

Creazione di un modello accurato di parametri meteorologici per un
simulatore di sistemi ferroviari nell'ambito del progetto Stingray

Università di Pisa



Lorenzo Gerini

14 Ottobre 2019

Introduzione

Questo progetto di tesi nasce come contributo al progetto Stingray, il quale si propone di studiare, progettare, sviluppare e sperimentare soluzioni che permettano un consistente risparmio energetico nell'esercizio delle stazioni ferroviarie, il tutto svolto con un approccio Internet of Things. Il progetto nasce dalla collaborazione di diversi soggetti tra cui i Dipartimenti di Ingegneria dell'Informazione dell'Università di Pisa e Firenze e l'Istituto di Scienza e Tecnologie dell'Informazione - CNR.

L'obiettivo posto da questa tesi è quello di creare un modello di parametri meteorologici che venga integrato all'interno del Simulatore di Sistemi Ferroviari utilizzato nell'ambito del progetto Stingray, in modo da arricchirne le funzionalità e rendere le simulazioni stesse più precise e coerenti. Per perseguire questo scopo sono stati utilizzati dati meteorologici raccolti mediante sensori posizionati in diverse aree della città di Pisa. I dati sono stati elaborati in modo da poter generare diversi profili meteorologici, questi sono stati poi adattati al formato già specificato all'interno del simulatore per descrivere il modello meteorologico. Lo sviluppo di questo progetto passa attraverso diverse fasi, seguite anche nella stesura di questo documento, partendo da una presentazione del Progetto Stingray e da una analisi delle principali componenti del Simulatore di Sistemi Ferroviari che interagiscono con il modello meteorologico.

Come obiettivo secondario si è posto quello di realizzare una politica che regoli l'attività delle scaldiglie, elementi di fondamentale importanza sia per i sistemi ferroviari stessi che per il fine del progetto Stingray rivolto al risparmio energetico.

Indice

1 Il Progetto STINGRAY	4
1.1 Il progetto	4
1.2 La gestione efficiente delle risorse energetiche	4
1.3 Architettura logica del Sistema Ferroviario	5
2 Creazione di profili meteorologici	8
2.1 Il modello meteorologico nel Simulatore	8
2.2 Elaborazione dei dati meteorologici reali	9
2.3 Generazione dei dati del profilo meteorologico	10
2.4 Scrittura del profilo meteorologico all'interno del simulatore	14
2.5 Analisi dei risultati	14
3 Realizzazione di una politica di regolazione dell'attività delle scaldiglie	18
3.1 La politica di accensione e spegnimento	18
3.2 Il Metodo Moltiplicativo di Holt-Winters	20
3.3 Inizializzazione del metodo	21
3.4 Implementazione del modello	21
3.5 Stima dei parametri	24
3.6 Analisi dei risultati	24
4 Conclusioni	29
Riferimenti	30

1 Il Progetto STINGRAY

1.1 Il progetto

STINGRAY (SmarT IntelliGent RAILwaY) è un progetto della Regione Toscana collocato all'interno del programma POR FESR Toscana 2014-2020. L'obiettivo di STINGRAY è quello di rinnovare il ruolo della stazione ferroviaria, tradizionalmente vista come un punto d'incontro per la città, in modo da migliorarne l'integrazione nelle Smart Cities del futuro. Le stazioni ferroviarie rappresentano un punto centrale per ogni centro urbano, ma tradizionalmente sono isolate dal contesto urbano che le circonda. Esse hanno un sistema di comunicazione e di distribuzione dell'energia privato, principalmente per prevenire blackout e intrusioni non autorizzate. Questo isolamento ha due lati negativi. Per prima cosa non c'è integrazione con le cosiddette Smart Cities, dove le informazioni e i dati prodotti dai diversi sistemi di trasporto (bike-sharing, car-sharing, e trasporto urbano tradizionale) sono sfruttati sinergicamente. In secondo luogo, il sistema stazione è escluso dalle moderne tecniche di risparmio energetico. STINGRAY si focalizza su questi punti attraverso lo studio, la progettazione e lo sviluppo di una infrastruttura di comunicazione per la stazione che, integrando tecnologie power line e wireless, si occupi di:

- sviluppare una rete LAN che copra l'area della stazione;
- permettere il controllo e il monitoraggio delle apparecchiature della stazione;
- creare servizi di valore aggiunto per i clienti e lo staff ferroviario;

STINGRAY rappresenta un'opportunità per una più efficiente gestione dei macchinari e delle risorse energetiche. Promuove l'innovazione tecnologica, ottenuta senza la costruzione di nuove e costose infrastrutture, grazie all'uso delle power line, garantendo un minore impatto ambientale. Sulla base di questi presupposti, il concetto di Smart Station è concepito per adottare i modelli e le politiche già integrati nelle Smart Cities e nel mondo IoT, in modo da innovare la gestione delle stazioni e delle risorse energetiche, eliminando lo spreco di risorse e altre inefficienze.

