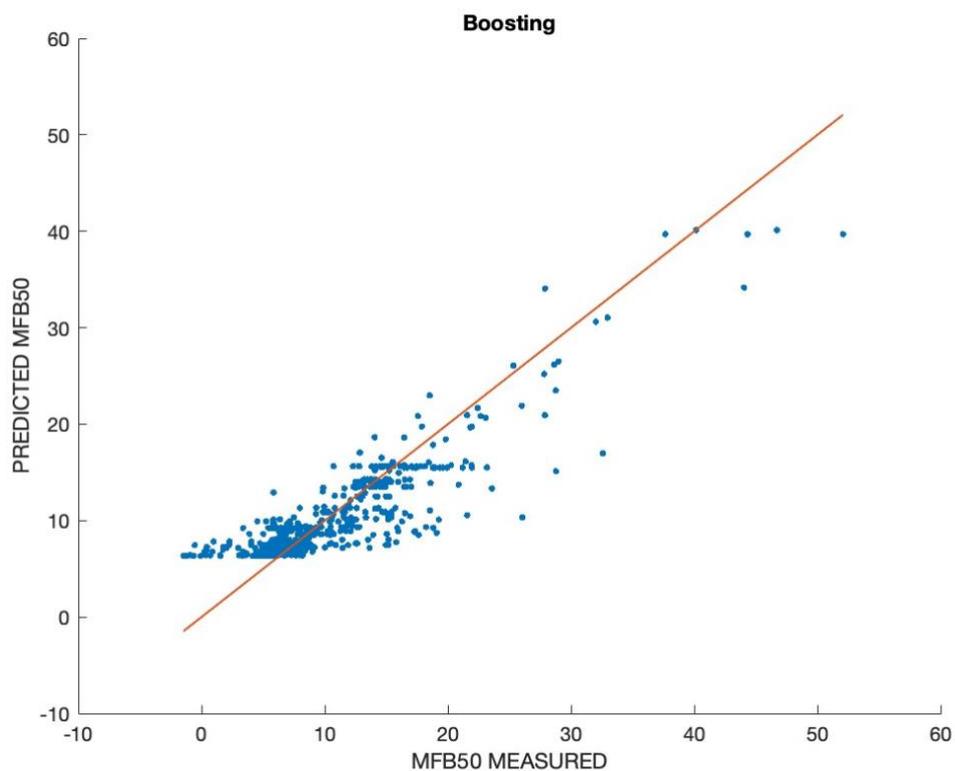


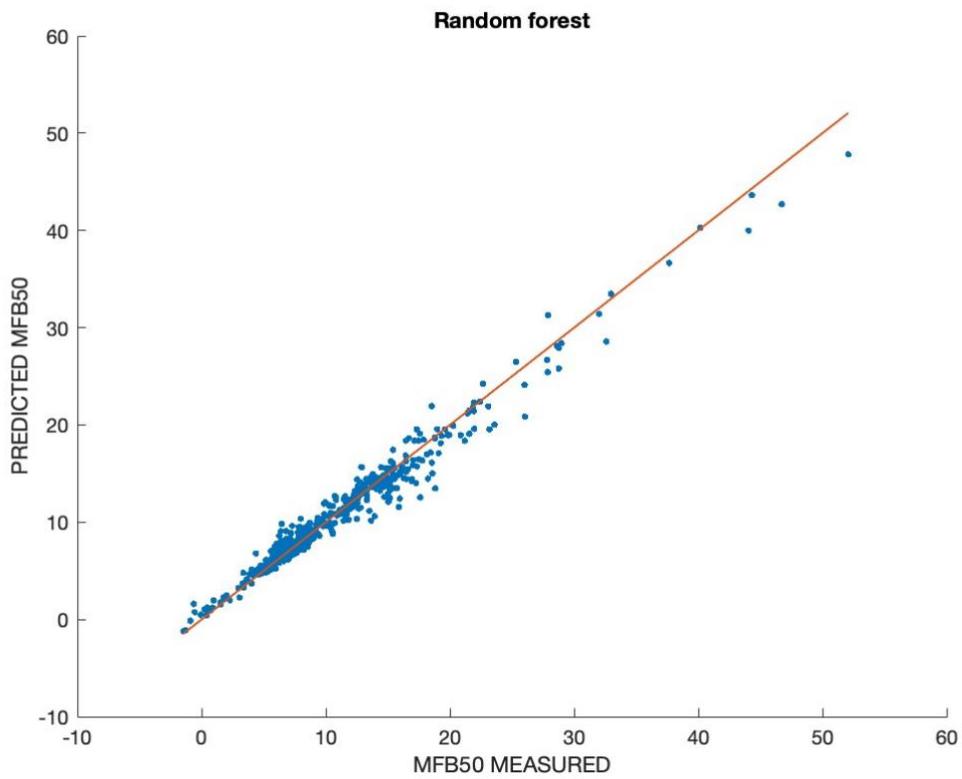
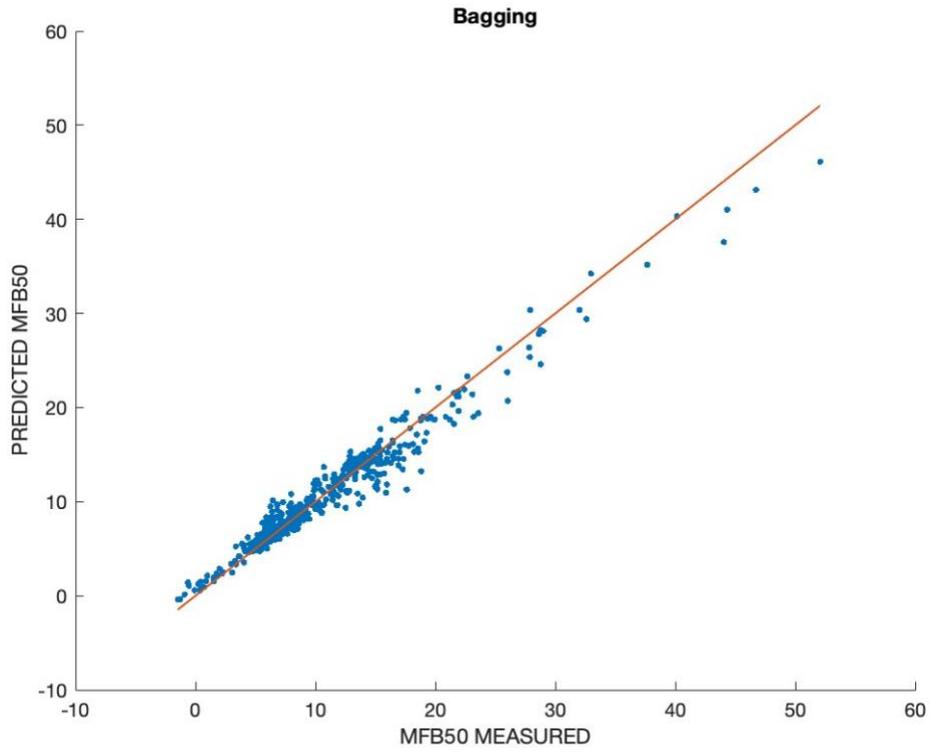


L'obiettivo era evitare l'overfitting, che avrebbe compromesso la generalizzazione del modello sui dati di test. Per questo motivo, si è scelto di adottare un numero di alberi decisionali sufficiente a garantire un RMSE basso senza eccedere, preservando l'equilibrio tra complessità

e accuratezza. Per il modello basato sul Boosting (LSBoost), il numero ottimale di alberi decisionali si è attestato a 55. Per i modelli di Bagging è stato scelto un numero di alberi pari a 10.

Dopo aver selezionato i parametri ottimali, i modelli sono stati allenati sul set di dati di training. Di seguito sono riportati i grafici che mostrano la previsione del valore di MFB50 rispetto ai valori reali. Come si può notare, i modelli basati su Bagging offrono una previsione più accurata rispetto al Boosting, con una minore varianza nei valori predetti.

