UNIVERSITA' POLITECNICA DELLE MARCHE

FACOLTÀ DI INGEGNERIA

Corso di Laurea triennale in **INGEGNERIA MECCANICA**

Sviluppo e validazione di un sistema di misura e ottimizzazione dei consumi energetici e del comfort termico basato su normalizzazione climatica heating degree days

Development and validation of a system for measuring and optimizing energy consumption and thermal comfort based on climate normalization using heating degree days

Relatore:

Prof. **Revel Gian Marco**

Tesi di Laurea di:

Monnati Riccardo

Correlatrici:

Dott.ssa **Cipollone Vittoria**

Prof.ssa **Morresi Nicole**

A.A. 2025/2026

ABSTRACT:

Negli edifici reali, la gestione del comfort termico e dei consumi energetici è spesso ostacolata dalla mancanza di misure complete, dall'assenza di sistemi di monitoraggio de consumi e dall'utilizzo di strategie di controllo statiche che non riflettono la dinamica termo-energetica dell'ambiente. In questo contesto, la presente tesi sviluppa e valida una metodologia di misura e controllo completamente data-driven, finalizzata a ricostruire il comportamento termo-energetico dell'edificio e a generare raccomandazioni di gestione dell'impianto.

La metodologia si basa su una campagna di monitoraggio condotta su un edificio reale, durante la quale sono stati misurati dati termo-igrometrici interni, variabili climatiche esterne, voti soggettivi di comfort (TSV) e consumi energetici ricavati dalle bollette mensili tramite Heating Degree Days (HDD). A partire da queste misure è stato costruito un simulatore dell'edificio composto da tre modelli di Machine Learning: modello termico, modello di comfort e modello energetico. Il simulatore costituisce l'ambiente virtuale utilizzato per addestrare un agente di Reinforcement Learning basato su Q-learning, incaricato di apprendere una politica di controllo ON/OFF capace di bilanciare comfort ed efficienza energetica.

I modelli ML hanno mostrato prestazioni solide: il modello termico, incaricato di prevedere l'evoluzione della temperatura interna, ha raggiunto un MAE pari a 0.22 °C e un R^2 pari a 0.92; il modello di comfort, utilizzato per stimare la condizione di comfort/discomfort percepita dagli occupanti, ha ottenuto un'accuracy pari a 0.88 e un F1-score pari a 0.93; il modello energetico, impiegato per stimare i consumi associati alle azione dell'impianto, ha fornito consumi coerenti con le accensioni simulate. La politica RL appresa, valutata esclusivamente in simulazione ha permesso di migliorare simultaneamente comfort ed efficienza energetica: la percentuale di step in comfort è aumentata dal 45% al 51%, mentre i consumi energetici si sono ridotti da 846 kWh a 505 kWh, con un risparmio complessivo di circa il 40%.

Nel complesso, la tesi dimostra come l'integrazione tra misure reali, modelli data-driven e RL possa costituire una base solida per la progettazione di sistemi di controllo termo-energetico avanzati, capaci di ottimizzare comfort e consumi in edifici reali.

INDICE

INTRODUZIONE 1

1. Stato dell'Arte 3

1.1 Definizione e misura del comfort termico negli edifici 3

1.2 Misura dei consumi energetici in un edificio 5

1.3 Modelli di Machine learning per la misura del comfort e dell'energia nell'ambiente costruito 6

1.4 Ruolo del Reinforcement Learning nel controllo energetico e ottimizzazione del comfort 8

1.5 Limiti degli approcci esistenti 10

1.6 Obiettivi della tesi 12

2. Materiali e Metodi 13

2.1 Dataset reale e preparazione dei dati 16

2.2 Simulatore dell'edificio 19

2.2.1 Modello Termico 20

2.2.2 Modello Comfort 23

2.2.3 Modello energetico 25

2.3 Reinforcement Learning 27

2.3.1 Agente RL 29

2.3.2 Ambiente di Reinforcement Learning 29

2.3.3 Stato 30

2.3.4 Azioni 31

2.3.5 Reward 31

2.4 Algoritmo di Q-learning 32

2.4.1 Q-table e discretizzazione 34
2.4.2 Parametri del Q-learning 37
2.4.3 Procedura di training 39
2.5 Esecuzione della politica RL 40
2.6 Caso Studio 42
3. Risultati 46
3.1 Temperatura: reale vs RL 49
3.2 Comfort: reale vs RL 50
3.3 Consumi: reale vs RL 51
3.4 Azioni: reale vs RL 52
4. Discussione e Conclusione 53
4.1 Discussione dei risultati 53
4.2 Limiti del lavoro e sviluppi futuri 55
5. Bibliografia 58

INTRODUZIONE

Il presente elaborato di tesi affronta il problema della misura e della stima delle grandezze termo-energetiche necessarie alla gestione degli edifici: condizioni termo-igrometriche interne, comfort termico percepito dagli occupanti e consumi energetici. La corretta valutazione di tali grandezze è fondamentale per garantire ambienti confortevoli, e al contempo, contenere i consumi energetici all'interno di edifici.

A livello globale, il settore edilizio è responsabile del 40% della domanda energetica complessiva e i sistemi di riscaldamento, ventilazione e condizionamento dell'aria rappresentano i principali consumatori di energia, arrivando ad assorbire fino al 60% dei consumi totali [1]. Nonostante l'importanza del tema, il controllo degli impianti termici si basa ancora oggi su logiche semplici e tradizionali, come accensioni programmate o regolazioni manuali. Tali approcci, sebbene molto diffusi, risultano essere inadeguati nel bilanciare in modo ottimale comfort ed efficienza energetica, in particolare in edifici con dinamiche termiche complesse, andando a causare sprechi energetici e ritardi nella risposta termica. Inoltre, ancora oggi, la gestione degli impianti termici si basa spesso su misure incomplete o non disponibili, che impediscono una valutazione accurata del comportamento termo-energetico dell'edificio e del comfort percepito dagli occupanti. Molti edifici reali, infatti, risultano essere privi sia di sensori in grado di misurare direttamente i consumi energetici (smart meter), sia di strumenti che permettano di quantificare il comfort termico.

Per superare tali limitazioni, nel presente lavoro viene proposta una metodologia di misura e stima basata su dati reali. In primo luogo, vengono utilizzati i dati termo-igrometrici raccolti durante la campagna di misura per valutare il comfort percepito dagli occupanti tramite la sensazione termica percepita dagli utenti TSV (Thermal Sensation Vote). Parallelamente, vengono sfruttati i dati meteorologici per il calcolo degli Heating Degree Days (HDD), un indicatore del fabbisogno termico giornaliero. Le bollette energetiche mensili dell'edificio sono poi utilizzate per costruire una regressione tra HDD e consumo energetico, consentendo di stimare i consumi anche in assenza di misure energetiche dirette. Tale approccio risulta essere coerente con le analisi condotte da

ISPRA (Istituto Superiore per la Protezione e la Ricerca Ambientale), le quali mostrano una forte correlazione tra Gradi Giorno e consumo energetico [2].

A partire da queste misure reali, è stato sviluppato un simulatore data-driven dell'edificio, costruito tramite modelli di Machine Learning (ML), in grado di modellare e simulare il comportamento termico dell'edificio.

Su tale simulatore è stato infine addestrato un agente di Reinforcement Learning (RL), che utilizza le grandezze misurate e stimate per generare raccomandazioni di controllo dell'impianto orientate al comfort degli occupanti e alla riduzione dei consumi energetici. L'RL rappresenta uno strumento che permette di sfruttare le misure ottenute per proporre strategie di gestione più efficienti rispetto alle logiche tradizionali [3].

Lo scopo fondamentale del presente lavoro di tesi è dunque sviluppare e validare una metodologia di misura data-driven che consenta di stimare il comportamento termo-energetico dell'edificio e il comfort percepito dagli occupanti, anche in assenza di misure energetiche dirette, e utilizzare tali informazioni per fornire all'utente delle raccomandazioni ottimali di accensione e spegnimento dell'impianto in grado di garantire il benessere termico e ridurre i consumi energetici.

L'elaborato si articola in quattro capitoli. Nel primo capitolo viene presentato lo stato dell'arte relativo alle principali questioni affrontate. Nel secondo capitolo vengono descritti i materiali e metodi utilizzati per realizzare il progetto, in particolar modo ci si sofferma su come è stato implementato il sistema sul caso di studio reale. Nel terzo capitolo vengono illustrati i risultati ottenuti. Infine, il quarto capitolo presenta la discussione dei risultati, le limitazioni riscontrate e i possibili sviluppi futuri.

1. Stato dell'Arte

1.1 Definizione e misura del comfort termico negli edifici

La definizione più accettata a livello globale di comfort termico è quella introdotta da Fanger nel 1970, secondo cui "*il comfort è quella condizione mentale di soddisfazione termica dell'individuo nei confronti dell'ambiente circostante*" [4]. Questa definizione viene adottata dalle principali normative; tra le più diffuse e utilizzate abbiamo la UNI 7730 [5], nell'area europea, e la ASHRAE 55 [6], a livello statunitense. Pur partendo da basi metodologiche diverse, entrambe le norme quantificano il comfort affidandosi a due specifici indici di prestazione: il PMV (Predicted Mean Vote) e il PPD (Predicted Percentage of Dissatisfied). Il PMV esprime la sensazione termica media delle persone in un ambiente chiuso, stimata su una scala numerica da -3 (molto freddo) a +3 (molto caldo), con lo 0 che rappresenta il benessere termico. Esso è funzione di sei variabili, quattro ambientali, tra cui la temperatura dell'aria interna (T_a), la temperatura media radiante (T_r), la velocità dell'aria interna (V_a) e l'umidità relativa interna (RH), e le restanti due personali, come la resistenza termica dell'abbigliamento (I_{cl}) e l'attività metabolica (M).

$$PMV = f\,(T_a,\ T_r,\ V_a,\ RH,\ I_{cl},\ M) \qquad (1)$$

Il PPD invece esprime la percentuale di persone insoddisfatte in un determinato ambiente. Esso può essere relazionato al PMV in tal modo:

PMV	PPD
0	< 5%
±1	< 26%
±2	< 73%

Tabella 1. Relazione tra PMV e PPD

Sebbene entrambi gli indici siano ampiamente utilizzati nella letteratura scientifica per valutare il comfort in ambiente controllati, diversi studi mostrano che quando l'indice PMV viene confrontato con le reali sensazioni termiche rilevate sul campo, presenta

discrepanze significative [7]. Ciò è dovuto alla complessità delle variabili coinvolte, spesso difficili da monitorare in modo continuo negli edifici reali.

Per questo motivo negli ultimi anni si è diffuso l'utilizzo di indicatori semplificati, capaci di rappresentare il comfort termico in modo più diretto e facilmente misurabile. Tra questi spicca il modello sPMV sviluppato da Buratti e colleghi che permette di stimare il comfort termico utilizzando solamene due variabili: la temperatura e l'umidità dell'aria [8]. Il modello di Buratti, tuttavia, non risulta sufficientemente accurato poiché non considera le diversità presenti tra le zone climatiche. Per questo motivo sono stati sviluppati dei modelli di sPMV ottimizzato che tengono conto dell'adattamento dell'individuo al clima del luogo in cui vive [9].

Un altro parametro per la valutazione del comfort è il *"Thermal Sensation Vote" (TSV),* il quale rappresenta un voto che indica la percezione termica soggettiva dell'occupante espressa su la scala ASHRAE [6] a 7 punti, da -3 (molto freddo) a +3 (molto caldo):

Scale	Thermal Sensation
+3	Hot
+2	Warm
+1	Slightly Warm
0	Neutral
-1	Slightly Cool
-2	Cool
-3	Cold

Figura 1. Scala voti TSV [10].

A differenza del PMV, che misura il comfort partendo da parametri oggettivi e considerando la sensazione media di un gruppo di occupanti, il TSV è un giudizio individuale raccolto durante esperimenti o campagne di misura, e quindi risulta particolarmente adatto quando si vuole caratterizzare il comfort percepito in edifici reali a partire da dati misurati.

Nel presente lavoro il TSV è stato adottato come parametro principale per la misura del comfort termico, poiché rappresenta direttamente la sensazione termica percepita dal singolo occupante. A differenza degli indici derivati da modelli fisici, il TSV consente di quantificare il comfort in modo immediato e basato sull'esperienza reale dell'utente, elemento centrale nella gestione energetica di un edificio. Infatti, le decisioni di controllo, come accendere e spegnere il riscaldamento, dipendono dalla percezione termica individuale [11] [12]: un occupante che esprime un voto negativo tenderà ad aumentare la temperatura fino a raggiungere una condizione neutra. Per questo motivo, il TSV costituisce una misura particolarmente utile sia per caratterizzare il comfort durante la campagna di monitoraggio, sia per sviluppare modelli predittivi e strategie di controllo orientate al benessere dell'utente e alla riduzione dei consumi energetici.

1.2 Misura dei consumi energetici in un edificio

Parallelamente alla misura del comfort, il calcolo dei consumi energetici rappresenta un altro tema fondamentale. La letteratura evidenzia come i consumi dipendano da vari parametri come le condizioni esterne, l'isolamento dell'edificio, il comportamento degli utenti e la strategia con cui viene regolato l'impianto [13]. Tuttavia, un problema ricorrente negli edifici reali è la mancanza di contatori intelligenti o sistemi di monitoraggio in grado di misurare direttamente l'energia utilizzata dall'impianto di riscaldamento. Per sopperire a questa mancanza diversi studi propongono l'utilizzo del metodo Heating Degree Days (HDD), che si dimostra uno degli strumenti più efficaci per normalizzare i consumi rispetto alle condizioni climatiche [2]. Dal punto di vista matematico, gli HDD consistono nella sommatoria delle differenze tra una temperatura di riferimento (T_{rif} °C, in genere 18°C o 20°C) e la temperatura media (T_m, °C) esterna, su base giornaliera:

$$HDD = \mathbf{\Sigma}\ (Trif - Tm) \qquad (2)$$

Gli HDD quantificano il fabbisogno termico giornaliero necessario a mantenere condizioni interne comfortevoli. Anche analisi condotte da ISPRA confermano la forte correlazione tra gradi giorno e consumi energetici, mostrando come l'andamento dei consumi segua in modo coerente la variazione dei gradi giorno, come si può evincere dal grafico sottostante [2]:

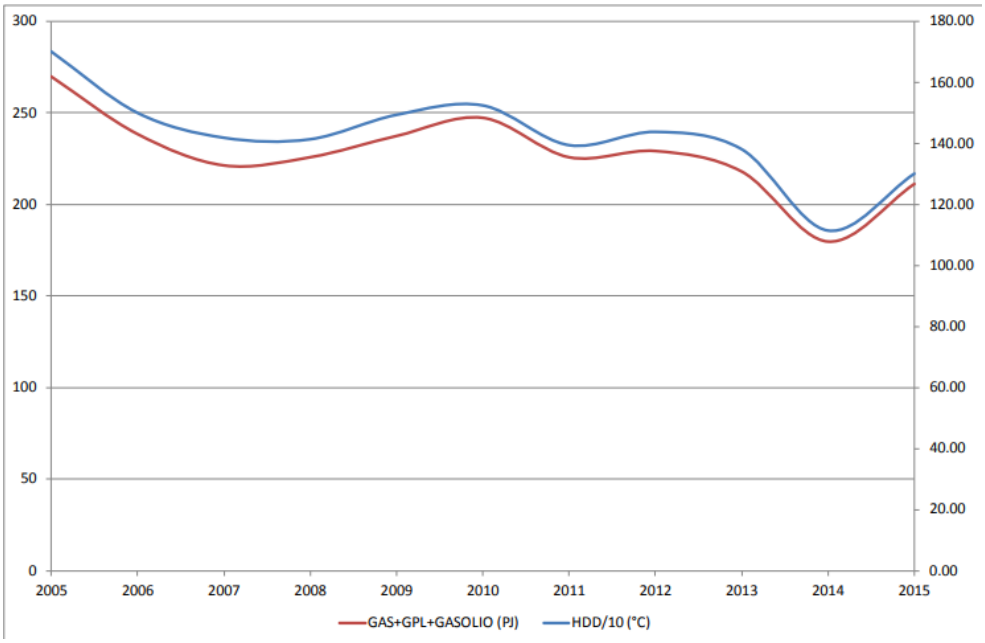


Figura 2. Andamento HDD e consumi di combustibili [2].

Dal punto di vista applicativo, una volta calcolati gli HDD è possibile stimare i consumi energetici tramite dei modelli di regressione che mettono in relazione il fabbisogno termico (espresso dagli HDD) con i consumi reali misurati e ricavati dalle bollette.