1.2 La gestione efficiente delle risorse energetiche

L'uso efficiente dell'energia a tutti gli stadi della catena energetica, dalla produzione al consumo finale, sta acquisendo sempre più importanza in tutti i settori. Aspetti economici e considerazioni sull'impatto ambientale sono alla base di questo nuovo trend. STINGRAY si pone

l'obiettivo di migliorare l'efficienza energetica in due particolari aree di maggiore interesse per le Smart Station del futuro: il sistema di riscaldamento degli scambi ferroviari e il sistema di illuminazione della stazione. Uno scambio ferroviario è un meccanismo che permette ai treni di essere guidati da un binario all'altro. Per assicurare l'esecuzione corretta delle operazioni, è equipaggiato con un sistema di riscaldamento che previene la formazione di ghiaccio quando le temperature sono basse. Nel progetto STINGRAY, il sistema di riscaldamento e le politiche di controllo applicate su di esso, sono analizzati in funzione dell'ambiente in cui il sistema opera (condizioni meteorologiche), tenendo conto anche di eventi di perdita di comunicazione, sfruttando indicatori rappresentativi del consumo di energia.

Il framework di modellazione e analisi fino a qui sviluppato nell'ambito del progetto STINGRAY è caratterizzato dalle seguenti componenti: 1) rappresentazione della struttura del sistema di riscaldamento degli scambi ferroviari, tipicamente organizzato in una gerarchia di componenti collegati mediante una infrastruttura di comunicazione PLC; 2) rappresentazione di politiche di controllo del riscaldamento, espresse in termini di soglie parametriche; 3) rappresentazione di sofisticati elementi meteorologici, come temperatura, umidità e le relazioni tra queste; 4) rappresentazione di una varietà di modelli di fallimento che coinvolgono l'infrastruttura di comunicazione e/o i sensori per la raccolta dei dati meteorologici; 5) abilità di eseguire analisi di sensitività variando i valori rappresentativi di ogni modello. Le analisi ottenute sono utili per comprendere l'impatto dei diversi fenomeni nella valutazione dei consumi energetici e sono un utile base per ideare migliori politiche di riscaldamento che tengano conto del consumo energetico.

Le analisi delle politiche messe in atto per migliorare l'efficienza energetica sono eseguite studiando i dati prodotti da un Simulatore di Sistemi Ferroviari, che presenta al suo interno una architettura logica del sistema ferroviario e, più precisamente, del sistema di riscaldamento, così come è stata definita nell'ambito del progetto STINGRAY. Questa architettura è descritta nel paragrafo che segue.

1.3 Architettura logica del Sistema Ferroviario

Prima di definire nei dettagli il sistema che è stato preso in considerazione per questo progetto, è bene introdurre due componenti che stanno alla base dello stesso, le scaldiglie e gli scambi ferroviari. Le scaldiglie sono resistenze elettriche installate in prossimità degli scambi

ferroviari, cioè i punti di giunzione che permettono alle rotaie di allinearsi. La loro funzione è quella di scaldare gli scambi in modo da impedire al ghiaccio di bloccarli.

Il sistema di riferimento all'interno del Simulatore è logicamente strutturato in tre parti, come mostrato in Figura 1: i) n scaldiglie SH_1, \dots, SH_n , una per ogni scambio, le quali scaldano gli scambi attraverso l'elettricità, ii) il sistema di controllo delle scaldiglie, che decide quando accendere o spegnere le scaldiglie stesse in modo da prevenire il congelamento degli scambi ferroviari, e iii) le condizioni meteo, cioè i dati meteorologici, che variano nel tempo, usati dal sistema per decidere quali azioni di controllo intraprendere.