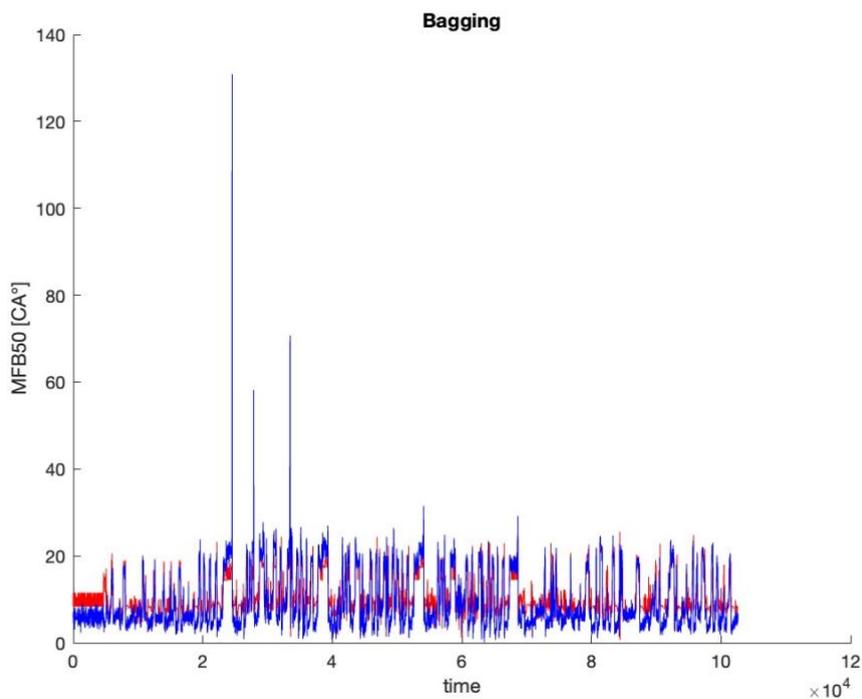
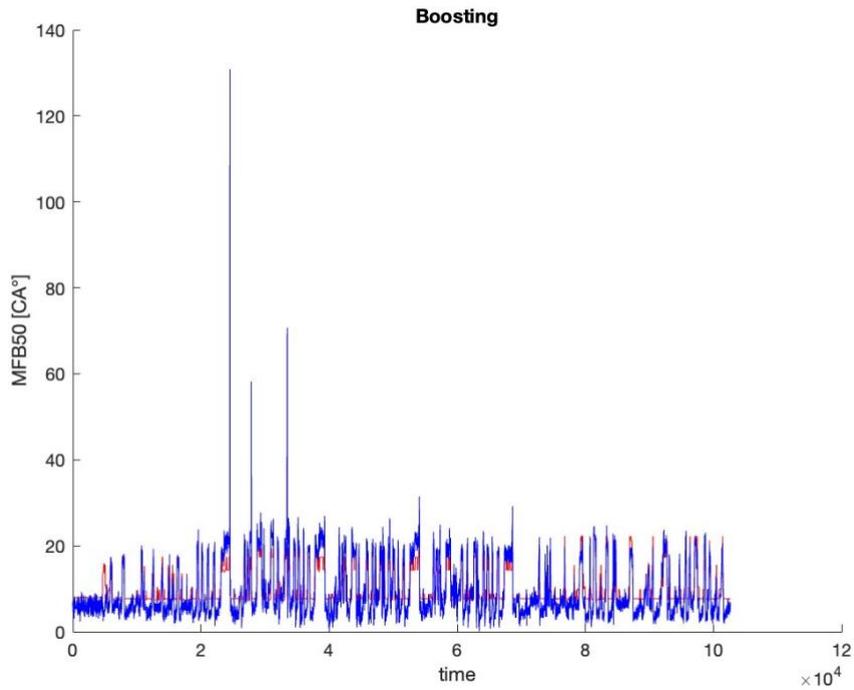