In questa tesi sono stati utilizzati gli HDD e i consumi ricavati dalle bollette mensili per costruire un modello di regressione in grado di stimare l'andamento dei consumi energetici dell'edificio, anche in assenza di misure dirette. Integrando la misura del comfort (TSV), con la stima dei consumi tramite HDD, si riesce a ottenere una base solida per l'ottimizzazione del sistema di riscaldamento tramite l'utilizzo del RL.

1.3 Modelli di Machine learning per la misura del comfort e dell'energia nell'ambiente costruito

Negli ultimi anni, il ML è diventato uno strumento fondamentale nella modellazione degli edifici. A differenza dei modelli fisici tradizionali, che esigono conoscenze dettagliate delle proprietà termiche dell'edificio e delle condizioni a contorno, gli approcci data-driven sfruttano esclusivamente lo storico delle misurazioni e permettono di descrivere il comportamento dell'edificio anche in presenza di dinamiche non lineari o difficili da modellare analiticamente [13].

La letteratura mostra che il ML sia ampiamente utilizzato per stimare il comfort percepito dagli occupanti, integrando variabili ambientali e climatiche. Deb et al. evidenziano che i modelli data-driven possono ricostruire parametri di comfort a partire da misure facilmente disponibili, grazie alla loro capacità di catturare relazioni non lineari tra temperatura, umidità e percezione termica [13]. Lu et al. dimostrano che modelli come Random Forest e reti neurali possono prevedere la sensazione termica degli occupanti utilizzando dataset di voti soggettivi (TSV), risultando particolarmente efficaci nella modellazione del comfort percepito [14]. Boutahri e Tilioua confermano che il ML può stimare il comfort termico utilizzando esclusivamente variabili ambientali, rendendolo uno strumento centrale nella caratterizzazione termo-igrometrica degli edifici [15].

Il ML è ampiamente impiegato anche per la previsione dei consumi energetici, sfruttando dati climatici, caratteristiche dell'edificio e misure storiche di energia. Deb et al. mostrano come modelli quali Random Forest e Gradient Boosting, siano particolarmente efficaci nel prevedere consumi giornalieri e mensili, grazie alla loro capacità di gestire dataset complessi e variabili climatiche eterogenee [13]. Boutahri e Tilioua evidenziano che modelli ML possono stimare i consumi energetici partendo da variabili ambientali e parametri di comfort, integrando in modo efficace le informazioni climatiche e il comportamento degli occupanti [15]. Questi approcci permettono di stimare il comportamento energetico dell'edificio anche quando non sono disponibili misure dirette dei consumi, come nel caso di edifici privi di contatori intelligenti.

La previsione della temperatura interna è un altro ambito in cui il ML ha dimostrato grande efficacia. Lu et al. mostrano che modelli data-driven possono simulare la dinamica termica interna a partire da variabili climatiche esterne, stato dell'impianto e caratteristiche dell'edificio [14]. Studi recenti evidenziano che modelli come Random Forest sono particolarmente robusti in presenza di rumore o misure irregolari, mentre le reti neurali risultano adatte alla modellazione di serie temporali e alla previsione della temperatura su orizzonti estesi [15].

In particolare, il modello Random Forest è considerato uno dei modelli più robusti ed affidabili nell'ambito del comfort termico e ottimizzazione energetica degli edifici [15]. I suoi punti di forza sono elencati di seguito:

- Robustezza con dati reali: la Random Forest gestisce bene condizioni di rumore o misure non perfettamente regolari, riducendo il rischio di overfitting [13].
- Capacità di catturare relazioni non lineari: ad esempio, la percezione del comfort (TSV) dipende in modo non lineare da temperatura interna e umidità. La Random Forest è efficace nel modellare tali relazioni [14].
- Interpretabilità e stabilità: la Random Forest permette di analizzare l'importanza delle variabili, facilitando la comprensione del comportamento del modello.

Nel presente lavoro, il Machine Learning è stato utilizzato per costruire un simulatore data-driven dell'edificio, composto da tre modelli distinti:

- Modello termico, per prevedere la temperatura interna;
- Modello di comfort, per stimare il TSV;
- Modello energetico, per stimare i consumi.

Questi modelli, addestrati sui dati raccolti durante la campagna di misura, costituiscono la base della metodologia proposta e permettono di ricostruire il comportamento termo-energetico dell'edificio anche in assenza di misure dirette dei consumi. Il simulatore rappresenta quindi l'ambiente su cui verranno successivamente sviluppate e validate le strategie di controllo basate sul RL.

1.4 Ruolo del Reinforcement Learning nel controllo energetico e ottimizzazione del comfort

Determinare un compromesso adeguato tra consumo energetico e comfort termico interno rappresenta uno degli aspetti centrali da considerare per il controllo degli impianti termici. La letteratura evidenzia come la regolazione della temperatura interna influenzi direttamente sia il benessere degli occupanti sia il fabbisogno energetico dell'edificio, rendendo necessario un controllo dinamico capace di adattarsi alle condizioni climatiche e alle preferenze individuali [1], [15].

Fino a pochi anni fa, la maggior parte degli edifici utilizzava tecniche di controllo tradizionali, basate sul valore di impostazione statico della temperatura, per gestire i sistemi di riscaldamento e raffreddamento. Tuttavia, con questa soluzione, veniva

costruito un ambiente termico uguale per tutti gli ospitanti e veniva ignorata la preferenza termica individuale [1].

L'apprendimento per rinforzo (RL), è stato ampliamente applicato nell'ingegneria del controllo reale. Alcuni ricercatori lo hanno utilizzato per ottimizzare l'apertura e la chiusura delle finestre [16], altri hanno proposto un algoritmo RL per il controllo combinato di ventilatori e condizionatori [17]. Il suo punto di forza risiede nella capacità di apprendere strategie di regolazione ottimali direttamente dall'interazione con l'ambiente, andando a migliorare continuamente le proprie prestazioni e richiedendo un intervento umano minimo [18]. I principali vantaggi, che emergono dalla letteratura, nell'utilizzo di un sistema di controllo RL sono:

- Apprendimento continuo: l'algoritmo RL interagisce con l'edificio e con l'impianto, migliorando le decisioni col passare del tempo [16], [17].
- Gestione predittiva: l'agente RL riesce a prevedere le dinamiche dell'edificio e quindi ad attivare l'impianto in anticipo [18].
- Maggiore stabilità della temperatura interna: diversi studi mostrano che l'RL riduce le fluttuazioni termiche rispetto ai controlli tradizionali [17], [18].

A seconda del particolare caso in esame, per il controllo termo-energetico di un edificio, la letteratura distingue tra due grandi famiglie di algoritmi di Reinforcement Learning: Model-Free e Model-Based. Il primo impara la politica di controllo interagendo direttamente con l'edificio reale o con simulatori fisici dettagliati (EnergyPlus, Modelica). Il suo punto di forza è riuscire ad apprendere politiche ottimali anche in ambienti complessi [19]. D'altra parte, però, richiede lunghi tempi di addestramento e può comportare rischi per il comfort degli occupanti durante la fase iniziale di training, poiché l'agente deve provare azioni potenzialmente non ottimali direttamente sull'edificio reale [16], [20]. Il secondo invece, il Model-Based RL, prevede l'utilizzo di un simulatore dell'edificio, costruito con dei modelli ML, tramite il quale l'agente può imparare la politica di controllo. Dal momento che non richiede interazioni con l'edificio reale, il Model-Based RL non comporta alcun rischio per gli occupanti, permette un training molto più rapido e risulta ideale quando si hanno a disposizione solo dati storici [19]. D'altra parte, esso può risultare meno performante del Model-free se il modello è impreciso. Nel nostro preciso caso in esame, per esigenze tecniche è possibile applicare

solamente il Model-Based RL, dal momento che l'edificio non dispone di sensori energetici, i dati disponibili sono statici e non è possibile far esplorare l'edificio reale all'algoritmo RL a causa dell'assenza di un sistema HVAC automatizzato.

Tra gli algoritmi di RL più utilizzati nel controllo degli edifici è presente il Q-learning: un metodo tabellare che permette all'agente di apprendere la politica ottimale associando a ogni coppia stato-azione un valore di qualità (Q-value). Esso è particolarmente adatto al Model-Based RL dal momento che funziona bene con simulatori data-driven, richiede solo la discretizzazione dello stato, è semplice, stabile e facilmente interpretabile. Esso è stato utilizzato con successo in diversi studi su edifici reali[1], [20], dimostrando la sua efficacia nel tagliare i consumi e migliorare il comfort.

1.5 Limiti degli approcci esistenti

Nonostante i numerosi passi in avanti fatti negli ultimi anni, l'utilizzo del ML e del RL per il controllo termico ed energetico degli edifici presenta ancora diverse criticità.

Un primo ostacolo riguarda la disponibilità e la qualità dei dati. In primo luogo, la mancanza di dati energetici affidabili rappresenta uno dei principali limiti per l'uso di queste tecnologie: molti edifici residenziali non dispongono di smart meters o sensori dedicati alla misura dell'energia termica, rendendo difficile stimare i consumi in modo diretto [2]. Inoltre, dataset incompleti, rumorosi, o non sincronizzati potrebbero compromettere l'accuratezza dei modelli ML, impattando poi sulla qualità del controllo RL [13].

Un ulteriore limite riguarda la misura del comfort. La maggior parte dei sistemi di controllo collega il comfort ai valori di PMV/PPD, ignorando la percezione termica individuale. Broday et al. mostrano che il PMV può discostarsi significativamente dalle sensazioni reali degli occupanti, rendendo difficile modellare correttamente il legame tra comfort percepito e consumo energetico [7]. Questo comporta che i modelli ML e RL basati su PMV possano non riflettere il comportamento reale degli utenti, che invece regolano l'impianto in base alla propria sensazione termica.

Altri limiti vengono fuori dal tipo di approccio RL utilizzato. Gao e Wang evidenziano che gli algoritmi Model-Free RL richiedono un numero elevato di interazioni con

l'ambiente per apprendere una politica di controllo stabile [19]. Ciò comporta tre problematiche principali:

- Esplorazione rischiosa: durante le fasi iniziali del training è molto probabile che l'algoritmo compia delle decisioni sbagliate, andando a generare sprechi di energia o condizioni di discomfort per gli occupanti dell'edificio [16].
- Tempi di apprendimento molto lunghi: l'addestramento spesso richiede migliaia di episodi, spesso incompatibili con l'uso in edifici reali [19].
- Dipendenza dai simulatori: queste tecniche funzionano bene solo se si ha a disposizione un simulatore fisico (EnergyPlus, Modelica) [3].

Anche la soluzione alternativa, cioè il Model-Based RL, presenta dei limiti. In particolare, la principale criticità consiste nella costruzione e nella validazione di un buon modello, aspetto che va ad influenzare in modo diretto la qualità del controllo [19]. Innanzitutto, per costruire un buon simulatore occorre una fase iniziale di raccolta dati di ottima qualità, in modo tale da evitare problemi comuni come l'overfitting, la scarsa generalizzazione o la sensibilità al rumore [13]. Se il modello ML non cattura correttamente le dinamiche termiche, l'algoritmo RL imparerà strategie subottimali o non collegate al caso reale. Inoltre, eventuali errori di previsione del modello ML si propagano nel processo decisionale dell'agente RL, andando quindi ad avere un impatto negativo su tutto il lavoro.

Un'altra problematica sono i limiti degli algoritmi esistenti. Ad esempio, il Q-learning, pur essendo semplice e tradizionale, presente dei limiti pratici evidenti: richiede la discretizzazione dello stato, che può diventare complessa, fatica se gli spazi di stato sono continui o molto grandi, e converge lentamente in ambienti con dinamiche lente. Si potrebbe pensare a questo punto di utilizzare degli algoritmi di Deep RL. Tuttavia, essi, pur garantendo una maggiore potenza, presentano altri limiti: richiedono dataset molto più ampi, la fase di tuning spesso risulta essere complessa, potrebbero essere instabili durante l'addestramento e richiedono potenza di calcolo elevata. Proprio per questa serie di motivi, in ambito residenziale, il Deep RL risulta essere una soluzione eccessiva rispetto al reale bisogno e ai dati disponibili [18].

Infine, un limite comune a tutti gli approcci riguarda la generalizzazione delle politiche di controllo. L'obiettivo di tutti questi progetti è, partendo da un esperimento su un singolo

edificio, riuscire a ricavare una strategia di controllo che possa essere estesa a più famiglie di edifici. Tuttavia, ciò non sempre è realizzabile anzi, nella maggior parte dei casi ogni edificio fa storia a sé e richiede un suo addestramento specifico, rendendo difficile generalizzare la politica appresa [3].

1.6 Obiettivi della tesi

L'obiettivo generale di questa tesi è sviluppare un sistema di controllo intelligente in grado di fornire all'utente raccomandazioni personalizzate di gestione dell'impianto di riscaldamento, basate su valori di temperatura ottimali che garantiscano contemporaneamente comfort termico e riduzione dei consumi energetici. L'idea centrale è sostituire le logiche di controllo tradizionali, basate su set-point statici, con un approccio data-driven capace di adattarsi alle condizioni reali dell'edificio e alla percezione termica dell'occupante.

Per raggiungere questo obiettivo finale, il lavoro è stato articolato in quattro attività principali, ciascuna strettamente legata alla fase di misura e alla modellazione dei dati raccolti.

La prima attività ha riguardato la campagna di misura svolta nell'edificio reale, finalizzata a raccogliere dati ambientali (temperatura interna, temperatura esterna, umidità, radiazione solare, CO_2) e dati soggettivi di comfort (TSV). Questa fase è fondamentale perché l'intero progetto si basa sulla misura diretta del comportamento termo-energetico dell'edificio e della percezione termica dell'occupante. Senza dati reali, non sarebbe possibile costruire modelli affidabili né simulare il comportamento dell'edificio.

Poiché l'edificio non dispone di smart meters o sensori energetici dedicati, la seconda attività ha riguardato la ricostruzione dei consumi energetici attraverso gli Heating Degree Days (HDD) e i consumi ricavati dalle bollette mensili. Questa fase permette di ottenere una stima coerente del fabbisogno energetico dell'edificio, indispensabile per valutare l'efficacia del controllo e per integrare il consumo energetico nella funzione di ricompensa dell'algoritmo RL.

La terza attività è la costruzione di un simulatore data-driven dell'edificio, basato su tre modelli di Machine Learning, che sia in grado di simulare il comportamento fisico dell'edificio. Esso ha il compito di prevedere la temperatura interna, il comfort termico e

i consumi energetici, diventando la base su cui verrà sviluppato l'algoritmo di controllo. Il simulatore è essenziale perché, non potendo far interagire l'agente con l'edificio reale, rappresenta l'unico ambiente sicuro e coerente in cui l'algoritmo può apprendere.

La quarta attività sarà quello di creare e addestrare un algoritmo di Reinforcement Learning, basato sul Q-learning, che sia in grado di imparare, tramite un sistema di azione-ricompensa, come gestire l'impianto di riscaldamento in maniera ottimale, capendo quando è conveniente accendere l'impianto e quando spegnerlo, al fine di mantenere il comfort nella zona desiderata (-0.5< TSV <0.5) e ridurre i consumi.

L'ultimo obiettivo è valutare se la politica appresa dall'algoritmo RL permette di controllare l'edificio con risultati migliori rispetto alla gestione reale. A tal fine sono stati confrontati:

- La temperatura ottenuta con la politica RL con quella reale raccolta durante la campagna di misura,
- La percentuale di stati in comfort nel caso reale e nel caso simulato,
- I consumi energetici simulati dall'algoritmo con quelli reali,
- Le azioni di controllo (ON/OFF) reali e quelle definite dalla politica RL.

Questa analisi permette di verificare se l'algoritmo è in grado di proporre raccomandazioni di controllo più efficienti, basate su temperature ottimali e su una gestione intelligente dell'impianto.

2. Materiali e Metodi

Questa sezione è dedicata alla presentazione della metodologia sviluppata per analizzare il comportamento termo-energetico dell'edificio e per costruire il sistema di controllo basato su RL applicato al caso studio. L'approccio integra in modo sequenziale diverse attività: la raccolta dei dati in ambiente reale, la preparazione dei dati, la costruzione dei modelli predittivi e, infine, lo sviluppo dell'algoritmo di controllo.

La metodologia parte dalla campagna di misura, attraverso la quale sono stati acquisiti i dati ambientali e termici necessari per caratterizzare il funzionamento dell'edificio. Questi dati costituiscono la base per la fase successiva, dedicata alla preparazione e

normalizzazione delle variabili, incluse le misure climatiche esterne e la stima dei consumi energetici tramite HDD.