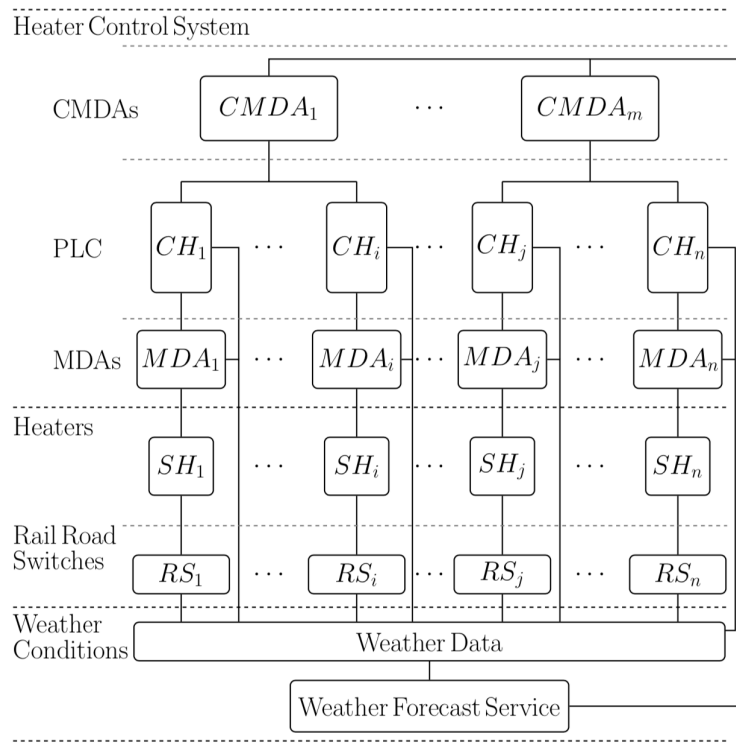


Fig 1. Architettura logica del sistema

Ogni scaldiglia SH_i è installata in prossimità di uno switch RS_i ed è alimentata attraverso delle power-line connesse al sistema energetico. Più scaldiglie possono essere connesse in serie sulla stessa power-line. All'istante t lo stato di ciascuna scaldiglia è descritto mediante il processo stocastico $\{X_i(t) | t \geq 0\}$ definito da

$$X_i(t) = \begin{cases} 1 & \text{Se } SH_i \text{ è accesa all'istante } t \\ 0 & \text{altrimenti} \end{cases}$$

Una scaldiglia consuma energia elettrica soltanto quando è accesa.

Il Sistema di Controllo delle scaldiglie è composto dai seguenti componenti logici:

i) n Moduli di Acquisizione Dati (MAD) MDA_1, \dots, MDA_n , uno per ogni scaldiglia partizioni in m differenti gruppi, ii) m Coordinatori di Moduli di Acquisizione Dati (CMAD) $CMAD_1, \dots, CMAD_m$ uno per ogni differente gruppo in cui l'insieme dei MAD è stato partizionato, iii) un sistema di Comunicazione Power-line (PLC), composto da n canali di comunicazione logica CH_1, \dots, CH_n , ognuno dei quali connette un MAD con il CMAD associato. CMAD e MAD hanno il compito di accendere o spegnere le scaldiglie in accordo alle politiche di accensione e spegnimento implementate, tra cui quella descritta in questo documento. Ogni MDA_i include un sensore per misurare la temperatura sul RS_i associato, questo sensore è installato vicino alla SH_i ed è alimentato dalla stessa power-line di SH_i , cioè della scaldiglia. Ogni $CMAD_j$ è alimentato dalla stessa power-line collegata a tutti i MAD controllati da $CMAD_j$.

Ogni stazione ferroviaria includerà al suo interno un numero diverso di CMAD e MAD, in base alla sua struttura. Le politiche di gestione delle scaldiglie, e lo stesso modello meteorologico soggetto di questa tesi, saranno indipendenti dalla conformazione della topologia della rete che collega i MAD ai corrispettivi CMAD, differente per ogni stazione ferroviaria. Come soggetto di studio, il simulatore sfrutta la struttura presente nella stazione di Lecco Maggianico (LC), in cui sono presenti 2 CMAD e 19 MAD di cui 9 associati al CMAD North e 10 associati al CMAD South.