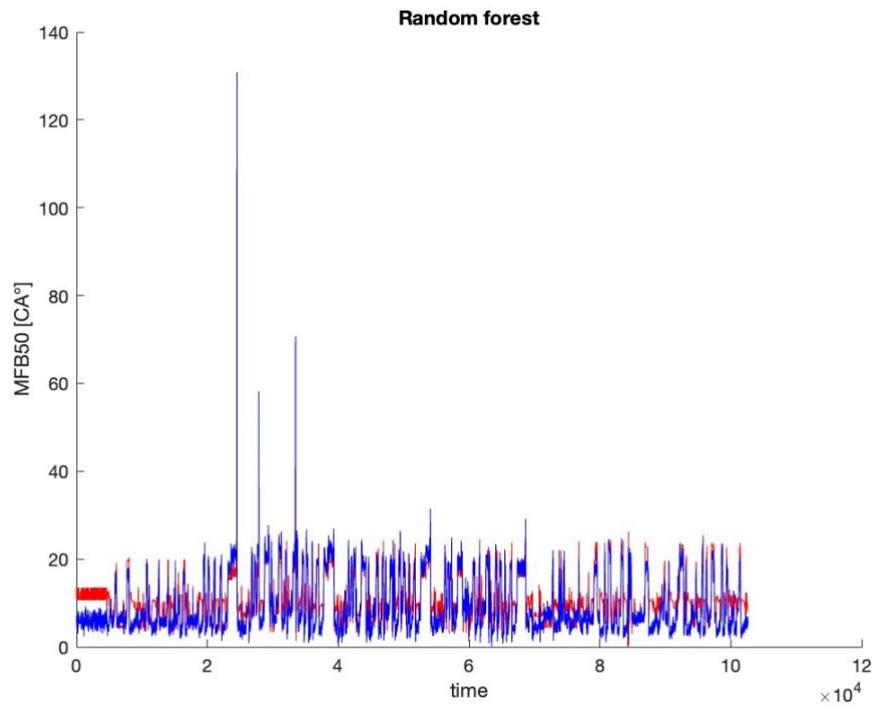


5.2. Testing

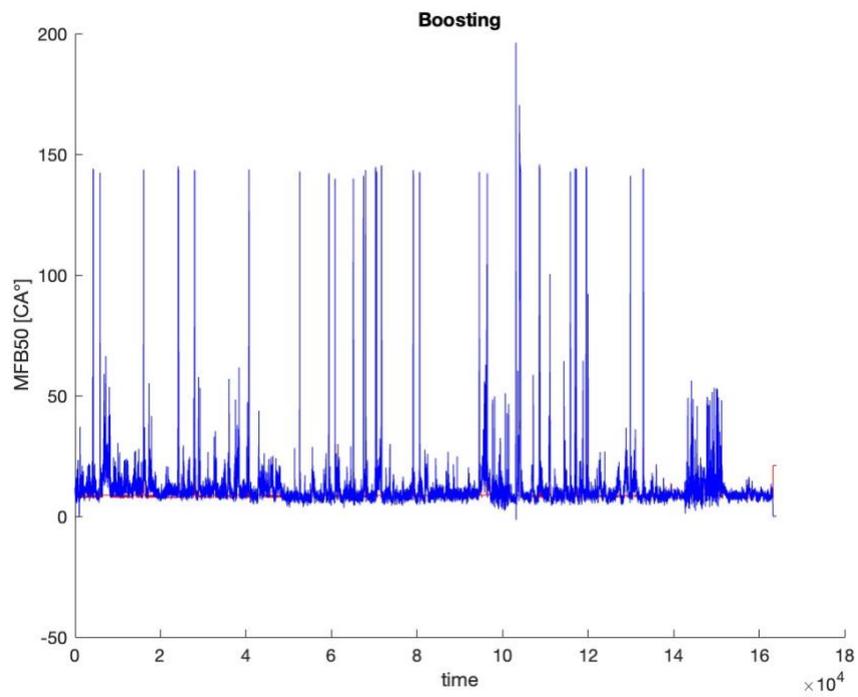
Durante la fase di testing, i modelli ottimizzati sono stati applicati a dati input relativi a cicli di guida reali. Sono stati utilizzati due cicli di guida differenti: il ciclo in strada e il ciclo in pista. Nei grafici sottostanti vengono riportati i risultati per ciascun modello, confrontando MFB50 misurato (linea blu) e predetto (linea rossa) in funzione del tempo.