A partire dai dati raccolti sono stati sviluppati dei modelli ML con l'obiettivo di ricostruire il comportamento dell'edificio. In particolare, sono stati utilizzati per prevedere l'evoluzione della temperatura interna, stimare il comfort percepito dagli occupanti e valutare i consumi energetici. L'insieme di questi modelli costituisce il simulatore data-driven, che rappresenta l'ambiente virtuale utilizzato per l'addestramento dell'algoritmo di RL.

La fase successiva riguarda la definizione dell'ambiente RL, che include la scelta delle variabili di stato, delle azioni disponibili e della funzione di ricompensa. Una volta definito l'ambiente, è stato sviluppato e addestrato l'algoritmo di controllo, che apprende una politica di gestione dell'impianto basata sull'interazione con il simulatore.

Per illustrare in modo intuitivo la strategia intrapresa e le scelte fatte, la figura 3 mostra il block diagram completo del progetto. In esso sono sintetizzate la sequenza delle fasi principali e le relazioni tra i diversi componenti del sistema: dalla raccolta de dati alla costruzione dei modelli, fino all'addestramento e alla validazione dell'algoritmo.

Nei paragrafi successivi, ciascuna fase verrà analizzata nel dettaglio, seguendo la struttura illustrata nel diagramma.

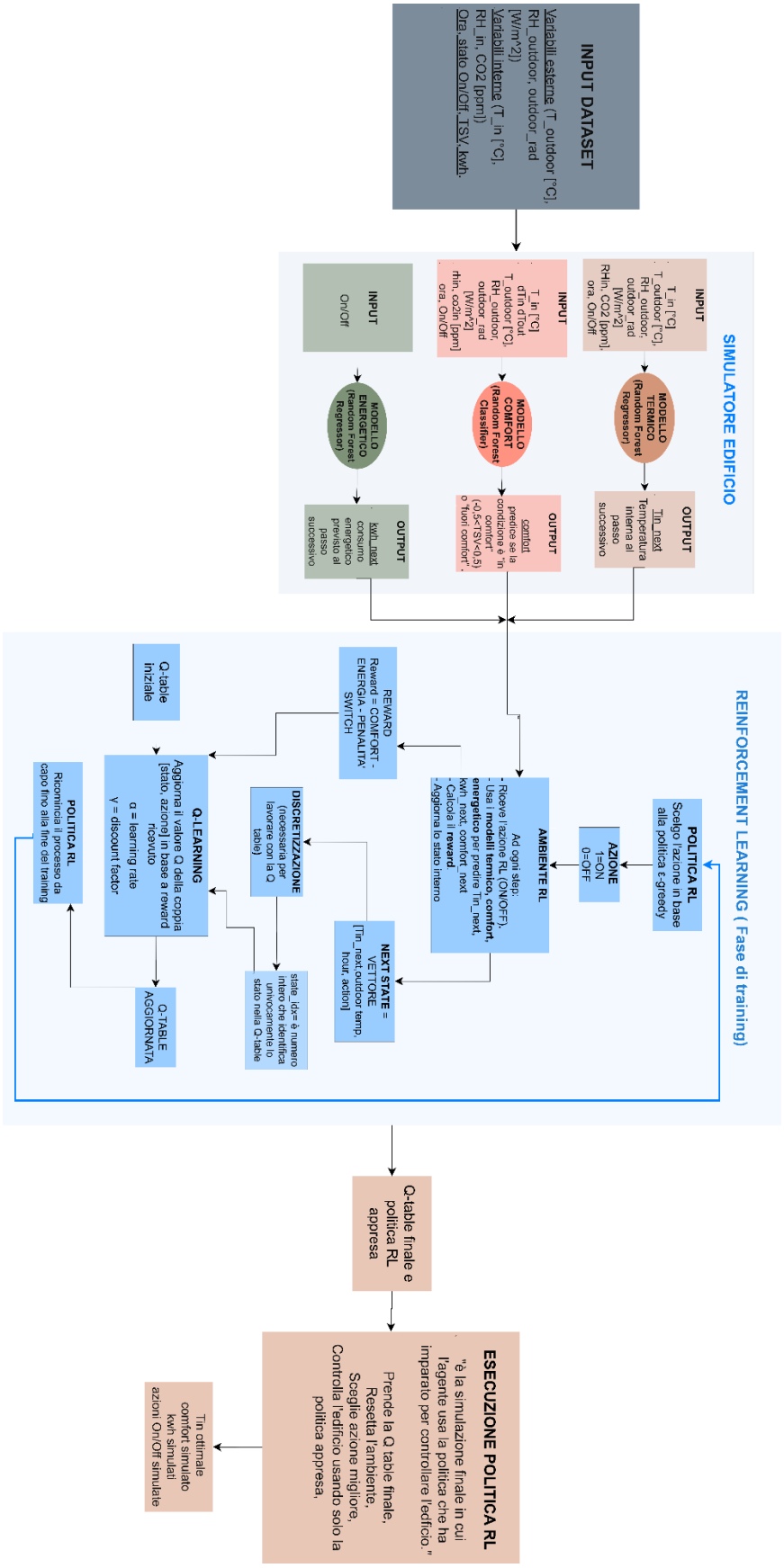


Figura 3. Schema a blocchi della metodologia sviluppata

2.1 Dataset reale e preparazione dei dati

I dati rappresentano il punto di partenza del lavoro e hanno un ruolo cruciale poiché la tipologia e la qualità influenza l'efficacia dei modelli ML e RL. Il dataset comprende i dati raccolti in un edificio di uffici, in Spagna, durante una campagna di misura svolta nel periodo di maggio 2025, come verrà descritto nel paragrafo 2.6. I dati raccolti durante la campagna possono essere suddivisi in quattro categorie principali:

CATEGORIA	VARIABILE	NOME NEL DATASET	UNITÀ DI MISURA	DESCRIZIONE
Dati ambientali indoor	Temperatura interna	T_in	°C	Misura della temperatura dell'aria all'interno dell'edificio
	Umidità relativa interna	RH_in	%	Percentuale di umidità presenta nell'ambiente interno
	Concentrazione di CO_2 interna	CO_2_in	ppm	Indicatore della qualità dell'aria e dell'occupazione
	Stato dell'impianto	Rf_on_any	ON/OFF	Indica se il sistema di riscaldamento è attivo o spento
Dati meteorologici esterni	Temperatura esterna	T_out	°C	Fondamentale per il calcolo degli HDD
	Umidità relativa esterna	RH_out	%	Percentuale di umidità presente nell'aria esterna
	Radiazione solare	Global_rad	W/m^2	Energia solare incidente sull'edificio
Comfort termico	Thermal Sensation Vote	TSV	Scala (da+3 a -3)	Voto soggettivo di comfort termico espresso dagli occupanti
Dati energetici	Consumi mensili ricavati dalle bollette		kWh	Fondamentali insieme agli HDD per ricavare i consumi istantanei.

Tabella 2. Dati raccolti durante la campagna

Tutti i dati sopra citati (tranne i consumi mensili) sono stati raccolti ad intervalli di un'ora per ogni giorno fino alla fine della campagna di misura.

Il primo step è stato l'analisi dei dati disponibili e la loro preparazione per le fasi successive di modellazione. Poiché si tratta di una tesi basata su misure sperimentali, questa fase è di fondamentale importanza: un dataset pulito, ordinato e fisicamente coerente è la condizione necessaria per ottenere modelli affidabili e un simulatore stabile. Per realizzare ciò ho eseguito i seguenti passaggi:

- **Ricostruzione del dataset grezzo**: il file originale contenente i dati provenienti dalla campagna di misura non era immediatamente utilizzabile: le informazioni erano contenute in un'unica colonna, con righe formattate come testo CSV. Per prima cosa ho quindi trasformato il file originale in un DataFrame con una colonna per ogni variabile misurata.
- **Pulizia e organizzazione dei dati**: una volta ricostruita la tabella sono state effettuate operazioni di pulizia come: rimozione di colonne spurie create automaticamente da Excel, ordinamento cronologico delle misure. Questi passaggi garantiscono che il dataset sia ordinato e pronto per l'analisi.
- **Analisi esplorativa (EDA)**: sono stati esaminati i tipi dato (numerico, testuale, temporale), la presenza di valori mancanti, la percentuale di NaN (Not A Number) per ogni variabile, eventuali date non convertibili. Questa analisi permette di individuare errori di misure, sensori malfunzionanti o problemi di registrazione.
- **Controllo della coerenza fisica**: oltre ai controlli statistici, è stato verificato che le misure fossero fisicamente plausibili, cioè compatibili con il comportamento dell'edificio. Sono stati verificati ad esempio che la temperatura interna non presentasse salti improvvisi non giustificabili, e che non fossero presenti valori fisicamente impossibili (es. umidità > 100%).
- **Visualizzazione delle serie temporali**: le principali variabili sono state rappresentate graficamente nel tempo. La visualizzazione permette di: individuare outliers, verificare la continuità delle misure, controllare la coerenza tra le variabili.

Oltre ai parametri ambientali interni ed esterni, è stato necessario integrare nel dataset anche le informazioni ricavate dalle bollette energetiche mensili, che rappresentano

l'unica fonte disponibile per stimare il consumo reale dell'edificio, dal momento che nel caso studio analizzato, i consumi termici non vengono memorizzati in modo continuo a causa dell'assenza di smart meter.

Le informazioni contenute nelle bollette sono state utilizzate per verificare la relazione tra consumo energetico (kWh) e condizioni climatiche esterne, espresse tramite gli Heating Degree Days (HDD). L'analisi ha mostrato una correlazione fortemente lineare tra le due variabili ($R^2 = 0.956$), confermando che il consumo dell'edificio dipende principalmente dalla severità climatica del periodo. Questo risultato, giustificato con la figura 4, dimostra che la normalizzazione climatica adottata è coerente e che gli HDD possono essere utilizzati come variabile predittiva per stimare i consumi energetici.

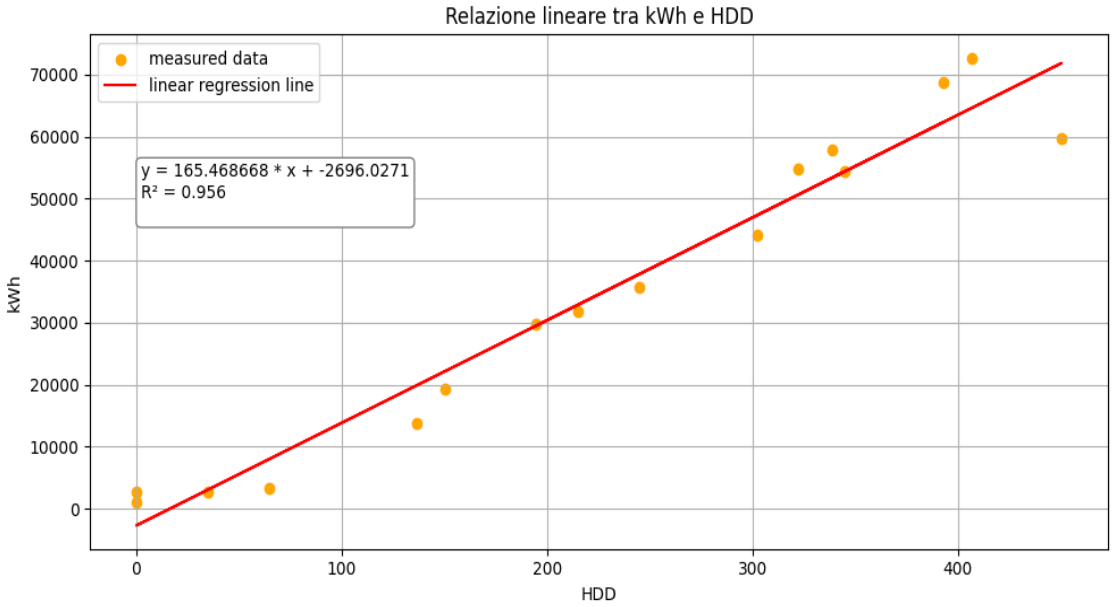


Figura 4. Relazione lineare tra kWh e HDD.

Questa fase è particolarmente importante perché il modello energetico utilizzato nel simulatore dell'edificio, e successivamente integrato nell'algoritmo di Reinforcement Learning, si basa proprio sulla relazione tra HDD e consumo reale. In altre parole, il dato energetico utilizzato dall'RL non proviene da misure dirette, ma viene ricostruito tramite HDD, grazie alla correlazione verificata con i consumi ricavati dalle bollette.

Il risultato di tutte le operazioni descritte sopra è un dataset unico e completo, organizzato come una serie temporale e contenente tutte le variabili misurate durante la campagna di

monitoraggio. Questo dataset rappresenta la base su cui si fonda l'intera metodologia: partendo da esso vengono costruiti i modelli ML e l'ambiente RL.

2.2 Simulatore dell'edificio

La simulazione delle prestazioni degli edifici (BPS) è definita come una simulazione software basata sulla fisica che analizza le prestazioni dei sistemi di un edificio, incorporando input quali le caratteristiche dell'edificio, i programmi di occupazione e le condizioni ambientali per prevedere parametri come il consumo energetico e il comfort termico [21]. Esso rappresenta l'ambiente virtuale all'interno del quale viene addestrato il modello RL.

Lo sviluppo del simulatore data-driven nasce direttamente dalle caratteristiche del caso studio. L'intero progetto si basa infatti sui dati raccolti durante la campagna di misura in un edificio reale che, pur disponendo di sensori ambientali interni ed esterni, non era dotato di smart meter per la registrazione continua dei consumi energetici. In assenza di misure energetiche istantanee e non potendo far interagire l'algoritmo con l'edificio reale, si è reso necessario costruire un modello digitale dell'edificio capace di riprodurne il comportamento termico ed energetico sulla base dei dati misurati.

Il simulatore data-driven, basato sui modelli ML, consente di:

- Ricostruire l'evoluzione della temperatura interna e del comfort a partire dai dati reali;
- Stimare il fabbisogno energetico tramite la relazione tra HDD e consumi ricavati dalle bollette;
- Fornire all'algoritmo RL un ambiente coerente, sicuro e controllabile in cui apprendere una strategia di gestione dell'impianto.

Ciascun modello ML si occupa della previsione di una specifica variabile: temperatura interna, comfort termico e consumo energetico. L'integrazione di questi tre modelli va a rappresentare un ambiente realistico con il quale l'algoritmo RL può interagire.

Dal punto di vista operativo, il simulatore opera seguendo una sequenza ben precisa. A ogni passo temporale:

- Riceve lo stato corrente dell'edificio.

- Riceve l'azione scelta dall'agente RL, ovvero l'accensione o lo spegnimento dell'impianto.
- Elabora lo stato e l'azione tramite i modelli ML, dove ciascuno di essi restituisce la previsione della propria variabile al passo successivo.
- Aggiorna lo stato dell'ambiente, che viene restituito all'agente insieme alla ricompensa.

Questo meccanismo permette all'agente di esplorare diverse strategie di controllo e di apprendere una policy ottimale basata sulla massimizzazione della ricompensa e sull'equilibrio tra comfort e consumo energetico.

Nei paragrafi successivi saranno descritti in modo dettagliato ciascuno dei tre modelli ML utilizzati per costruire il simulatore.

2.2.1 Modello Termico

Il modello termico è uno dei tre modelli del simulatore e ha il compito di prevedere la temperatura interna dell'edificio. La temperatura interna ha un'importanza notevole per il controllo dell'impianto di riscaldamento, da essa infatti dipendono sia il comfort degli occupanti che il consumo energetico. Per questo motivo la costruzione di un modello termico valido e stabile è fondamentale ai fini di una buona riuscita del lavoro.

L'algoritmo selezionato è il Random Forest Regressor, implementato in linguaggio di programmazione python. Si è scelto di utilizzare questo tipo di algoritmo per queste ragioni principali:

- Robustezza al rumore: riesce a gestire bene dataset non regolari [22].
- Capacità di catturare relazioni non lineari complesse [13].
- Stabilità e interpretabilità: permette di analizzare l'importanza delle feature.
- Ottimi risultati su dataset medio piccoli, come quello disponibile [23].

Il modello è stato addestrato con le variabili a disposizione contenute nel dataset iniziale. La tabella sottostante racchiude le feature utilizzate come input e la variabile target (label) da prevedere:

TIPO	VARIABILE	NOME NEL DATASET	UNITÀ DI MISURA
Input	Temperatura interna	T_in	°C
Input	Temperatura esterna	T_out	°C
Input	Umidità interna	RH_in	%
Input	Umidità esterna	RH_out	%
Input	Radiazione solare	Global_rad	W/m^2
Input	CO_2 interna	CO_2_in	ppm
Input	Heating Degree Days	HDD	
Input	Ora del giorno	hour	h
Input	Stato impianto	Rf_on_any	ON/OFF
Label	Temperatura interna futura	T_in(t+1)	°C

Tabella 3. Feature e lable del modello termico

Queste feature permettono al modello di rappresentare la dinamica termica dell'edificio, garantendo una previsione ottimale della temperatura interna futura.