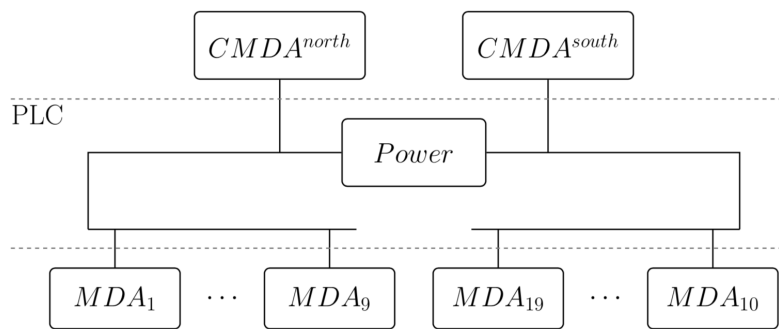


Fig 2. Architettura logica del sistema PLC nella stazione di Lecco Maggianico

2 Creazione di profili meteorologici

2.1 Il modello meteorologico nel Simulatore

Il modello meteorologico all'interno del Simulatore è definito mediante due matrici, la matrice *temperature* e la matrice *humidity*, che descrivono l'evoluzione meteorologica di una giornata. La matrice *temperature* è definita come segue

$$\begin{bmatrix} T_{MAD_{1,1}} & \cdots & T_{MAD_{1,m}} \\ \vdots & \ddots & \vdots \\ T_{MAD_{n,1}} & \cdots & T_{MAD_{n,m}} \\ T_{A_1} & \cdots & T_{A_m} \end{bmatrix}$$

Dove $T_{MAD_{i,j}}$ rappresenta la temperatura rilevata in prossimità del MAD di indice i all'istante j , con $1 \leq i \leq n$, $n = 19$ e $1 \leq j \leq m$, $m = 144$ prendendo in considerazione la stazione di Lecco Maggianico e la periodicità del Simulatore che prevede aggiornamenti meteorologici ogni 10 minuti. Infatti, tenendo conto che la matrice descrive l'evoluzione meteorologica di una giornata, e che gli aggiornamenti devono avvenire ogni 10 minuti, i valori di temperatura per ogni MAD risultano essere 144. Lo stesso vale per i valori della temperatura dell'aria $T_{A_1} \dots T_{A_m}$, collocati nell'ultima riga della matrice.

La matrice *humidity* è definita secondo lo schema della matrice *temperature*, ovvero le prime n righe corrispondono alle umidità rilevate dai MAD, mentre l'ultima riga corrisponde all'evoluzione dell'umidità dell'aria.

Queste matrici sono definite staticamente all'interno di uno dei file che compongono il simulatore (*params_def_in.h*), non variano tra una simulazione e l'altra e non permettono quindi una analisi precisa delle politiche messe in atto per migliorare l'efficienza energetica. Lo scopo, da qui in poi, sarà quello di implementare un sistema esterno al simulatore che produca profili meteorologici (intesi come coppie di matrici (*temperature*, *humidity*)) che possano essere utilizzati all'interno del simulatore come modelli meteorologici alternativi a quello statico già presente.

2.2 Elaborazione dei dati meteorologici reali

Al fine di generare profili meteorologici, ovvero generare i dati necessari per l'inizializzazione delle matrici *temperature* e *humidity*, sono stati utilizzati dati reali (comprendenti valori come temperatura, umidità, ora e data di raccolta ecc.) rilevati da tre sensori dislocati in zone diverse della città di Pisa. Questi dati sono stati raccolti tra il 01/01/2017 e il 31/12/2017. In questo periodo di tempo i sensori non hanno raccolto dati in maniera continuativa, in particolare i sensori 1 e 3 sono stati attivi per più tempo rispetto al sensore 2. In modo da facilitare l'elaborazione dei dati, il data set originale è stato suddiviso in tre data set distinti, ognuno relativo a uno dei sensori.

Per produrre i dati che andranno poi a formare il profilo meteorologico è stato realizzato un programma C++, Weather Profile Generator (*wpg*) il cui output sarà un profilo meteorologico composto da:

- Un vettore *airTemperature*, contenente 144 valori che descrivono l'andamento della temperatura dell'aria nell'arco di una giornata.
- Un vettore *airHumidity*, contenente anch'esso 144 valori relativi, questa volta, all'andamento dell'umidità dell'aria.
- Un vettore *dewPoint*, contenente i valori del dew point calcolati a partire dai due vettori precedenti.
- Una matrice *MADsTemp*, contenente, per ogni MAD, i valori di temperatura rilevati nelle vicinanze del dispositivo nell'arco della giornata.
- Una matrice *MADsHumi*, contenente per ogni MAD i valori di umidità rilevati nelle vicinanze del dispositivo nell'arco della giornata.

Il programma consente, specificando alcuni parametri in ingresso, di inizializzare il valore di due variabili sulla base delle quali vengono poi elaborati i dati. Le due variabili sono *correctionFactor* e *onlyWinter*, definite nel file *wpg.h*.