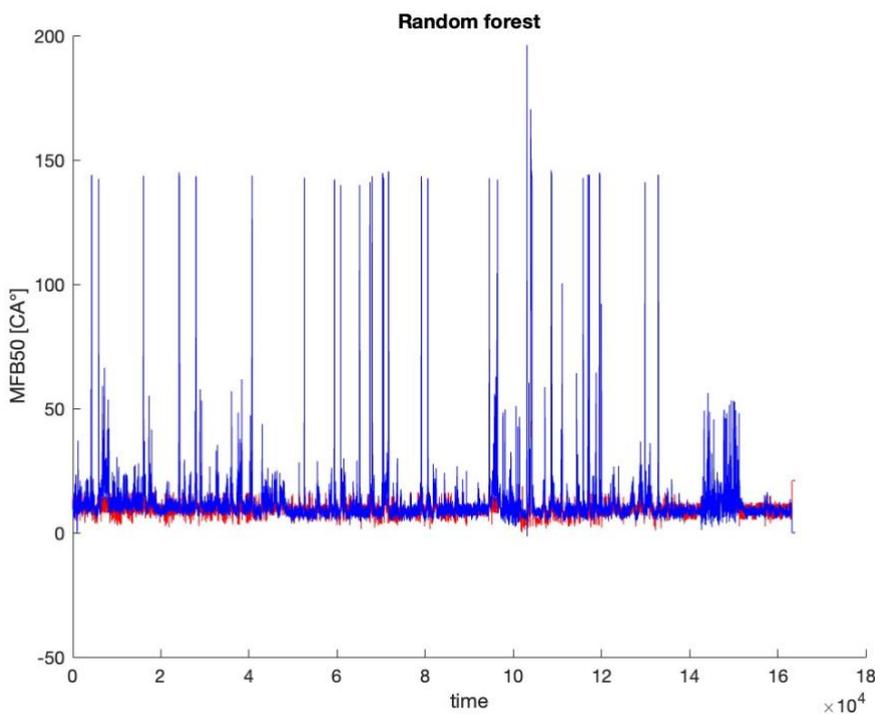
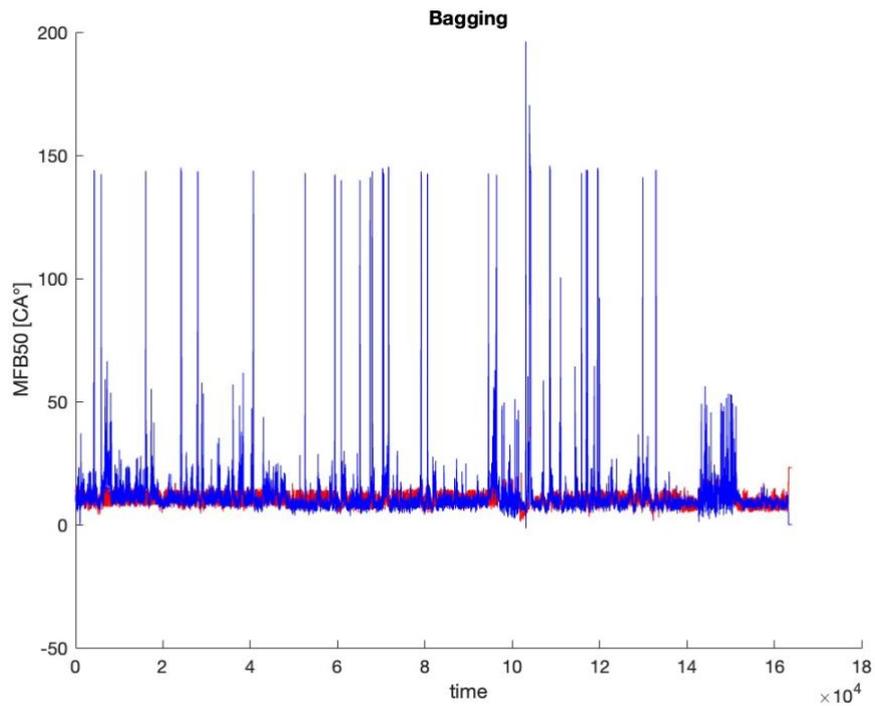
- **Track test**