Gli iperparametri utilizzati per l'addestramento del modello sono riportati nella tabella seguente:

IPERPARAMETRO	VALORE	DESCRIZIONE
N_estimators	400	Numero di alberi nella foresta
Max_depth	20	Profondità massima degli alberi
Min_samples_leaf	2	Numero minimo di campioni in una foglia
Random_state	42	Fissa la casualità per garantire la riproducibilità

Min_samples_split	2	Numero minimo di campioni per dividere un nodo
Bootstrap	True	Campionamento con reinserimento per costruire gli alberi

Tabella 4. Iperparametri per addestramento modello

L'addestramento del modello è stato interamente eseguito in Python, suddividendo il dataset in un insieme di training (80%) e uno di test (20%) ed è stato validato tramite la tecnica di cross-validation. Essa consiste nel suddividere il dataset in più parti; il modello viene addestrato su una parte e poi testato sulla rimanente e questo processo viene ripetuto più volte, andando a cambiare di volta in volta i dati usati per l'addestramento e quelli per il test. In questo modo si ottiene una valutazione più affidabile delle prestazioni del modello verificando che esso funzioni bene in diverse condizioni del dataset.

Le metriche utilizzate per la validazione del modello sono:

- **Mean Absolute Error (MAE):** misura l'errore medio tra i valori reali e i valori previsti:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \qquad (3)$$

- **Coefficient of Determination (R^2):** indica la percentuale di varianza spiegata dal modello:

$$R^2 = 1 - \frac{RSS}{TSS} \qquad (4)$$

  Dove: RSS è la devianza residua o somma dei quadrati dei residui, mentre TSS è la devianza totale o somma totale dei quadrati.

In particolare, le metriche ottenute dal modello termico sono:

MAE	R^2
0.22 °C	0.92

Tabella 5. Metriche modello termico

Dunque, come emerge dalla tabella, il modello evidenzia dei risultati molto buoni, con un errore assoluto medio (MAE) pari a 0.22 °C, indicando che la differenza media tra temperatura reale e quella prevista è inferiore a un quarto di grado. Allo stesso tempo il

coefficiente R^2 pari a 0.92 dimostra che il modello è in grado di spiegare oltre il 92% della variabilità della temperatura interna, confermando l'ottima capacità predittiva dell'algoritmo Random Forest.

2.2.2 Modello Comfort

Il secondo componente fondamentale del simulatore dell'edificio è rappresentato dal modello di comfort, il quale, ha il compito di prevedere se all'intervallo temporale successivo, l'ambiente interno si trova in una situazione di comfort termico oppure no.

Nel presente progetto, il comfort è stato definito utilizzando il parametro TSV (Thermal Sensation Vote), raccolto tramite voti soggettivi degli occupanti nel periodo di durata dell'esperimento.

Al fine di ottenere una misura di comfort, utilizzabile nell'algoritmo RL, il TSV è stato trasformato in una variabile binaria. In particolar modo, una situazione è stata definita come situazione di "comfort" se $-0.5 \leq TSV \leq +0.5$, mentre tutte le altre condizioni vengono considerate come "discomfort". Dunque, a differenza del modello termico, che forniva in output una variabile continua, il modello di comfort fornisce una variabile binaria, andando a indicare semplicemente se all'istante successivo mi troverò in una condizione di comfort o discomfort.

Per la stima del comfort termico è stato utilizzato il Random Forest Classifier, dal momento che bisogna gestire un problema di classificazione.

Il modello è stato addestrato interamente in python, con una serie di variabili che descrivono sia le condizioni interne che quelle esterne. Poiché la percezione termica non dipende solo dal valore istantaneo della temperatura, ma anche dalla sua evoluzione nel tempo, sono state incluse nel modello anche le variazioni di temperatura interna ed esterna, che risultano particolarmente informative.

La tabella seguente riassume le input features e la variabile target del modello di comfort:

TIPO	VARIABILE	NOME NEL DATASET	UNITÀ DI MSIURA
Input	Temperatura interna	T_in	°C
Input	Variazione temperatura interna	dTin	°C

Input	Temperatura esterna	T_out	°C
Input	Variazione temperatura esterna	dTout	°C
Input	Umidità relativa interna	RH_in	%
Input	Umidità esterna	RH_out	%
Input	Radiazione solare	Global_rad	W/m^2
Input	CO_2 interna	CO_2_in	ppm
Input	Stato impianto	Rf_on_any	ON/OFF
Input	Ora del giorno	Hour	h
Label	Comfort termico futuro	TSV_next	Scala (da+3 a -3)

Tabella 6. Feature e label del modello di comfort

Anche in questo caso, il modello è stato addestrato suddividendo il dataset iniziale in training (80%) e test (20%), gli iperparametri sono stati impostati ugualmente al modello termico (tabella 4.) ed è stato successivamente valutato con diverse metriche utilizzate quando si ha a che fare con Random Forest classifier:

- **Accuracy**: rappresenta la frazione di previsioni corrette rispetto al totale delle osservazioni. Indica quindi quanto spesso il modello indovina la classe corretta.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (5)$$

  Dove: TP è True Positives, TN è True Negatives, FP è False Positives e FN è False Negatives.

- **F1-score**: è la media armonica tra precisione e recall, e misura l'equilibrio tra la capacità del modello di evitare falsi positivi (precision) e evitare falsi negativi (recall).

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision+recall} \qquad (6)$$

I risultati ottenuti in seguito all'addestramento del modello sono riportati nella tabella seguente.

ACCURACY	F1-SCORE
0.88	0.93

Tabella 7. Metriche modello comfort

L’accuracy pari a 0.88 è un risultato notevole, e indica che il modello riesce a classificare correttamente l’88% degli stati totali, o in altre parole, circa nove volte su dieci, riesce a stabilire se l’edificio si trova in una condizione di comfort o discomfort. Il valore dell’F1-score pari a 0.93 conferma che il modello è in grado di riconoscere gli stati in comfort in modo affidabile e coerente. Questa affidabilità è necessaria per garantire che l’algoritmo RL apprenda una politica corretta e basata su informazione veritiere, evitando di penalizzare o premiare azioni sulla base di previsioni errate.

Le metriche ottenute mostrano che il modello è in grado di rappresentare in modo adeguato la relazione tra le variabili ambientali e la percezione termica degli occupanti, risultando sufficientemente accurato per essere integrato nel simulatore e utilizzato nelle fasi di addestramento dell’agente RL.

In conclusione, il modello di comfort descritto riceve la temperatura prevista dal modello termico, le condizioni climatiche esterne e lo stato dall’impianto e restituisce un valore binario che indica se sono in una condizione di comfort oppure no. Questo valore risulterà utile per definire la ricompensa che l’agente riceve dall’ambiente in base all’azione intrapresa.

2.2.3 Modello energetico

Il terzo e ultimo componente del simulatore dell’edificio è il modello energetico, il quale si occupa della previsione dei consumi associati all’impianto di riscaldamento.

A differenza dei due precedenti modelli che si basavano su un gruppo numerosi di variabili, il modello energetico utilizza poche feature; infatti, esso sfrutta semplicemente lo stato dell’impianto, gli HDD, e i dati energetici mensili per ottenere una misura real-time dei consumi energetici. Le bollette utilizzate coprono il periodo compreso tra gennaio 2024 e maggio 2025, fornendo una base energetica reale su cui costruire la stima dei consumi. Poiché le bollette forniscono un consumo mensile complessivo, è stato necessario ricavare da questo valore una stima del consumo giornaliero e successivamente orario. Il procedimento si basa sugli HDD: il consumo del mese viene distribuito sui singoli giorni in proporzione ai gradi giorno di ciascun giorno. In pratica, i giorni più freddi — cioè quelli con HDD più elevati — ricevono una quota maggiore del consumo mensile, mentre i giorni più miti ne ricevono una quota minore. Lo stesso

principio viene applicato per ottenere il consumo orario: il consumo giornaliero viene suddiviso tra le ore della giornata in base agli HDD calcolati ora per ora, assegnando più consumo alle ore più fredde e meno consumo alle ore più calde. In questo modo si ottiene una ricostruzione oraria dei consumi coerente con il fabbisogno termico reale dell'edificio. Le feature utilizzate dal modello energetico sono visualizzabili nella tabella sottostante:

TIPO	VARIABILE	NOME NEL DATASET	UNITÀ DI MISURA
Input	Stato dell'impianto	Rf_on_any	ON/OFF
Input	Heating Degree Days	HDD	
Input	Consumo mensile bolletta	Monthly_kwh	kWh
Input	Ora del giorno	hour	h
Label	Consumo energetico previsto	Kwh_next	kWh

Tabella 8. Feature e label modello energetico

La feature più rilevante tra queste è sicuramente lo stato dell'impianto. Infatti, esso fa sì che il consumo energetico risulti completamente dipendente dalle azioni RL e non dalle condizioni climatiche iniziali. Con questa scelta, si ottiene un grafico dei consumi valido, in cui ci sono picchi di consumo perfettamente sincronizzati con le accensioni simulate.

Questo modello è stato sviluppato con una Random Forest Regressor, implementata in python ed è stato validato con le stesse metriche del modello termico (3), (4), producendo i seguenti risultati.

MAE	R^2
1.38	0.55

Tabella 9. Metriche modello energetico

Sebbene esso non sia in grado di riprodurre perfettamente la variabilità dei consumi reali, esso riesce a svolgere il compito richiesto dal simulatore: prevedere dei consumi coerenti con le azioni simulate dall'agente RL.

2.3 Reinforcement Learning

Il Reinforcement Learning è una tecnica di apprendimento basata sulla ricompensa, utilizzata per risolvere problemi decisionali complessi. L'agente RL compie interazioni autonome con l'ambiente in cui opera tramite un meccanismo di tentativi ed errori, riceve feedback sotto forma di ricompense e penalità, e utilizza tali informazioni per migliorare la propria strategia di controllo [1].

A differenza degli apprendimenti supervisionati, il RL si distingue per il suo carattere iterativo, in cui un agente impara attraverso un sistema di ricompense e punizioni. L'obiettivo finale sarà quello di apprendere una policy ottimale, ovvero la strategia che massimizza la ricompensa.

La struttura concettuale del RL è composta da questi termini chiave:

- **Ambiente**: rappresenta il contesto fisico o virtuale in cui l'agente opera. In questo caso l'ambiente è il simulatore data-driven dell'edificio, che ha il compito di fornire all'agente una rappresentazione coerente dell'edificio, su cui imparare una strategia di controllo.
- **Agente**: colui che prende le decisioni. Nel nostro caso è l'algoritmo di Q-learning, che decide se accendere o spegnere l'impianto.
- **Stato**: condizione specifica dell'ambiente in un dato istante. Ha lo scopo di fornire all'agente tutte le informazioni necessarie per valutare la situazione corrente e scegliere l'azione più appropriata.
- **Azione**: decisione presa dall'agente. In questo progetto l'azione è binaria: ON oppure OFF.
- **Ricompensa**: feedback che l'agente riceve dall'ambiente in seguito ad un'azione. Ha lo scopo di guidare l'apprendimento dell'agente.

Il funzionamento di un qualsiasi algoritmo di RL è descritto dalla figura sottostante:

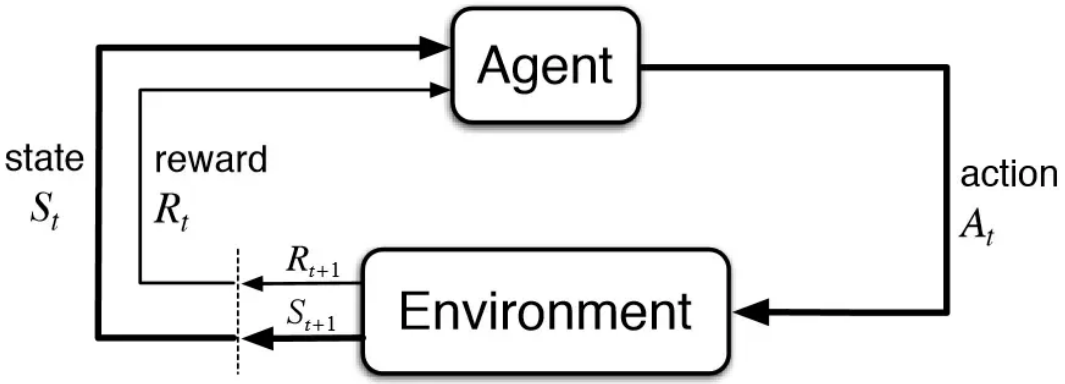


Figura 5. Schema processo RL

Ad ogni passo temporale, l'agente inizialmente osserva lo stato corrente, poi sceglie un'azione e successivamente riceve dall'ambiente una ricompensa in base alla qualità dell'azione intrapresa. L'ambiente a sua volta, a seguito dell'azione, evolve nello stato successivo. Questo processo viene ripetuto per molte volte, permettendo con il tempo di raggiungere una politica di controllo ottimale.

Un aspetto distintivo del RL sta nel compromesso tra esplorazione e sfruttamento:

- **Esplorazione**: durante l'esplorazione l'agente prova azioni diverse per ampliare la propria conoscenza e raccogliere nuove informazioni sull'ambiente. L'obiettivo è quello di migliorare la strategia nel lungo termine, cioè scoprire potenzialmente azioni che possono portare a ricompense più elevate in futuro.
- **Sfruttamento**: rappresenta la tendenza a usare azioni già note come vantaggiose. L'agente tenderà a scegliere le azioni che hanno portato i migliori risultati in passato.

Il compromesso tra questi due elementi è la chiave del RL. Infatti, se l'agente esplora troppo, rischia di non ottenere ricompense nell'immediato; se invece sfrutta troppo rischia di rimanere intrappolato in strategie subottimali.

In questa tesi, il RL viene sfruttato per apprendere una strategia di controllo di un impianto di riscaldamento, in grado di mantenere il comfort e minimizzare i consumi energetici. L'agente non riceve istruzione esplicite su quando accendere o spegnere l'impianto, ma, grazie al Reinforcement Learning, riesce ad apprendere una strategia di gestione basata sull'esperienza, sfruttando il simulatore creato nei paragrafi precedenti. In questo modo

l’agente può sperimentare molti episodi, valutare le azioni migliori e convergere verso una politica di controllo stabile ed efficacie.

Nei prossimi paragrafi vengono definiti l’ambiente, lo stato, le azioni, la ricompensa, relative a questo caso studio, e infine l’algoritmo di Q-learning utilizzato per l’addestramento dell’agente.

2.3.1 Agente RL

Nel Reinforcement Learning, l’agente è l’entità decisionale che interagisce con l’ambiente, osserva lo stato corrente, seleziona un’azione e aggiorna la propria strategia sulla base della ricompensa ricevuta. Il suo compito è apprendere autonomamente una politica di controllo che massimizzi la ricompensa nel lungo periodo. Nel settore del controllo termo-energetico degli edifici, l’agente RL viene utilizzato per prendere decisioni che bilanciano comfort e consumo energetico, adattandosi dinamicamente alle condizioni del sistema [1], [3], [16], [17].

Nel presente lavoro, l’agente è implementato tramite l’algoritmo di **Q-learning**, che apprende una politica di gestione dell’impianto di riscaldamento scegliendo tra due azioni possibili: accendere o spegnere il sistema. L’agente non riceve istruzioni esplicite su quando intervenire, ma costruisce la propria strategia attraverso l’esperienza accumulata nel simulatore, valutando gli effetti delle azioni sul comfort e sul consumo energetico [18].

Grazie a questo processo iterativo, l’agente è in grado di convergere verso una politica stabile ed efficace, che riflette il compromesso ottimale tra comfort degli occupanti ed efficienza energetica.

2.3.2 Ambiente di Reinforcement Learning

Nel RL, l’ambiente è il contesto in cui si muove un agente. È all’interno dell’ambiente che si svolge il ciclo di osservazione, decisione e feedback. La prima funzione principale svolta dall’ambiente è fornire all’agente lo stato, ossia comunicare ad esso la condizione attuale in cui si trova. Successivamente riceve l’azione scelta dall’agente e cambia lo stato in base all’azione stessa. Come ultima funzione va a fornire la ricompensa (reward), cioè

invia all'agente un valore numerico che indica se l'azione è stata utile o meno al raggiungimento dell'obiettivo.