La prima permette di definire un fattore di correzione che andrà ad agire sui valori di temperatura dei dati originali con un decremento pari al valore della variabile stessa. Il valore di default è 5, quindi, in caso nessun valore venisse specificato all'avvio del programma, ogni temperatura verrà decrementata di 5°C. In caso il parametro fosse impostato a 0 nessun decremento

verrebbe applicato alle temperature originali, ottenendo così valori che variano nell'intervallo $[-5.62, 40.53]$. Il valore di *correctionFactor* inserito deve essere compreso tra 0 e 20.

La seconda è una variabile di tipo booleano che, se impostata a *true*, permette di escludere dall'elaborazione dei dati tutti i valori corrispondenti a mesi non invernali. Il valore impostato di default è *true*, producendo temperature che variano nell'intervallo $[-5.62, 23.80]$. In caso all'avvio dell'esecuzione fosse impostato il valore *false* verrebbero presi in considerazione tutti i dati originali.

Il programma *wpg* è composto da due funzioni principali: *generateWeatherProfile(...)* e *writeWeatherProfile(...)*. La prima si occupa di importare i dati dei tre data set precedentemente creati, elaborare i dati in essi contenuti ed infine generare i dati per le matrici *temperature* e *humidity* da scrivere nel file *params_def_in.h* del simulatore. Tutte queste fasi sono gestite da funzioni intermedie, tra cui:

- *importData(...)*

Permette di importare i dati contenuti nel data set specificato come argomento, salvandoli in una matrice realizzata tramite il tipo *vector*.

- *onlyWinterData(...)*

Questa funzione viene eseguita in caso la variabile *onlyWinter* avesse valore *true*. Essa permette di creare un nuovo data set a partire dal data set precedentemente importato, il quale conterrà soltanto dati relativi a mesi non invernali.

Il data set importato e fin qui elaborato viene scelto randomicamente tra i tre disponibili. In caso la variabile *onlyWinter* fosse attiva, il data set relativo ai dati del sensore 2 viene escluso direttamente dalla scelta poiché esso non contiene rilevazioni acquisite in mesi invernali. Questo data set finale sarà quello a partire dal quale verranno estratti i dati per inizializzare le varie componenti del profilo meteorologico.

2.3 Generazione dei dati del profilo meteorologico

Una volta ottenuto il data set finale, viene scelto, anche questa volta in maniera randomica, un indice *randSample*, compreso nell'intervallo $[1, \text{dimDataset} - n\text{Samples}]$, dove *dimDataset*

rappresenta la dimensione del data set in questione e $nSamples$ rappresenta il numero di rilevazioni raccolte nell'arco di una giornata dai sensori. L'indice così ottenuto viene poi corretto, in modo che esso si riferisca alla prima rilevazione della stessa giornata a cui esso appartiene. Ovvero si avrà

$$randSample = randSample - (randSample \% nSamples)$$

Partendo da $randSample$, vengono estratti dal data set i valori di temperatura e umidità necessari a coprire l'arco di una giornata, cioè la durata di una simulazione. Questo viene effettuato attraverso la funzione `initAirValues(...)`. Poiché, i dati originali contengono rilevazioni raccolte ogni 30 minuti, mentre il simulatore lavora con valori che vengono aggiornati ogni 10 minuti, i dati estratti dalla funzione `initAirValues(...)` vengono interpolati in modo da portare la dimensione dei vettori relativi ai valori di temperatura e umidità dell'aria da 48, numero di rilevazioni effettuate dai sensori nell'arco di una giornata, a 144. Ciò viene svolto mediante un'interpolazione lineare implementata nella funzione `expandAirValues(...)` che permette di ottenere i vettori di temperatura ($airTemperature$) e umidità ($airHumidity$) dell'aria utilizzati nel profilo meteorologico finale.

Utilizzando i due vettori appena citati, viene calcolato un terzo vettore ($dewPoint$) che andrà a far parte del profilo meteorologico, ovvero quello contenente i valori di dew point. Questo calcolo viene svolto mediante la funzione `initDewPoint(...)` che utilizza l'approssimazione di Magnus-Tetens in cui il dew point all'istante t_j è espresso in funzione di temperatura e umidità dell'aria nello stesso istante come segue

$$T^{dew}(t_j) = \frac{b\alpha(t_j)}{a - \alpha(t_j)}, \quad \text{con } \alpha(t_j) = \frac{aT_{A_j}}{b + T_{A_j}} + \ln(H_{A_j}),$$

dove $a = 17.27$ e $b = 237.7$ C° [18, 24], mentre T_{A_j} e H_{A_j} corrispondono a temperatura e umidità dell'aria all'istante t_j .