- **Road test**





Analizzando i risultati, si osservano dei picchi anomali di MFB50 nei grafici finali, specialmente durante rapide variazioni della percentuale del pedale dell'acceleratore, come avviene molto frequentemente nel ciclo di guida su strada. Questi picchi sono causati dal cosiddetto **fuel cut-off**. In breve, è una funzione di sicurezza che interrompe l'alimentazione di carburante al motore in determinate circostanze, principalmente per evitare danni al motore o migliorare l'efficienza. Nel nostro caso, la centralina motore (ECU), con lo scopo di limitare i giri e proteggere il motore, comanda delle mancate iniezioni in fase di rilascio del pedale

dell'acceleratore. Questo avviene soprattutto durante fasi di decelerazione, quando non è necessario fornire energia supplementare. Di conseguenza, non viene bruciato carburante e non vi è alcun rilascio di calore nel cilindro, motivo per cui non dovrebbe essere svolto alcun calcolo dell'MFB50,

Nonostante il fuel cut-off preveda la sospensione dell'iniezione di carburante, il sistema indicating restituisce comunque un valore numerico in tali cicli. Questo comportamento, come spiegato in precedenza, non dovrebbe verificarsi. Di conseguenza, i valori di pressione e MFB50 rilevati durante queste fasi non sono rappresentativi di una reale combustione e dovrebbero essere interpretati con cautela.

5.3. Analisi dei Risultati

Durante la fase di testing, i modelli sono stati applicati a cicli di guida reali, mostrando differenze nei risultati tra il road test e il track test. Nel track test, caratterizzato da condizioni più stabili, LSBoost ha ottenuto un errore RMSE più basso rispetto al Random Forest, mentre nel road test, il Random Forest ha offerto prestazioni migliori, dimostrando una maggiore capacità di generalizzazione in condizioni variabili.

Modello	RMSE road test	RMSE track test
LSBoost	12.9731	3.5615
Bagging	12.5921	3.8278
Random Forest	12.2317	4.2661

Come detto prima, in alcuni cicli, a causa del fuel cut-off, i valori di MFB50 non sono del tutto veritieri e questo spiega il motivo per cui i valori di RMSE sul ciclo di guida su strada risultano essere più elevati rispetto a quelli su pista.

Tuttavia, i modelli di ensemble continuano a fornire previsioni accettabili, dimostrando una buona capacità di generalizzazione anche in condizioni più complesse.

Capitolo 6.

6. Conclusioni

I modelli di ensemble learning hanno dimostrato di essere strumenti efficaci per la modellazione dell'indice MFB50 in contesti automotive. In particolare, i modelli di Bagging e Random Forest hanno mostrato maggiore affidabilità nelle previsioni rispetto a LSBoost, soprattutto in condizioni di test su strada, che presentano maggiore variabilità.

In sintesi, le principali conclusioni sono:

1. Random Forest e Bagging hanno dimostrato una maggiore robustezza e affidabilità nelle previsioni dell'indice MFB50 rispetto a LSBoost.
2. LSBoost, nonostante sia competitivo, ha mostrato una leggera tendenza a sovra-adattarsi nel road test, dove il Random Forest ha offerto prestazioni più equilibrate.
3. La scelta del modello ottimale dipende dalle esigenze specifiche: se l'obiettivo è ridurre la varianza, il Random Forest è la scelta migliore; se invece si ricerca la precisione nel track test, LSBoost può essere altrettanto valido.

Le tecniche di ensemble learning applicate in questo lavoro si sono dimostrate un valido approccio per migliorare l'accuratezza delle previsioni in sistemi complessi, come quelli dei motori a combustione interna, in condizioni di test variabili.

Riferimenti e Bibliografia

- [1] G. Ferrari, Motori a combustione interna, Società Editrice Esculapio, 2016
- [2] J. B. Heywood, Internal Combustion Engine Fundamentals, McGraw-Hill Education, 1988.
- [3] G. Minelli, Motori endotermici alternativi, Pitagora, 1985.
- [4] Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [5] Breiman, L. *Random Forests*. Machine Learning, 45(1), 5-32, 2001.
- [6] Hastie, T., Tibshirani, R., & Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.
- [7] MATLAB Documentation. *fitrensemble - Train Ensemble of Regression Learners*, The MathWorks, Inc.
- [8] MATLAB Documentation. *TreeBagger - Bootstrap Aggregated (Bagged) Decision Trees*, The MathWorks, Inc.
- [9] MATLAB Documentation. *Bayesian Optimization Algorithm, Tune Random Forest Using Quantile Error and Bayesian Optimization*, The MathWorks, Inc.