La costruzione di un ambiente incide sulla qualità dell'apprendimento. Infatti, se l'ambiente fosse troppo semplificato potrebbe produrre politiche inefficaci, d'altra parte se troppo complesso può rendere l'apprendimento troppo lento.

Un altro aspetto importante dell'ambiente RL è la sua natura simulativa. Esso permette all'agente di non interagire con l'edificio reale ma con una sua ricostruzione, cosicché l'agente è libero di esplorare nelle prime fasi di apprendimento senza il rischio di andare incontro a sprechi energetici o situazioni di discomfort per gli occupanti. Soltanto una volta trovata la politica ottimale, essa potrà essere applicata all'edificio reale.

Nel presente lavoro, sono stati utilizzati i tre modelli ML (termico, comfort, energetico) per costruire l'ambiente di RL. Questi modelli garantiscono una ricostruzione del comportamento dell'edificio, e in particolar modo, permettono di capire come evolve la temperatura, il comfort e il consumo energetico all'interno dell'edificio. Ad ogni passo temporale l'ambiente riceve l'azione (accendere o spegnere il riscaldamento) scelta dall'agente in seguito all'osservazione dello stato, successivamente va ad assegnare una ricompensa in base alla bontà dell'azione intrapresa, e infine utilizza i modelli predittivi per calcolare il nuovo stato, che verrà osservato dall'agente nel passo temporale successivo.

Nei paragrafi successivi verranno trattati gli elementi che compongono l'ambiente: lo stato, le azioni, e la ricompensa.

2.3.3 Stato

Nel Reinforcement Learning, lo stato è la descrizione della situazione in cui si trova l'ambiente in un dato momento. Lo stato deve contenere tutti gli elementi necessari che permettano di avere una rappresentazione completa del sistema, affinché l'agente possa avere una visione totale della situazione corrente e possa scegliere l'azione più appropriata.

Lo stato deve contenere esclusivamente solo quelle variabili che influenzano la dinamica dell'edificio e la ricompensa. Variabili aggiuntive ma non necessarie non fanno altro che

aumentare la complessità dello spazio degli stati, rendendo più difficile il raggiungimento della politica ottimale.

Nel contesto di questa tesi, lo stato contiene al suo interno la temperatura prevista dal modello termico, il comfort previsto al passo successivo, il consumo energetico stimato e lo stato dell'impianto. Queste informazioni rappresentano la base su cui l'agente costruisce le proprie decisioni.

2.3.4 Azioni

Nel Reinforcement Learning, un'azione è una decisione presa dall'agente. Ogni azione intrapresa va a cambiare lo stato dell'ambiente e a generare una ricompensa.

In generale, le azioni devono essere scelte in modo da rappresentare le possibilità di intervento realmente disponibili nel sistema che si vuole controllare.

In questo lavoro, le azioni sono legate al funzionamento dell'impianto di riscaldamento. L'agente, infatti, può scegliere tra due azioni possibili: spegnere o accendere l'impianto. La scelta di utilizzare questo set binario di azioni deriva dalla natura reale del sistema, in cui l'impianto radiante non dispone di livelli di modulazione o potenze intermedie, ma può lavorare solo in modalità acceso/spento.

L'utilizzo di un set binario di azioni semplifica l'apprendimento, riducendo la complessità dello spazio delle azioni e permettendo all'agente di esplorare rapidamente l'effetto di ciascuna azione. Inoltre, un secondo vantaggio è che facilita la comprensione della relazione tra azione e ricompensa, da parte dell'agente.

2.3.5 Reward

Nel Reinforcement Learning, la ricompensa è un valore numerico inviato dall'ambiente all'agente in seguito ad ogni sua azione intrapresa. Essa indica se l'azione fatta è "buona" o "cattiva" rispetto all'obiettivo. Azioni che generano reward positive vengono rafforzate, mentre azioni che generano reward negative rappresentano un errore e vengono progressivamente evitate.

Una buona funzione di reward deve essere coerente con gli obiettivi del problema, deve fornire un feedback chiaro e guidare l'agente verso la policy ottimale. È consigliato

utilizzare reward non troppo complicate perché potrebbero rendere l'apprendimento instabile o portare l'agente a comportamenti inattesi.

Nel presente lavoro, la reward è stata creata per bilanciare i due obiettivi principali: mantenere il comfort per gli occupanti e ridurre il consumo energetico. Quando l'edificio si trova nella zona di comfort termico (-0.5 < TSV < +0.5), l'agente riceve una reward positiva; quando esce dalla zona di comfort riceve una penalizzazione. In python questa componente è implementata tramite una variabile *comfort_term*, che assume valore:

- +1.0 se il comfort è rispettato,
- -1.0 se il comfort non è rispettato

D'altra parte, il consumo dell'impianto viene penalizzato in modo proporzionale, evitando accensioni inutili o eccessive.

Inoltre, per evitare comportamenti oscillatori (accensioni o spegnimenti troppo frequenti), è stata introdotta una penalità aggiuntiva (0.1) quando l'agente cambia azione rispetto al passo precedente. Questa penalità riflette il comportamento reale di un impianto radiante, che non dovrebbe essere modulato troppo frequentemente.

Matematicamente la reward implementata in python può essere rappresentata da questa formula:

$$reward = w_comfort \cdot comfort_term - w_energy \cdot kwh_next - penalty_switch \qquad (7)$$

Dove w_comfort e w_energy sono rispettivamente i pesi del comfort e dell'energia nella reward, e è stato scelto il primo maggiore del secondo per conferire al comfort un ruolo predominante.

In sintesi, la funzione di reward rappresenta il meccanismo attraverso il quale l'agente impara a bilanciare comfort e consumo energetico. Attraverso la reward, l'agente impara che mantenere il comfort è desiderabile, ma che farlo in modo inefficiente comporta un costo.

2.4 Algoritmo di Q-learning

Il Q-learning è un popolare algoritmo di apprendimento per rinforzo senza modello (model-free) [1], [3]. Permette all'agente di imparare a scegliere l'azione migliore,

soltanto interagendo con l'ambiente, attraverso un meccanismo di tentativi ed errori, senza il bisogno di conoscere preventivamente le regole del sistema.

L'agente, man mano che compie azioni, aggiorna progressivamente una tabella di valori chiamata Q-table. La Q-table è l'organo centrale del Q-learning, essa è una tabella che contiene un "valore" per ogni coppia stato-azione, e tale valore indica quanto sia conveniente compiere una certa azione in un dato stato, in termini di ricompense future attese. La Q-table viene continuamente aggiornata durante la fase di addestramento, in questo modo, l'agente avrà una stima sempre più affidabile della politica ottimale, cioè delle azioni che possono massimizzare la reward.

I modelli Q-learning seguono un processo iterativo di molti episodi, in cui i diversi componenti lavorano insieme per addestrare l'agente. Il ciclo può essere riassunto da questo schema:

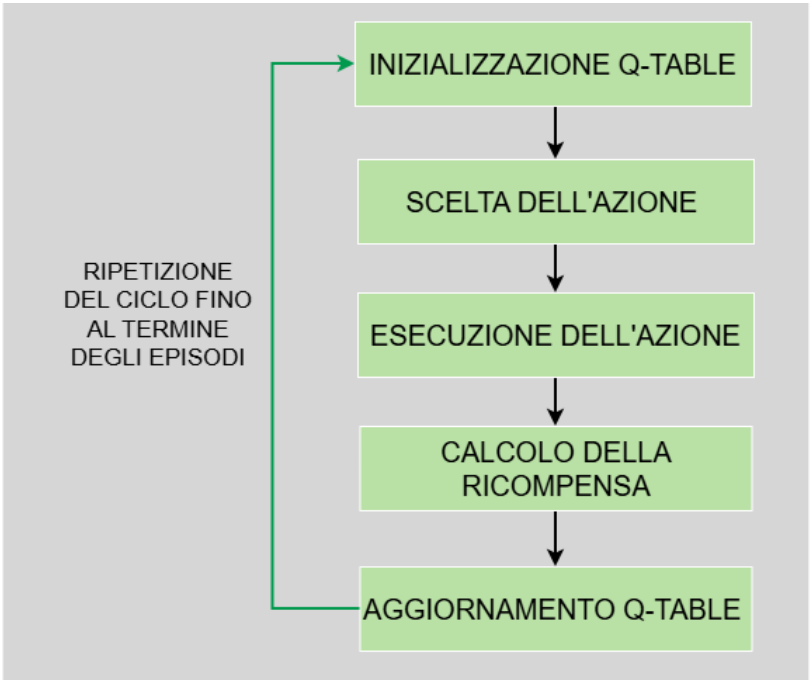


Figura 6. Schema processo Q-learning

Osservando la figura 6. il processo svolto dall'algoritmo Q-learning può essere scomposto in questa sequenza di fasi:

1. **Osservazione dello stato iniziale (S)**

L’ambiente fornisce all’agente uno stato iniziale (S) che descrive la situazione corrente.

2. **Selezione dell’azione (A)**
   In base allo stato attuale, l’agente sceglie un’azione sulla base della Q-table.
3. **Esecuzione dell’azione e risposta dell’ambiente**
   L’agente esegue l’azione selezionata e l’ambiente fornisce un nuovo stato (S’)e una ricompensa (R).
4. **Aggiornamento Q-table**
   L’agente aggiorna la tabella Q regolando il valore della coppia stato-azione in base alla ricompensa ricevuta e al nuovo stato.
5. **Ripetizione del ciclo**
   Il ciclo si ripete per molti episodi, permettendo all’agente di apprendere la politica ottimale.

In questa tesi si è sfruttato un algoritmo di Q-learning per addestrare un agente in grado di controllare l’impianto di riscaldamento dell’edificio in questione. Grazie alla natura model-free dell’algoritmo, l’agente non ha bisogno di conoscere la dinamica termica dell’edificio, ma impara direttamente dai modelli predittivi che compongono l’ambiente.

Nei paragrafi seguenti verranno descritti la discretizzazione dello stato, la realizzazione della Q-table, gli iperparametri del Q-learning e la procedura di training.

2.4.1 Q-table e discretizzazione

La Q-table è una matrice utilizzata nell’apprendimento per rinforzo tramite l’algoritmo Q-learning.

Essa contiene per ogni coppia stato-azione, una stima del valore Q che indica quanto quell’azione sia vantaggiosa in quello stato, in base a un sistema di ricompense e penalità.

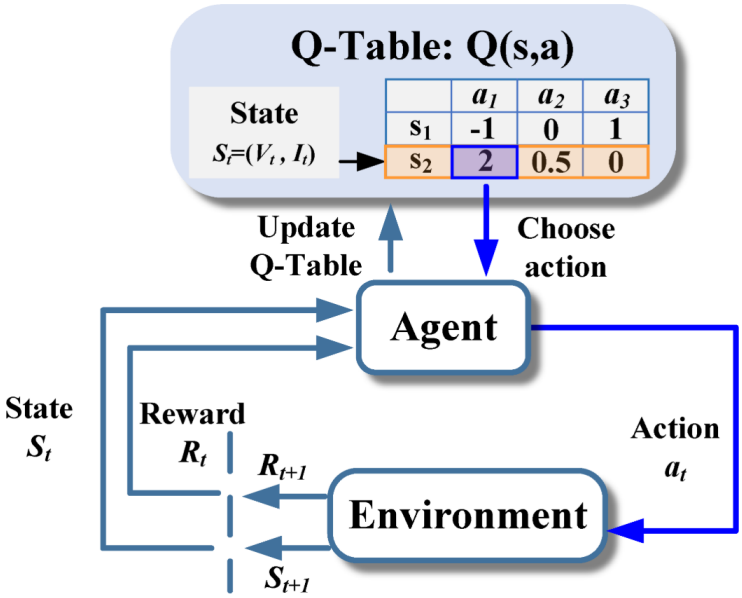


Figura 7. Q-table

Come emerge dalla figura 7, le righe della tabella corrispondono agli stati possibili dell'ambiente, mentre le colonne alle azioni possibili da parte dell'agente. I valori interni (valori Q) indicano la convenienza di tale azione in quello stato. Questi valori verranno aggiornati progressivamente durante l'apprendimento, in base alle ricompense ricevute. Man mano che si prosegue con l'interazione con l'ambiente, la Q-table sarà sempre più ricca di informazioni utili: i valori Q, inizialmente nulli, verranno modificati continuamente, rendendo la tabella una rappresentazione sempre più accurata della politica ottimale.

Per costruire questa tabella è necessaria la discretizzazione dello spazio degli stati: il Q-learning non può lavorare con variabili continue, poiché in questo caso ci sarebbe bisogno di una tabella infinita. Di conseguenza, in questo lavoro, lo stato continuo prodotto dall'ambiente è stato discretizzato, in modo da trasformarlo in un indice intero, utilizzabile nella Q-table.

Le variabili che fanno parte dello stato continuo prodotto dall'ambiente sono cinque: temperatura interna (T_in, °C), temperatura esterna (T_out, °C), ora del giorno (hour, h), giorno della settimana e lo stato dell'impianto (rf_on_any, ON/OFF). Ciascuna variabile è stata scomposta in un numero limitato di intervalli. Di seguito in Figura 8 viene riportato l'esempio della discretizzazione della temperatura interna:

```python
def discretize_tin(tin):
    if tin < 20.5: return 0
    elif tin < 21.0: return 1
    elif tin < 21.5: return 2
    elif tin < 22.0: return 3
    elif tin < 22.5: return 4
    else: return 5
```

Figura 8. Discretizzazione T interna

Come appare nella figura 8, la temperatura interna è stata suddivisa in sei range, e in base al range di appartenenza viene assegnato un valore intero da 0 a 5. La discretizzazione è stata fatta anche per le altre quattro variabili ed è riassunta nella tabella di seguito:

VARIABILE	DISCRETIZZAZIONE	VALORI POSSIBILI
Temperatura interna (T_in, °C)	Suddivisa in 6 range (Fig.7)	0-5
Temperatura esterna (T_out, °C)	In 3 range: T_out < 10°C T_out < 20°C T_out > 20°C	0-2
Ora del giorno (hour)	Intervalli di 4 ore	0-5
Giorno della settimana (weekday)	2 categorie: feriale/festivo	0-1
Stato dell'impianto (rf_on_any)	ON/OFF	0-1

Tabella 10. Discretizzazione delle variabili

Dato che ogni variabile dello stato è stata discretizzata in un numero finito di categorie, il numero totale di stati dell'ambiente RL è dato dal prodotto delle categorie di ciascuna variabile:

$$N_{stati} = N_{Tin} \cdot N_{Tout} \cdot N_{Hour} \cdot N_{Weekday} \cdot N_{RF} \qquad (8)$$

Ora sapendo che: $N_{Tin} = 6$, $N_{Tout} = 3$, $N_{Hour} = 6$, $N_{Weekday} = 2$, $N_{RF} = 2$, il numero totale di stati utilizzati nella Q-table sarà:

$$N_STATES = 6x3x6x2x2 = 432$$

In questo modo, ogni possibile combinazione delle variabili discrete corrisponde a un numero tra 0 e 432.

La Q-table quindi avrà una dimensione di 432 x 2, dove le due colonne sono le due azioni possibili (accensione/spegnimento). Inizialmente essa, sarà composta da tutti zeri, dato che la conoscenza iniziale è nulla, mentre, man mano che l'agente compie azioni, essa verrà aggiornata con la formula classica di aggiornamento del Q-learning, basata sull'equazione di Bellmann [91], riportata qui sotto:

$$Q(s, a) = Q(s, a) + \alpha \cdot [\, r + \gamma \cdot max\, Q(s', a') - Q(s, a)\,] \qquad (9)$$

Il primo termine rappresenta il valore attuale della coppia stato-azione, α è il tasso di apprendimento, R è la ricompensa immediata dopo aver svolto l'azione a, mentre il termine all'interno della parentesi quadra indica l'errore di differenza temporale, cioè la differenza tra il valore stimato e il guadagno reale atteso. Questa formula permette all'agente di aggiornare la stima del valore Q, combinando la ricompensa immediata con la migliore ricompensa futura attesa.

Nel paragrafo successivo verranno descritti gli iperparametri che guidano l'apprendimento.

2.4.2 Parametri del Q-learning

La scelta dei parametri del Q-learning è un passo cruciale nella creazione di un algoritmo di successo. Infatti, è proprio dal tuning (ottimizzazione) di questi valori che dipende la velocità e la qualità di apprendimento dell'agente e sono proprio i parametri a determinare come l'agente aggiorna i valori Q nel tempo. La tabella seguente mostra i parametri e i relativi valori impostati nel codice Python; ciascun elemento sarà descritto dettagliatamente nei paragrafi successivi.