Per completare il profilo meteorologico rimangono da individuare le matrici contenenti i dati relativi ai MAD, ovvero le matrici $MADsTemp$ e $MADsHumi$. Per quanto riguarda questi ultimi si tratta di passare da un dato più generico, cioè quello delle temperature e delle umidità dell'aria, ad uno più specifico che contiene dati precisi rilevati da sensori, in questo caso i MAD.

Per prima cosa si ricavano i valori delle temperature rilevate dai MAD. A tale scopo è stata implementata la funzione `initMADsTemp(...)` che, partendo dal vettore *airTemperature*, genera per ogni MAD 144 valori, uno per ogni istante, come segue:

- All'istante t_0 le temperature rilevate dai MAD sono le stesse di quelle dell'aria.
- Per ogni istante t_j , $j > 0$ viene individuata la fascia oraria in cui l'istante t_j si trova. Un istante può appartenere a una tra quattro fasce orarie: 00:00 — 07:00, 07:00 — 13:30, 13:30 — 20:00 o 20:00 — 00:00. Determinata la fascia oraria a cui tale istante appartiene, viene valutata, in funzione di essa, la probabilità che il valore della temperatura subisca un incremento o un decremento. La probabilità che la temperatura incrementi in una determinata fascia oraria h è data da $P_u(h) = U(h)/(U(h) + D(h))$, dove $U(h)$ è pari al numero di volte in cui la temperatura incrementa all'interno della fascia oraria h e $D(h)$ è il numero di volte in cui essa diminuisce, andando a valutare i valori delle temperature all'interno del vettore *airTemperature* già ottenuto. La probabilità che la temperatura diminuisca nella stessa fascia oraria è invece data semplicemente da $P_d(h) = 1 - P_u(h)$.
- A questo punto si definisce la temperatura del MAD all'istante t_j come $T_{MAD_{i,j}} = T_{MAD_{i,j-1}} + v$ (1), dove v è un parametro che assume valori positivi con probabilità pari a $P_u(h)$ e valori negativi con probabilità pari a $P_d(h)$. Il valore di v viene scelto randomicamente all'interno di due intervalli: $[min(V_j^+), max(V_j^+)]$ (2), nel caso la variazione fosse positiva, e $[min(V_j^-), max(V_j^-)]$ (3) nel caso la variazione fosse negativa. Dove V_j^+ è l'insieme delle variazioni positive che si possono ottenere a partire dalla temperatura T_{A_j} e V_j^- è l'insieme delle variazioni negative che si possono ottenere a partire dalla stessa temperatura.
- I due intervalli sopracitati vengono valutati come segue. Sia T_{A_j} la temperatura dell'aria all'istante t_j , si prendono in considerazione tutti i valori delle temperature appartenenti al vettore *airTemperature* che si trovano nell'intervallo $[T_{A_j} - e, T_{A_j} + e]$ dove $e = 0.1$ e si definiscono gli insiemi V_j^+ e V_j^- rispettivamente come
 - l'insieme delle variazioni positive, definite come $v = T_{k+1} - T_k > 0$, che si possono ottenere a partire dai valori di temperatura T_k appartenenti all'intervallo $[T_{A_j} - e, T_{A_j} + e]$ e

- l'insieme delle variazioni negative, definite come $v = T_{k+1} - T_k < 0$, che si possono ottenere a partire dai valori di temperatura T_k appartenenti all'intervallo $[T_{A_j} - e, T_{A_j} + e]$.

- Definiti questi insiemi si possono individuare la massima e minima variazione positiva, cioè i valori di massimo e minimo nell'insieme V_j^+ e la massima e minima variazione negativa, cioè i valori di massimo e minimo nell'insieme V_j^- , ed ottenere quindi gli intervalli (2) e (3). All'interno della funzione `initMADsTemp(...)` questo procedimento è stato implementato come segue

```

for (int k = 0; k < nSamples; k++) {
    if (airValues[k] <= airValues[j]+e && airValues[k] >= airValues[j]-e){
        if (k != nSamples-1) {
            if (airValues[k+1] > airValues[k]) {
                if (maxPosVariation == 0)
                    maxPosVariation = minPosVariation = airValues[k+1] - airValues[k];
                else if (airValues[k+1] - airValues[k] > maxPosVariation)
                    maxPosVariation = airValues[k+1] - airValues[k];
                else if (airValues[k+1] - airValues[k] < minPosVariation)
                    minPosVariation = airValues[k+1] - airValues[k];
            } else if (airValues[k+1] < airValues[k]) {
                if (maxNegVariation == 0)
                    maxNegVariation = minNegVariation = airValues[k] - airValues[k+1];
                else if (airValues[k] - airValues[k+1] > maxNegVariation)
                    maxNegVariation = airValues[k] - airValues[k+1];
                else if (airValues[k] - airValues[k+1] < minNegVariation)
                    minNegVariation = airValues[k] - airValues[k+1];
            }
        }
    }
}

```

Dove `airValues` rappresenta il vettore *airTemperature* e `nSamples` il numero delle sue componenti, cioè 144.

- Dagli intervalli (2) e (3), viene estratto randomicamente un valore di v tramite il quale si può individuare infine il valore $T_{MAD_{i,j}}$, come in (1), relativo alla temperatura rilevata dal MAD i all'istante t_j .
- Per tenere conto della diversa collocazione, all'interno della stazione di Lecco Maggianico, dei 19 MAD, si applica ai MAD che fanno capo al CMAD North una ulteriore componente r scelta randomicamente all'interno dell'intervallo $[0, 0.5]$. L'espressione che permette di individuare la temperatura rilevata dai MAD North diventa quindi $T_{MAD_{i,j}} = T_{MAD_{i,j-1}} + v - r$.

Effettuando questo procedimento per ogni istante t_j , $1 \leq j < 144$ e per ogni MAD, vengono definiti tutti i valori necessari per ottenere la matrice *MADsTemp*.

Si completa infine il profilo meteorologico calcolando la matrice *MADsHumi*, a partire da *MADsTemp* e il vettore *dewPoint*. Questo calcolo viene effettuato dalla funzione *initMADsHumi(...)* che sfrutta la seguente formula per il calcolo dell'umidità relativa a partire da dew point e temperatura

$$H_{MAD_{i,j}} = e^{\frac{aT_j^{dew}}{b + T_j^{dew}} - \frac{aT_{MAD_{i,j}}}{b + T_{MAD_{i,j}}}}$$

dove $a = 17.27$ e $b = 237.7$ °C.

2.4 Scrittura del profilo meteorologico all'interno del simulatore

Una volta generato, il profilo meteorologico viene scritto direttamente all'interno del file *params_def_in.h*, file dove vengono definiti i valori di input per il simulatore. Il profilo meteorologico deve essere scritto sotto forma di matrici temperature e humidity, matrici con $nMADs+1$ righe e $nSamples$ colonne. I valori necessari per definire queste matrici sono quelli inclusi nelle matrici *MADsTemp*, *MADsHumi* e nei vettori *airTemperature* e *airHumidity*. La funzione che prende in input queste strutture dati e le scrive all'interno del file del simulatore è la funzione *writeWeatherProfile(...)*. Ogni volta che il programma *wpg* viene eseguito, un profilo meteorologico (costruito a partire dai parametri in input al programma, cioè *correctionFactor* e *onlyWinter*) viene inserito nel file *params_def_in.h* aggiungendosi ai profili già presenti ed andando così a costruire una libreria di profili. Tra questi, ne viene scelto uno al momento della compilazione e questo sarà infine il profilo utilizzato per la simulazione.

2.5 Analisi dei risultati

Tramite il programma *wpg* fin qui descritto, è stato possibile originare una libreria di profili con differenti caratteristiche. I dati che li compongono sono stati sintetizzati nei grafici seguenti e confrontati con i grafici ricavati dal modello meteorologico nativo del simulatore, in modo da illustrarne la diversità e la coerenza stessa dei profili generati.

In **Fig. 2** sono riportati i grafici dell'evoluzione nel tempo di temperatura e umidità rilevate dai MAD nel Profilo Meteorologico n. 0, ovvero il profilo nativo del simulatore. In **Fig. 3** e **Fig. 4** sono riportati rispettivamente i grafici dei Profili Meteorologici n. 3 e n. 10, generati automaticamente. Nei grafici di temperatura e umidità, sono indicate in rosso le temperature relative ai MAD del gruppo MAD South e in blu le temperature relative ai MAD del gruppo MAD North. Nei grafici di temperatura viene riportato anche l'andamento del Dew Point, in nero, i cui valori, come descritto precedentemente, sono stati ricavati mediante l'approssimazione di Magnus-Tetens in funzione della temperatura e dell'umidità relativa.