PARAMETRO	VALORE UTILIZZATO	SIGNIFICATO
Learning rate (α)	0.1	Peso aggiornamento della Q-table
Discount factor (γ)	0.95	Peso delle ricompense future
Epsilon iniziale (ε)	0.5	Probabilità iniziale di esplorazione
Epsilon finale (ε)	0.05	Probabilità minima di esplorazione

Numero di episodi	300	Numero di cicli completi di apprendimento

Tabella 11. Parametri del Q-learning

Il primo parametro rilevante è il **tasso di apprendimento α.** Esso determina quanta importanza dare alle nuove informazioni rispetto alle vecchie. Può assumere valori tra 0 e 1: un valore troppo alto (vicino a 1) renderebbe l'apprendimento instabile poiché l'agente utilizzerebbe solo l'ultima esperienza, dimenticando il passato. Al contrario, un valore troppo basso renderebbe l'apprendimento troppo lento e l'agente rischia di non imparare nulla. Nel nostro contesto, ho scelto di impostare $\alpha = 0.1$, essendo uno dei valori standard più utilizzati nel Q-learning, dal momento che garantisce aggiornamenti abbastanza rapidi, assicurando stabilità al sistema.

Il secondo parametro fondamentale è il **fattore di sconto γ**. Esso determina l'importanza delle ricompense future rispetto a quelle immediate. Questo parametro è particolarmente rilevante nei sistemi termici, dove le azioni dell'agente non producono effetti istantanei. Infatti, l'accensione dell'impianto non modifica subito la temperatura interna, dal momento che l'edificio ha una certa inerzia termica, e quindi la temperatura e il comfort cambiano solo dopo un certo intervallo di tempo. Per questo è importante che l'agente consideri non solo la ricompensa immediata ma anche quelle future. Anche questo parametro varia da 0 a 1, e nel contesto di questa tesi è stato scelto di impostare $\gamma = 0.95$ che rappresenta un agente "lungimirante" che pianifica le sue azioni guardando molto avanti nel tempo.

Il terzo parametro di fondamentale importanza è il **tasso di esplorazione ε**. Esso regola la scelta tra provare nuove azioni (esplorazione) o usare quella già note (sfruttamento). La strategia più comune nel RL per risolvere il problema tra esplorazione e sfruttamento è il metodo *epsilon-greedy* con decadimento [24]: ad ogni passo temporale, l'algoritmo genera un numero casuale tra 0 e 1 e lo confronta con il valore di ε. A questo punto con probabilità $1 – \varepsilon$ l'agente si comporta in modo "ingordo" (greedy) e sceglierà l'azione con il valore Q più alto per lo stato attuale. Con probabilità ε, invece, l'agente ignora completamente le azioni già fatte e la sua esperienza, e sceglierà un'azione completamente casuale tra quelle disponibili. Ad esempio, se impostassi $\varepsilon = 0.2$, significa

che l'agente userà la migliore strategia nell'80% dei casi, mentre nei restanti 20% sceglierà un'azione completamente casuale.

Tuttavia, mantenere ε fisso per tutto l'addestramento non è efficiente dal momento che, all'inizio l'agente non sa nulla dell'ambiente e deve quindi esplorare molto per migliorare la sua conoscenza. Con il procedere degli episodi, però, l'agente dovrebbe applicare solo la politica appresa e sfruttare le azioni che ha identificato come vantaggiose. Per questo motivo si introduce il concetto di decadimento: esso consiste nel partire da un valore di ε abbastanza alto e lo si riduce gradualmente a ogni episodio o a ogni passo, moltiplicandolo per un fattore di decadimento fino a raggiungere un valore minimo. In particolare, in questo progetto è stato implementato un decadimento lineare di ε, che parte da un valore di 0.5 e scende fino a un minimo di 0.05. Questa strategia permette all'agente di esplorare ampiamente nelle prime fasi di addestramento e riducendo progressivamente l'esplorazione con l'aumentare degli episodi.

Infine, l'ultimo parametro necessario al Q-learning è il numero di episodi. In questo lavoro sono stati utilizzati 300 episodi, un valore che permette all'agente di esplorare l'intero dataset più volte senza tempi di calcolo eccessivi. Studi recenti mostrano come un numero di episodi compreso tra 200 e 500 è comunemente adottato per il controllo HVAC tramite RL, garantendo l'apprendimento di una politica stabile [16],[17].

Una volta definiti i parametri necessari all'algoritmo, il prossimo passo consiste nel descrivere la procedura di training adottata.

2.4.3 Procedura di training

Una volta definita la struttura della Q-table e il valore di impostazione dei parametri dell'algoritmo, si può procedere con la descrizione della procedura di training, tramite la quale l'agente impara la politica di controllo.

L'addestramento si basa su un ciclo iterativo di 300 episodi, nei quali, l'agente interagisce con l'ambiente simulato, osservando gli stati, selezionando le azioni, ricevendo ricompense e aggiornando di volta in volta i valori della Q-table. Questo procedimento serve all'agente per trasformare una tabella inizialmente vuota e piena di zeri, in una rappresentazione sempre più accurata della convenienza delle azioni nei vari stati.

La procedura di addestramento inizia con la reimpostazione dell'ambiente tramite la funzione *reset(),* che fornisce lo stato iniziale dell'edificio. Questo stato, continuo, viene subito discretizzato, in modo da essere idoneo per accedere alla Q-table. Successivamente l'agente inizierà a scegliere le azioni in base alla politica ε-greedy, descritta nei paragrafi precedenti: con probabilità ε, l'agente sceglierà un'azione completamente casuale, mentre con probabilità 1-ε sceglierà l'azione che avrà il valore Q più elevato per quel preciso stato.

Questa logica permette all'agente di esplorare nelle prime fasi dell'addestramento, in cui ha bisogno di ampliare la propria conoscenza e capire quali azioni sono migliori, riducendo gradualmente l'esplorazione man mano che si va avanti con gli episodi. Infatti, è stata adottata la logica del decadimento lineare di ε, che porta l'agente a privilegiare lo sfruttamento nelle fasi finali del training.

Terminata la fase di selezione dell'azione tramite la politica ε-greedy, l'agente esegue l'azione nell'ambiente che, a sua volta, restituirà il nuovo stato, la ricompensa e un flag che indica se l'episodio è finito. A questo punto, il nuovo stato viene nuovamente discretizzato e la ricompensa, invece, verrà utilizzata per aggiornare i valori contenuti nella Q-table, tramite la formula di aggiornamento del Q-learning.

In questo modo, la Q-table diventerà sempre più ricca e informativa passo dopo passo. Infatti, episodio dopo episodio, l'agente accumula esperienza e migliora la propria politica, imparando quali azioni sono più vantaggiose nei diversi stati dell'ambiente.

In questo lavoro, il ciclo di training viene fatto proseguire per 300 episodi. Al termine dell'addestramento, vien salvata la Q-table finale, la quale verrà poi utilizzata per applicare la politica appresa, selezionando sempre l'azione con il valore Q più elevato.

Nei paragrafi successivi verrà analizzata la qualità della politica appresa e i risultati ottenuti.

2.5 Esecuzione della politica RL

Una volta terminata la fase di addestramento e ottenuta la Q-table finale, il passaggio successivo consiste nell'esecuzione della politica RL.

In questa fase, l'agente smette di esplorare l'ambiente, non sceglie più le azioni in base alla politica ε-greedy, e non aggiorna più i valori Q. Infatti, a questo punto inizia la parte di sfruttamento (vedere sezione 2.3), in cui, l'agente va ad applicare la politica appresa dopo i 300 episodi di training, scegliendo in ogni stato l'azione con il valore Q più alto all'interno della Q-table, cioè quella più conveniente.

Dal punto di vista operativo, l'esecuzione della politica RL può essere descritta attraverso i seguenti passaggi:

- **Inizializzazione dell'ambiente:** L'ambiente viene inizializzato tramite la funzione *reset(),* che restituisce il primo stato continuo dell'edificio.
- **Discretizzazione dello stato:** lo stato continuo viene immediatamente trasformato nel corrispondente stato discreto, necessario per utilizzare la Q-table.
- **Selezione dell'azione: l**'agente sceglie l'azione da eseguire selezionando sempre l'azione con il valore Q più alto associato allo stato corrente.
- **Esecuzione dell'azione:** l'zione scelta viene eseguita nell'ambiente, che restituisce: il nuovo stato, la ricompensa, e le variabili simulate (temperatra interna, comfort, consumo energetico, azione applicata)
- **Memorizzazione dei risultati:** Tutti i valori ottenuti vengono salvati in apposite liste, che permettono alla fine di poter rappresentare l'intera simulazione dell'agente tramite dei grafici
- **Avanzamento temporale:** Il processo viene ripetuto per ogni passo temporale fino al termine del dataset, momento in cui l'ambiente segnala la fine dell'episodio.

Alla fine dell'esecuzione della politica RL, l'algoritmo produce una serie di risultati che rappresentano il comportamento dell'agente e l'evoluzione dell'edificio durante l'intera simulazione. In particolare, l'output finale comprende: la sequenza delle azioni fatte dall'agente e la sequenza degli stati dell'edificio (temperatura interna, TSV, consumo energetico).

Questa fase è fondamentale per valutare in modo diretto la politica appresa. Infatti, successivamente, l'intera simulazione dell'agente viene confrontata con i dati reali dell'edificio: temperatura interna, comfort reale, consumi energetici, azioni di controllo

effettivamente applicate. Tale confronto, permette di verificare l'efficacia della politica RL, e quindi, capire se essa è in grado di rappresentare una strategia di controllo migliore rispetto alle logiche tradizionali adottate nell'edificio reale.

2.6 Caso Studio

Il caso studio trattato in questa tesi riguarda un edificio situato a Boecillo (Valladolid), in Spagna. Questo edificio è stato soggetto ad una campagna di misura svolta nel periodo compreso tra il 6 maggio 2025 e il 16 maggio 2025, per una durata complessiva di dieci giorni. L'edificio è adibito ad uso ufficio e presenta un impianto di riscaldamento a pannelli radianti, il quale è attivo per tutti i giorni da mezzanotte fino alle undici del mattino. Nella figura di seguito è presente la mappa dell'ambiente in cui è stata svolta la campagna di misura:

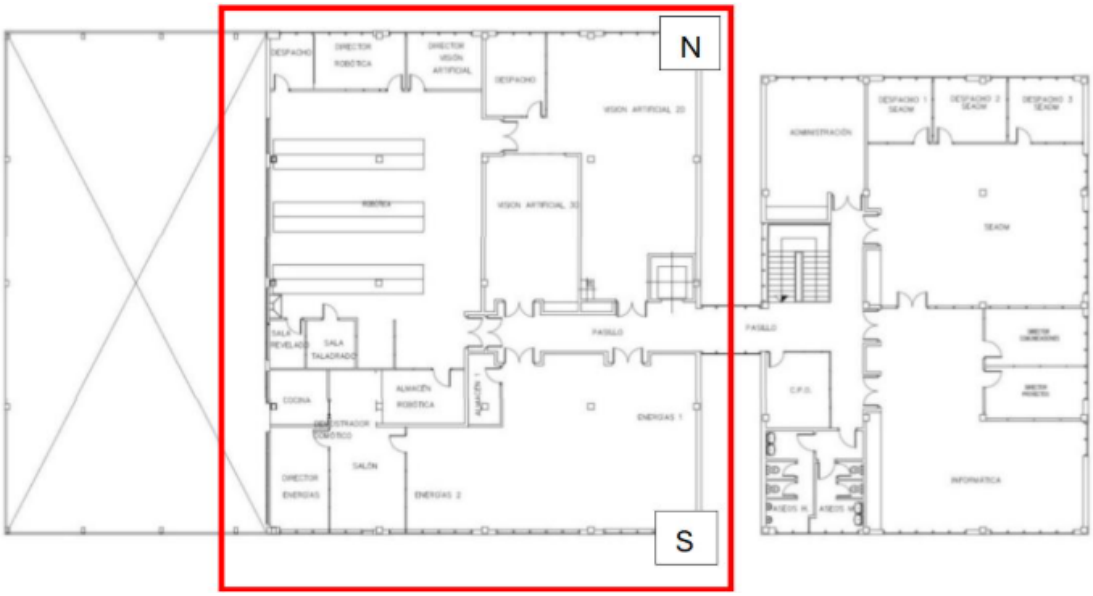


Figura 9. Mappa dell'ambiente di misura

Durante il periodo di durata dell'esperimento, l'edificio è stato continuamente monitorato, con acquisizione dei dati ad intervalli regolari di un'ora, come descritto nella sezione 2.1 dedicata al dataset reale. In totale hanno partecipato alla campagna di misura 18 partecipanti, di diverso genere, di diversa età e diversa morfologia, ai quali è stato chiesto ad ogni ora di compilare dei questionari online in cui esprimevano il loro voto di sensazione termica in quel preciso istante, in termini di TSV, su una scala che va da -3 a +3, come riportato nella figura di seguito:

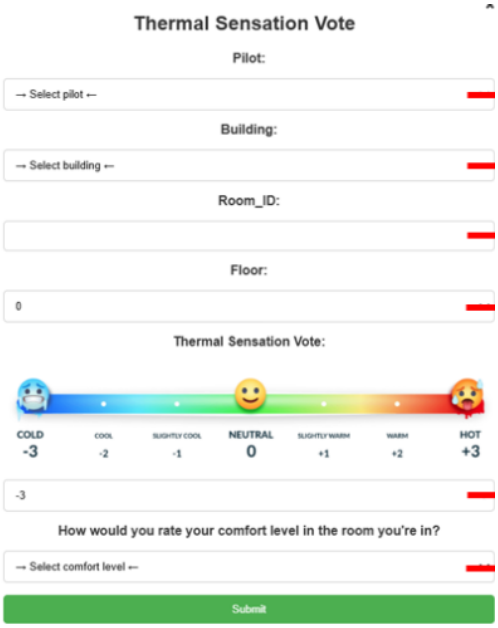


Figura 10. Questionario online

Ai dati relativi al comfort termico si aggiungono anche i dati meteorologici esterni, tra cui la temperatura esterna [°C] e l'umidità esterna, raccolte tramite stazioni meteorologiche nelle vicinanze. In particolar modo, la temperatura esterna riveste un ruolo cruciale, poiché rende possibile il calcolo degli HDD (Heating Degree Days). Gli HDD combinati poi con i consumi energetici mensili ricavati dalle bollette, hanno reso possibile ricostruire l'andamento dei consumi energetici dell'edificio in kWh durante l'intero periodo di misura.

Parallelamente ai dati esterni sono stati raccolti anche i parametri ambientali interni tramite i sensori "Inbiot", installati nell'ambiente di misura, in grado di monitorare in continuo dati come temperatura interna [°C], umidità relativa e concentrazione di CO_2 [ppm], che risultano essenziali per ricostruire l'andamento termico dell'edificio nel tempo.

La tabella seguente riassume le variabili monitorate, con le relative unità di misura e la fonte di acquisizione.

VARIABILE	UNITÀ DI MISURA	FONTE
Temperatura interna	°C	Sensore Inbiot
Umidità relativa interna	%	Sensore Inbiot
CO_2 interna	ppm	Sensore Inbiot
Temperatura esterna	°C	Stazione meteo

Umidità esterna	%	Stazione meteo
TSV	Scala -3…+3	Questionario online
Stato impianto		
Consumi mensili	kWh	Bollette
Radiazione solare	W/m^2	

Tabella 12. Variabili monitorate

La combinazione di dati ambientali interni (T_in, RH, CO_2), dati esterni (T_out, HDD), dati energetici ricavati dalle bollette, voti di comfort (TSV) e stato dell'impianto (On/Off), costituisce un dataset completo e coerente che ha permesso di costruire i tre modelli ML, grazie ai quali, è stato realizzato il simulatore dell'edificio.

Per l'acquisizione dei dati ambientali interni sono stati utilizzati i sensori "Inbiot", strumenti professionali per il monitoraggio continuo delle condizioni termo-igrometriche e della qualità dell'aria negli ambienti indoor.



Figura 11. Sensore Inbiot

Dal punto di vista tecnico, i dispositivi Inbiot, sono dotati di una serie di funzionalità che ne facilitano l'installazione, l'utilizzo e la gestione all'interno di reti di monitoraggio complesse. Tra le principali caratteristiche si evidenziano:

- **Pulsante touch multifunzione**, utile per interazione rapide con il dispositivo.
- **Anello luminoso di stato personalizzabile**, progettato per fornire un feedback visivo immediato sullo stato del sensore.
- **Connettore USB Type-C**, per alimentazione e configurazione.
- **Connettività Wi-Fi,** che permette la trasmissione dati in tempo reale verso piattaforme di raccolta e analisi dati.

Dal punto di vista costruttivo, il dispositivo presenta dimensioni compatte (110 x 110mm) e un peso di 230 grammi, risultando adatto sia a installazioni da tavolo, sia a montaggio a parete.

L'unità Inbiot integra tre sensori principali, ciascuno dedicata alla misura di una delle tre variabili ambientali interne essenziali per la valutazione del comfort termico e della qualità dell'aria:

VARIABILE	TECNOLOGIA	RANGE	RISOLUZIONE	ACCURATEZZA
Temperatura interna (T_in, °C)	Silicon bandgap	-40°C→ 145°C	0.1°C	±0.5°C
Umidità relativa (RH_in, %)	Sensore capacitivo	0 → 100%	1%	±2%
Anidride carbonica (CO_2_in, ppm)	Sensore NDIR	400 → 10.000 ppm	1 ppm	± (30 ppm + 3% del valore letto)

Tabella 13. Caratteristiche dei sensori Inbiot

L'ultima fase del caso studio consiste nell'adattare il simulatore dell'edificio e l'agente di Reinforcement Learning alle condizioni reali osservate durante la campagna di misura. Questo passaggio è essenziale, poiché permette di trasformare l'insieme dei dati raccolti in un ambiente virtuale in cui l'agente può apprendere una strategia di controllo efficace e applicabile all'edificio reale.

Il simulatore dell'edificio è stato costruito combinando i tre modelli Machine Learning trattati in precedenza: modello termico, modello di comfort e modello energetico. Ciascun modello è stato addestrato utilizzando esclusivamente i dati raccolti durante la campagna di misura, garantendo che il comportamento simulato dell'edificio rispecchi fedelmente quello reale.

Una volta costruito il simulatore, è stato necessario configurare l'agente RL affinché potesse interagire direttamente con l'ambiente simulato e apprendere una strategia di controllo efficace. La configurazione dell'agente si basa su tre elementi fondamentali: lo stato, le azioni e la funzione di ricompensa. La tabella seguente mostra come ciascuno di questi elementi sia stato definito a partire dai dati reali raccolti durante la campagna di

misura, garantendo che il comportamento dell'agente sia coerente con la dinamica fisica dell'edificio:

COMPONENTE RL	DESCRIZIONE	VARIABILI UTILIZZATE	COLLEGAMENTO CON IL CASO REALE
Stato	Informazioni che descrivono la condizione dell'edificio al passo corrente	- Temperatura interna prevista - TSV stimato - Consumo energetico previsto, - Stato dell'impianto	Derivano dai tre modelli ML addestrati su dati reali e dallo stato reale dell'impianto durante la campagna di misura
Azioni disponibili	Scelte che l'agente può compiere per controllare l'impianto	0= Spegnere l'impianto 1= Accendere l'impianto	Corrispondono alle due condizioni operative dell'impianto osservate durante la campagna
Ricompensa	Valutazione dell'azione compiuta dall'agente	- Ricompensa positiva se l'ambiente è in comfort - Penalità in caso di discomfort - Penalità proporzionale al consumo energetico quando l'impianto è acceso	Basata sugli obiettivi del caso studio: mantenere comfort e ridurre i consumi.

Tabella 14. Adattamento componenti RL al caso reale

L'integrazione tra simulatore e agente RL permette di ottenere una politica di controllo stabile, coerente e basata sull'esperienza, che può essere successivamente applicata all'edificio reale.

3. Risultati

In questo capitolo vengono presentati e analizzati i risultati ottenuti applicando la strategia di controllo sviluppata tramite l'algoritmo di RL al caso studio descritto nel capitolo precedente. L'obiettivo è quello di confrontare la gestione reale dell'edificio con quella

simulata dall'agente RL, valutando quattro variabili fondamentali: temperatura interna, comfort termico, consumi energetici, azioni (On/Off).

L'analisi dei risultati permette di comprendere come l'agente RL interagisca con il simulatore dell'edificio e quali effetti produca sulle variabili di interesse. In particolare, si osserva come la politica appresa dall'agente sia in grado di modificare il comportamento dell'impianto, influenzando la temperatura interna, il livello di comfort percepito e il consumo energetico complessivo.

Di seguito è riportato il grafico complessivo dei risultati, che riunisce l'andamento della temperatura interna, del comfort termico, dei consumi energetici e delle azioni. Nei paragrafi successivi ciascun elemento verrà analizzato separatamente, in modo da comprendere l'effetto dell'RL su ogni singola variabile.

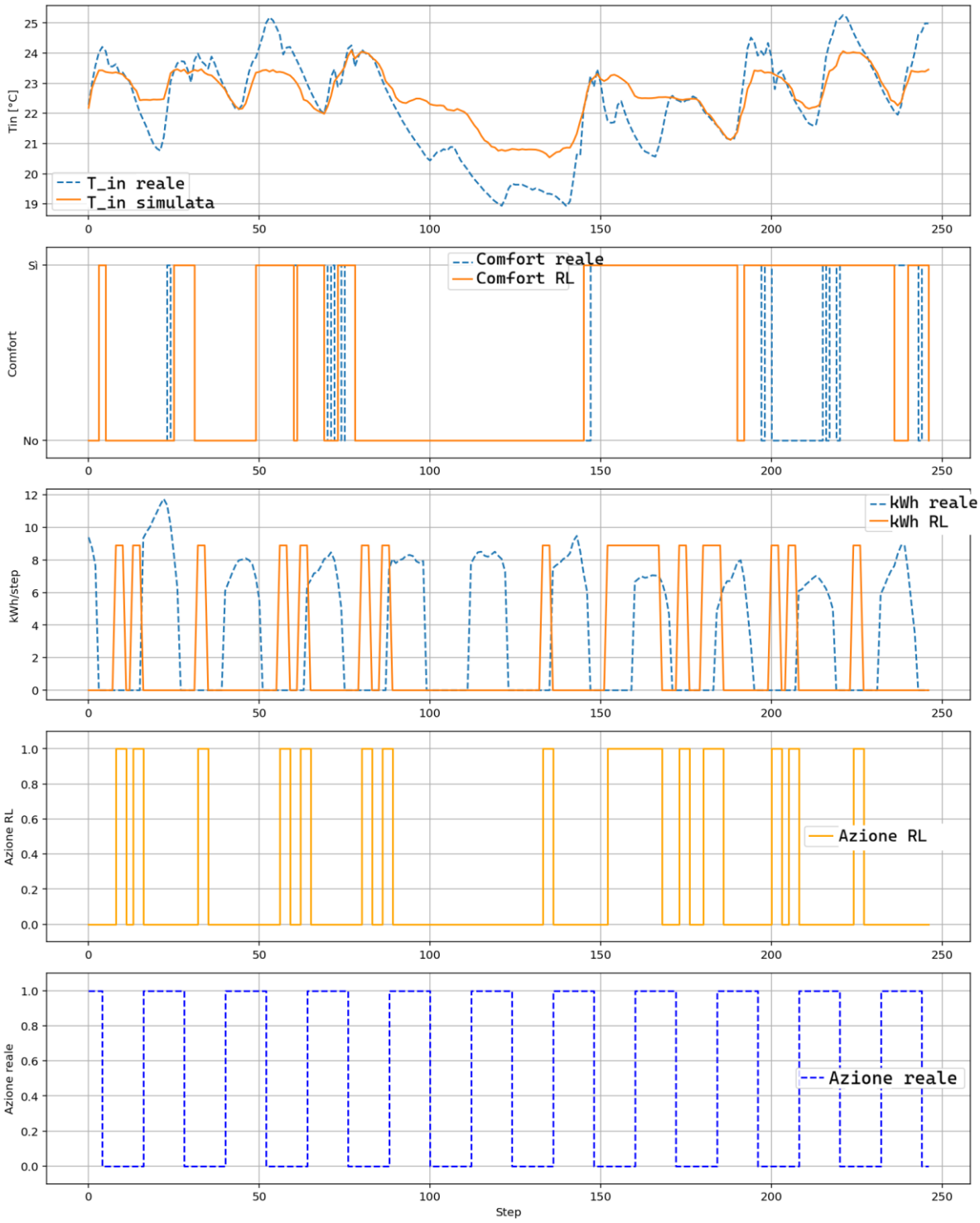


Figura 12. Grafico dei risultati

3.1 Temperatura: reale vs RL

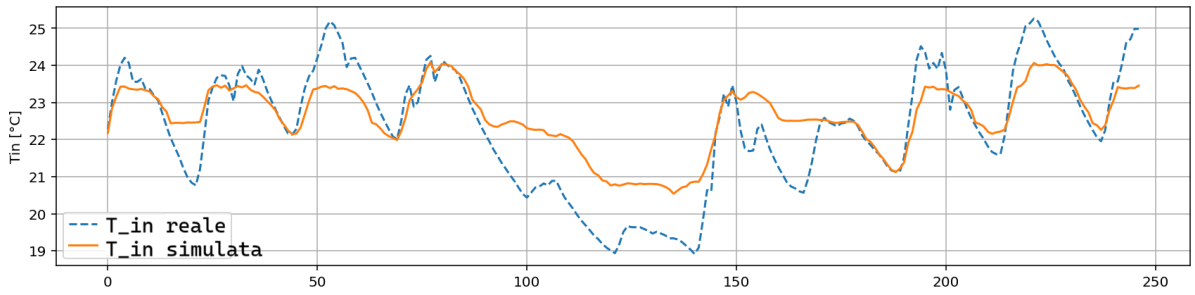


Figura 13. Confronto temperatura interna reale e simulata

Il primo confronto riguarda l'andamento della temperatura interna.

Nel grafico dedicato si osservano due curve:

- T_{in} reale (linea tratteggiata blu): rappresenta l'andamento della temperatura interna misurata durante la campagna di misura.
- T_{in} simulata RL (linea arancione continua): rappresenta la temperatura interna prevista dal simulatore quando l'agente RL applica le proprie azioni.

Le due curve mostrano un andamento molto simile con la temperatura simulata RL che segue fedelmente la curva della temperatura reale, confermando la qualità del simulatore e la stabilità del controllo.

La coerenza tra la temperatura reale e quella simulata dal modello RL è un risultato particolarmente significativo, perché dimostra come il modello termico utilizzato nel simulatore riproduce in maniera credibile la risposta dell'edificio senza deviazioni anomale. Allo stesso tempo, l'assenza di oscillazioni eccessive o variazioni improvvise di temperatura indica che le azioni dell'agente non introducono instabilità nel sistema, ma si integrano con la dinamica termica dell'edificio, mantenendo un comportamento stabile e coerente per tutto l'episodio simulato.

3.2 Comfort: reale vs RL

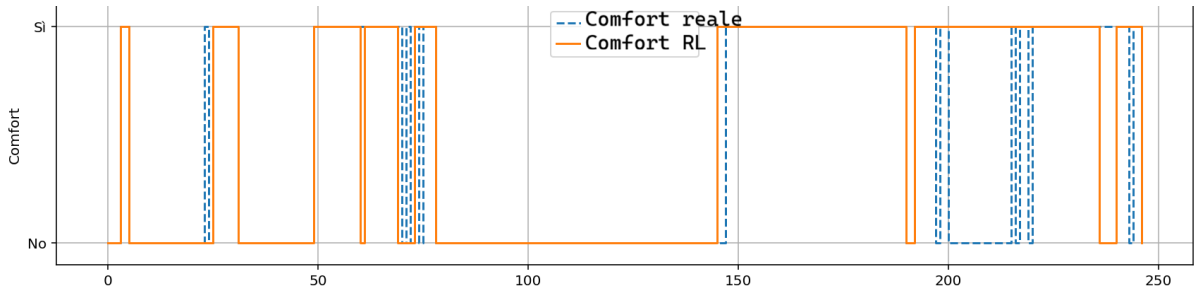


Figura 14. Confronto comfort reale vs RL

Il secondo confronto riguarda il comfort termico, rappresentato come una variabile binaria che indica se l'ambiente si trova in condizioni di comfort oppure di discomfort. La condizione di comfort corrisponde ai valori di TSV compresi tra –0.5 e +0.5, mentre valori al di fuori di tale intervallo indicano una situazione di discomfort.

Nel grafico dedicato si osservano due segnali:

- Comfort reale (linea tratteggiata blu), è la misura del comfort ottenuta durante la campagna di misura tramite i voti dei partecipanti in termini di TSV.
- Comfort RL (linea arancione continua), è il comfort stimato dal modello ML del simulatore quando l'agente applica le proprie azioni On/Off.

Dal grafico emerge come l'agente RL riesca a mantenere il comfort per più tempo rispetto al caso reale. Infatti, se la linea relativa al comfort reale presenta diversi intervalli di discomfort, il grafico simulato dal RL evidenzia una maggiore continuità degli stati in comfort.

Per entrambi i casi il comfort è stato calcolato applicando la stessa regola:

$$comfort = \begin{cases} 1 & se -0.5 < TSV < +0.5 \\ 0 & nei\ restanti\ casi \end{cases} \qquad (10)$$

Successivamente, è stata calcolata la percentuale di step in comfort:

$$Comfort\ \% = \frac{Numero\ di\ step\ con\ comfort = 1}{Numero\ totale\ di\ step} \cdot 100 \qquad (11)$$

Applicando le formule (10) e (11), i risultati che emergono sono i seguenti: il comfort reale risulta pari al 45% degli step totali, mentre il comfort simulato dall'agente RL raggiunge il 51%, garantendo un miglioramento percentuale del 6%

Questo incremento dimostra come l'agente sia riuscito a individuare un modo di gestire l'impianto più efficace, capace di mantenere condizioni di comfort (–0.5 < TSV < 0.5) più frequentemente rispetto alla gestione reale.

3.3 Consumi: reale vs RL

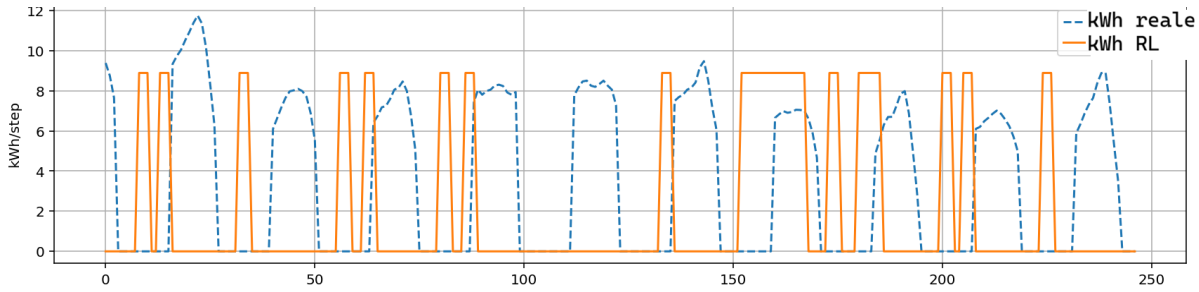


Figura 15. Confronto consumi reali vs RL

Il terzo confronto riguarda i consumi energetici, espressi in kWh.

Nel grafico dedicato sono riportati due andamenti:

- Consumo reale (linea tratteggiata blu), è il consumo durante i dieci giorni della campagna di misura.
- Consumo simulato dall'agente RL (linea arancione continua).

Osservando il grafico si nota chiaramente che, sia nel caso reale sia nel caso simulato, i consumi energetici sono strettamente legati alle accensioni dell'impianto: ogni volta che l'impianto di riscaldamento viene attivato si genera un picco di consumo, mentre quando rimane spento il consumo scende a zero. Questa relazione è evidente in entrambe le curve, ma ciò che cambia è la frequenza e la distribuzione delle accensioni. Infatti, l'agente RL adotta una strategia più selettiva, cerca di evitare accensioni superflue e cerca di accendere l'impianto solo quando è realmente necessario per mantenere il comfort.

Questa differenza è testimoniata dai valori numerici: il consumo reale totale registrato durante la campagna di misura è pari a 846 kWh, mentre il consumo che si ottiene se vengono seguite le azioni suggerite dall'algoritmo RL scende a 505 kWh. L'agente RL

riesce a limitare il numero di accensioni e a distribuirle in modo più efficiente, ottenendo una riduzione complessiva di 341 kWh, pari a circa il 40%.

3.4 Azioni: reale vs RL

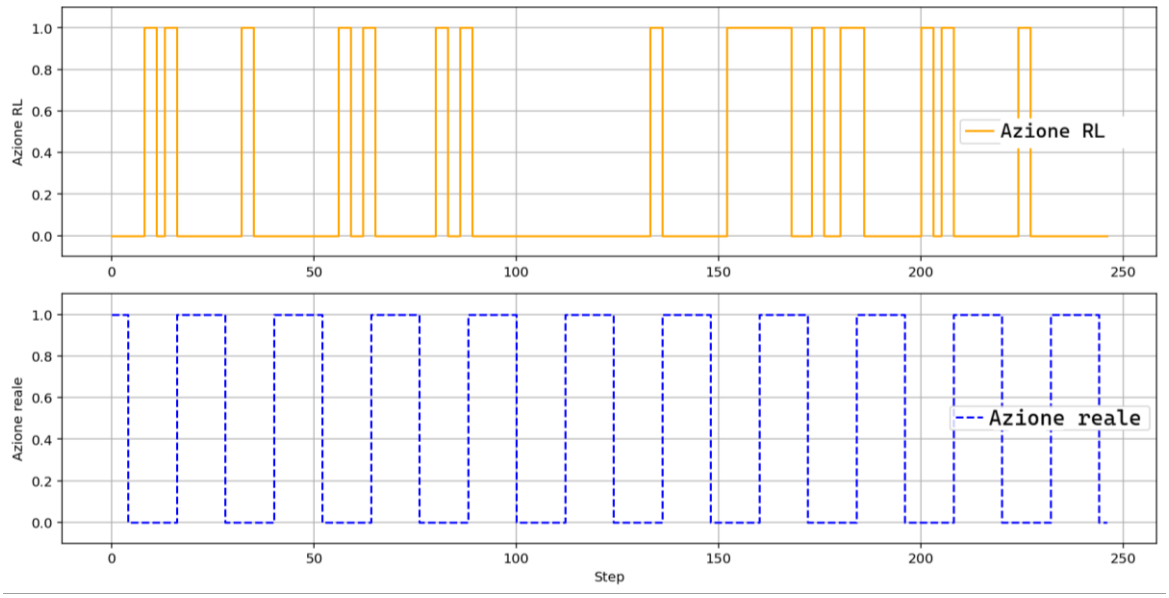


Figura 16. Confronto azioni reali vs simulate

L'ultimo grafico di confronto è relativo alle azioni On/Off dell'impianto.

Anche in questo caso si possono distinguere due curve relative alle due variabili confrontate:

- Azione reale (linea tratteggiata blu): rappresenta quando l'impianto era acceso o spento durante i giorni della campagna di misura.
- Azione RL (linea arancione): rappresenta le decisioni dell'agente RL.

Dal grafico delle azioni On/Off emerge una differenza sostanziale tra il comportamento reale dell'impianto e quello simulato dall'agente RL. Nel caso reale, l'impianto era sempre acceso ad intervalli regolari, in genere da mezzanotte alle undici del mattino, dall'altra parte invece l'obiettivo dell'agente RL è quello di trovare delle accensioni ottimali che gli permettano di mantenere il comfort all'interno dell'ambiente senza avere un consumo energetico eccessivo. Il diagramma delle azioni RL è il risultato del processo di apprendimento dell'agente, che ha compreso come un numero eccessivo di attivazioni dell'impianto porti a dei consumi energetici elevati, mentre un numero troppo ridotto di

accensioni potrebbe compromettere il comfort termico all'interno dell'edificio. Dunque, le azioni riportate nel grafico tramite la linea arancione, sono quelle azioni suggerite dall'agente RL, che se seguite portano a un miglioramento del comfort del 6% e a una riduzione dei consumi di 341 kWh.

4. Discussione e Conclusione

4.1 Discussione dei risultati

Il lavoro presentato in questa tesi affronta il problema della misura e della stima delle grandezze termo-energetiche necessarie alla gestione degli edifici, un tema centrale nella letteratura scientifica e di grande rilevanza applicativa. La corretta valutazione della temperatura interna, del comfort percepito dagli occupanti e dei consumi energetici rappresenta infatti un prerequisito fondamentale per garantire ambienti confortevoli e, al tempo stesso, contenere i consumi energetici. Tuttavia, come evidenziato nello stato dell'arte, la gestione degli impianti termici si basa ancora oggi su logiche semplici e non adattive, spesso prive di misure complete e incapaci di bilanciare in modo ottimale comfort ed energia.

In questo contesto, l'obiettivo della tesi è stato quello di sviluppare una metodologia di misura e controllo completamente data-driven, capace di stimare il comportamento termo-energetico dell'edificio e il comfort percepito dagli occupanti anche in assenza di misure energetiche dirette, e di utilizzare tali informazioni per generare raccomandazioni personalizzate di gestione dell'impianto. L'idea centrale è sostituire le logiche di controllo tradizionali basate su set-point statici con un approccio intelligente, adattivo e fondato sui dati reali dell'edificio.

Dal punto di vista scientifico, l'innovazione principale del lavoro risiede nella costruzione di un simulatore dell'edificio completamente data-driven, ottenuto combinando tre modelli di Machine Learning addestrati esclusivamente sui dati raccolti durante la campagna di misura.

Dal punto di vista misuristico, la tesi introduce una metodologia che integra misure ambientali interne, dati climatici esterni, consumi energetici ricostruiti e voti soggettivi di comfort, costruendo un dataset completo. La presenza di misure orarie, di feedback

soggettivi (TSV) e di dati energetici ricavati dalle bollette ha permesso di rappresentare in modo fedele il comportamento dell'edificio, superando uno dei limiti principali evidenziati nella letteratura: la mancanza di dati completi e affidabili negli edifici reali.

L'integrazione tra misure reali, modelli data-driven e Reinforcement Learning costituisce quindi il contributo metodologico più rilevante della tesi. Il simulatore sviluppato non è un modello teorico, ma una ricostruzione valida del comportamento dell'edificio, che permette all'agente RL di apprendere una politica di controllo senza rischi per gli occupanti e senza necessità di interagire con l'edificio reale.

In questo modo, il lavoro dimostra come sia possibile utilizzare dati misurati, modelli ML e algoritmi RL per generare raccomandazioni personalizzate di accensione e spegnimento dell'impianto, capaci di migliorare simultaneamente comfort ed efficienza energetica.

Per fornire una rappresentazione complessiva delle prestazioni del sistema proposto, nelle tabelle sottostanti sono riassunti i risultati principali dei tre modelli ML e dell'algoritmo di RL, permettendo di valutare in modo immediato il contributo di ciascun componente alla costruzione del simulatore e alla definizione della politica di controllo:

MODELLO / ALGORITMO	TIPO	VARIABILE PREVISTA	METRICHE PRINCIPALI
Modello termico (Random Forest Regressor)	ML	Temperatura interna futura (T_in(t+1)	MAE = 0.22°C R^2 = 0.92
Modello di comfort (Random Forest Classifier)	ML	Comfort (TSV_next)	Accuracy = 0.88 F!-score = 0.93
Modello energetico (Random Forest Regressor)	ML	Consumo energetico (kwh_next)	MAE = 1.38 kWH R^2 = 0.55

Tabella 15. Metriche modelli ML

ALGORITMO	TIPO	OUTPUT	METRICHE PRINCIPALI
Q-learning	RL	Azioni ottimali (ON/OFF)	Comfort: 45% → 51% Consumi: 846 kWh → 505 kWh

Tabella 16. Metriche algoritmo RL

Come evidenziato dalla tabella 15. Il modello termico mostra prestazione eccellenti. Ciò significa che il modello è in grado di prevedere la temperatura interna futura con un errore medio piccolo, garantendo una rappresentazione fedele della dinamica termica dell'edificio. Questa accuratezza è fondamentale perché la temperatura è la variabile fisica da cui dipendono sia il comfort sia il consumo energetico.

Il modello di comfort, con le sue metriche, riesce quindi a distinguere in modo affidabile le condizioni di comfort da quelle di discomfort, fornendo all'agente RL un'informazione coerente e stabile su cui basare la funzione di ricompensa.

Il modello energetico è il più semplice dei tre e presenta metriche più contenute (MAE = 1.38 kWh, R^2 = 0.55). Tuttavia, il suo obiettivo non è riprodurre perfettamente la variabilità dei consumi reali, bensì fornire una stima coerente con le azioni dell'agente RL. In questo senso, il modello risulta adeguato: ogni accensione simulata produce un picco di consumo sincronizzato, permettendo di valutare correttamente l'effetto delle decisioni dell'agente.

Infine, la politica RL appresa dall'agente mostra un miglioramento simultaneo del comfort e dell'efficienza energetica. Il comfort passa dal 45% al 51% degli step totali, mentre i consumi energetici si riducono da 846 kWh a 505 kWh, con un risparmio pari a circa il 40%. Questi risultati dimostrano che l'agente RL è in grado di individuare una strategia di controllo più razionale rispetto alla gestione reale, intervenendo nei momenti più opportuni e riducendo le accensioni superflue.

In conclusione, l'approccio basato sul Reinforcement Learning dimostra come sia possibile superare i limiti della gestione tradizionale dell'impianto, coniugando un maggior benessere termico con un netto taglio dei consumi energetici. Questo risultato conferma la validità dell'intero approccio metodologico e rappresenta un passo importante verso l'adozione di tecniche di controllo avanzate negli edifici reali.

4.2 Limiti del lavoro e sviluppi futuri

Nonostante i risultati ottenuti confermino la potenzialità e la validità dell' approccio basato sul Reinforcement Learning per il controllo termo-energetico degli edifici, il lavoro presenta delle criticità di cui è necessario discutere. Questi limiti non riducono il

valore del progetto, ma ne definiscono il campo di validità e forniscono importanti spunti per futuri miglioramenti del sistema trattato.

Il primo limite riguarda la quantità di dati reali disponibili poiché l'intera campagna di misura si è svolta nell'arco di dieci giorni, un periodo relativamente breve per addestrare opportunamente un algoritmo di Reinforcement Learning. La limitata durata del periodo di monitoraggio influisce sia sulla qualità dei modelli Machine Learning, riducendone la capacità di generalizzare a fronte di pochi esempi disponibili, sia sulla fase di addestramento dell'agente RL, che si trova ad operare in un ambiente simulato costruito su dati non particolarmente variabili. Un'estensione della campagna di misura su più settimane o mesi rappresenta quindi un naturale sviluppo futuro, permettendo di catturare una gamma più ampia di situazioni e di migliorare la robustezza sia del simulatore sia della politica RL.

Un secondo limite riguarda l'assenza di misure energetiche dirette. L'edificio non dispone di smart meter e ciò ha reso necessario stimare i consumi tramite Heating Degree Days e regressioni con le bollette mensili. Sebbene questa metodologia sia coerente con la letteratura, essa rimane una stima indiretta, che introduce un margine di incertezza nella funzione di reward. Un possibile sviluppo futuro consiste nell'installazione di sensori energetici dedicati, così da costruire un modello dei consumi basato su misure reali e definire una funzione di ricompensa più accurata.

Un terzo aspetto da considerare riguarda la validazione della metodologia. Pur essendo stata applicata a un edificio reale, la metodologia non è stata testata in una casistica dotata di un sistema HVAC automatizzato, condizione necessaria affinché l'agente RL possa interagire direttamente con l'impianto. La validazione esclusivamente su simulatore, seppur coerente con il contesto tecnico disponibile, non permette di verificare il comportamento dell'agente in presenza di variabilità non catturate dal modello. Un futuro sviluppo consiste quindi nell'applicare la metodologia in un edificio dotato di controllo automatico, così da testare la politica RL in condizioni reali e valutarne la stabilità nel tempo.

Infine, un ulteriore limite riguarda la generalizzabilità del sistema. Il simulatore e la politica RL sono stati costruiti e addestrati su un singolo edificio, con caratteristiche specifiche. Dato che ogni edificio presenta una propria dinamica termica, una propria

geometria, un proprio isolamento e un proprio comportamento degli occupanti, non è garantito che la politica appresa possa essere trasferita direttamente ad altri edifici senza una fase di adattamento o ri-addestramento. Un possibile sviluppo consiste nell'applicare la metodologia a più edifici, così da valutare la capacità del simulatore di generalizzare e da esplorare l'adozione di algoritmi di Deep Reinforcement Learning, capaci di gestire stati continui e dinamiche più complesse.

In sintesi, i limiti individuati riguardano la disponibilità dei dati, l'assenza di misure energetiche dirette, la mancanza di validazione su un sistema HVAC automatizzato e la difficoltà di generalizzare la politica appresa. Per ciascuno di essi esiste una chiara direzione di sviluppo: estendere la campagna di misura, installare smart meter, validare la metodologia in edifici automatizzati e adottare algoritmi RL più avanzati. Questi miglioramenti permetterebbero di trasformare il sistema sviluppato in questa tesi in un sistema di controllo avanzato, capace di ottimizzare comfort e consumi energetici in modo autonomo, affidabile e applicabile a una più ampia varietà di edifici.

5. Bibliografia

[1]Wei Li , Yifan Zhao, Jili Zhang, Changwei Jiang, Siyu Chen, Liangxi Lin, Yuegui W ang “Indoor temperature preference setting control method for thermal comfort and energy saving based on reinforcement learning”, 2023.

[2] ISPRA – Istituto Superiore per la Protezione e la Ricerca Ambientale, *Consumi energetici e heating degree days (HDD) a confronto. Proiezioni al 2050 degli HDD in differenti scenari climatici*, 2017.

[3] Zhipeng Deng, Qingyan Chen, “Reinforcement learning of occupant behavior model for cross-building transfer learning to various HVAC control systems”,2021.

[4] Fanger, P.O. Thermal Comfort: Analysis and Applications in Environmental Engineering. McGraw-Hill, 1970

[5] E. N. I. d. Unificazione, «UNI EN ISO 7730: Ergonomia degli ambienti termici - Determinazione analitica e interpretazione del benessere termico mediante il calcolo degli indici PMV e PPD e dei criteri di benessere termico locale,» 2006

[6] ASHRAE, «Standard 55-2023 - Thermal Environmental Conditions for Human Occupancy,» 2023.

[7] Evandro Eduardo Broday, Jéferson Aparecido Moreto, Antonio Augusto de Paula Xavier, Reginaldo de Oliveira, “The approximation between thermal sensation votes (TSV) and predicted mean vote (PMV): A comparative analysis”, 2019

[8] C. Buratti, P. Ricciardi e V. M., HVAC systems testing and check: A simplified model to predict thermal comfort conditions in moderate environments, » 2013.

[9] Mazzucchelli E. “Ottimizzazione di un modello per la misura del comfort indoor: sviluppo e validazione tramite un caso studio reale.” Tesi di laure triennale UNIVPM 2024.

[10] Anderson Energy Efficiency “How to quantify comfort with PMV”.

[11] N. Morresi, S. Casaccia, M. Sorcinelli, M. Arnesano, A. Uriarte, J. I. Torrens-Galdiz, G. M. Revel, "Sensing Physiological and Environmental Quantities to Measure Human Thermal Comfort Through Machine Learning Techniques", IEEE Sensors Journal, 2021

[12] A. Cipollone, N. Morresi, S. Casaccia, G. M. Revel, "A Questionnaire-Free Approach for Thermal Comfort Measurement Using Unsupervised Machine Learning on Physiological Signals", IEEE Sensors Journal, 2025.

[13] Deb, C., et al. "*A review on data-driven approaches for building energy prediction*". Renewable and Sustainable Energy Reviews, 2017

[14] Siliang Lu, Weilong Wang, Chaochao Lin, Erica Cochran Hameen, "Data-driven simulation of a thermal comfort-based temperature set-point control with ASHRAE RP884", 2019

[15] Youssef Boutahri, Amine Tilioua, "Machine learning-based predictive model for thermal comfort and energy optimization in smart buildings", 2024.

[16] Y. Chen, L.K. Norford, H.W. Samuelson, *et al.* "Optimal control of HVAC and window systems for natural ventilation through reinforcement learning", 2018.

[17] W. Valladares, M. Galindo, J. Gutierrez, *et al.* "Energy optimization associated with thermal comfort and indoor air control via a deep reinforcement learning algorithm", 2019.

[18] Fangzhou Guo , Sang woo Ham , Donghun Kim , Hyeun Jun Moon "Deep reinforcement learning control for co-optimizing energy consumption, thermal comfort, and indoor air quality in an office building", 2025.

[19] Cheng Gao, Dan Wang, "Comparative study of model-based and model-free reinforcement learning control performance in HVAC systems", 2023.

[20] Zhipeng Deng, Qingyan Chen "Reinforcement learning of occupant behavior model for cross-building transfer learning to various HVAC control systems", 2021.

[21] "Building Performance Simulation – an overview". ScienceDirect Topics.

[22] Breiman, L. "Random Forest. Machine Learning.", 2001.

[23] Fernandez-Delgado et al. “Do we need Hundreds of classifier to solve real world classification problems?” Journal of Machine Learning Research, 2014.

[24] Sutton e Barto, “Reinforcement Learning: An introduction”. 2018.