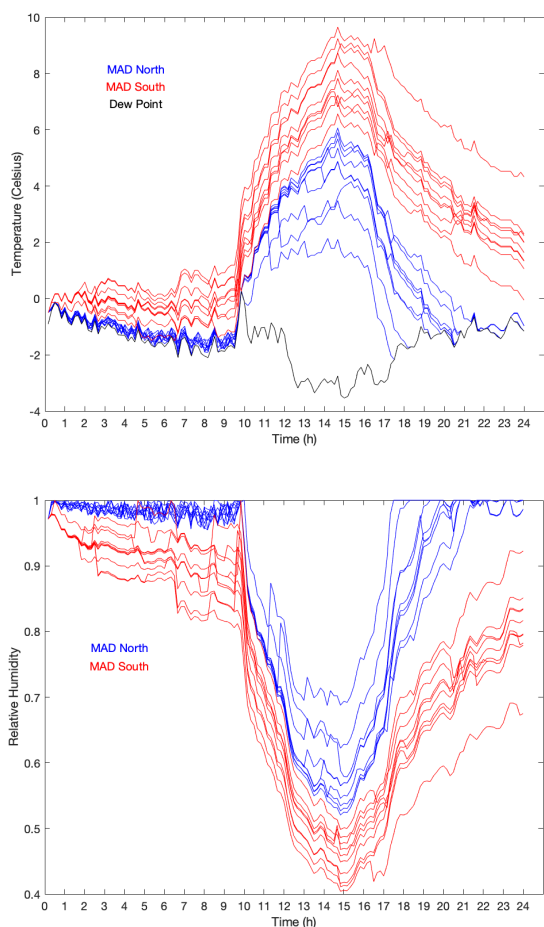


Fig. 2. Profilo Meteorologico n. 0

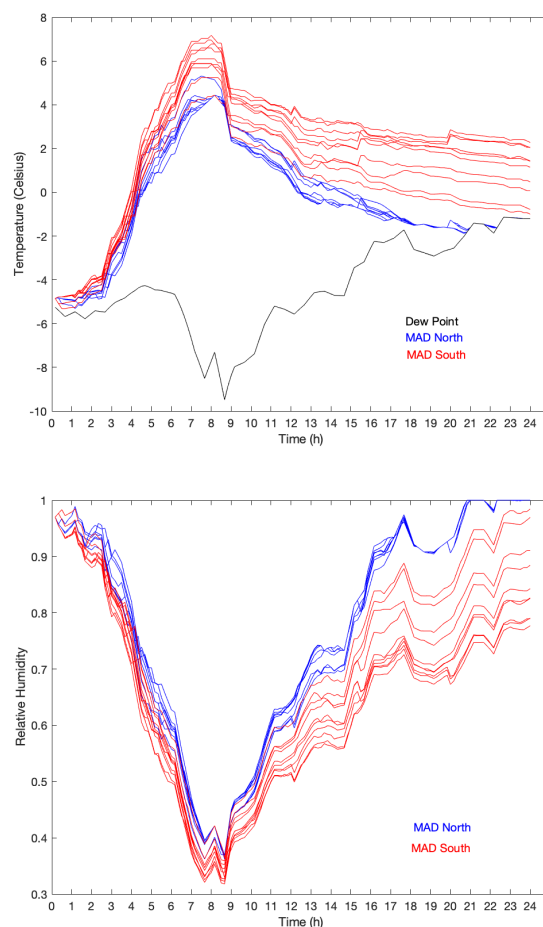


Fig. 3 Profilo Meteorologico n. 3

I profili meteorologici generati avranno caratteristiche diverse l'uno dall'altro, ciò influirà sui risultati delle simulazioni che varieranno in base a tali caratteristiche. La libreria generata permetterà di scegliere liberamente il profilo meteorologico sulla base degli aspetti che si vogliono mettere in risalto con l'esecuzione della simulazione. Ad esempio, se si volesse mettere a dura prova il sistema, si potrebbe scegliere il Profilo Meteorologico n. 3, il quale presenta temperature comprese nell'intervallo $[-5.31667, 7.17018]$. Mentre se si volessero testare le prestazioni del

sistema in condizioni più miti si potrebbe scegliere il Profilo Meteorologico n. 10 con temperature comprese nell'intervallo [4.88939, 10.606].

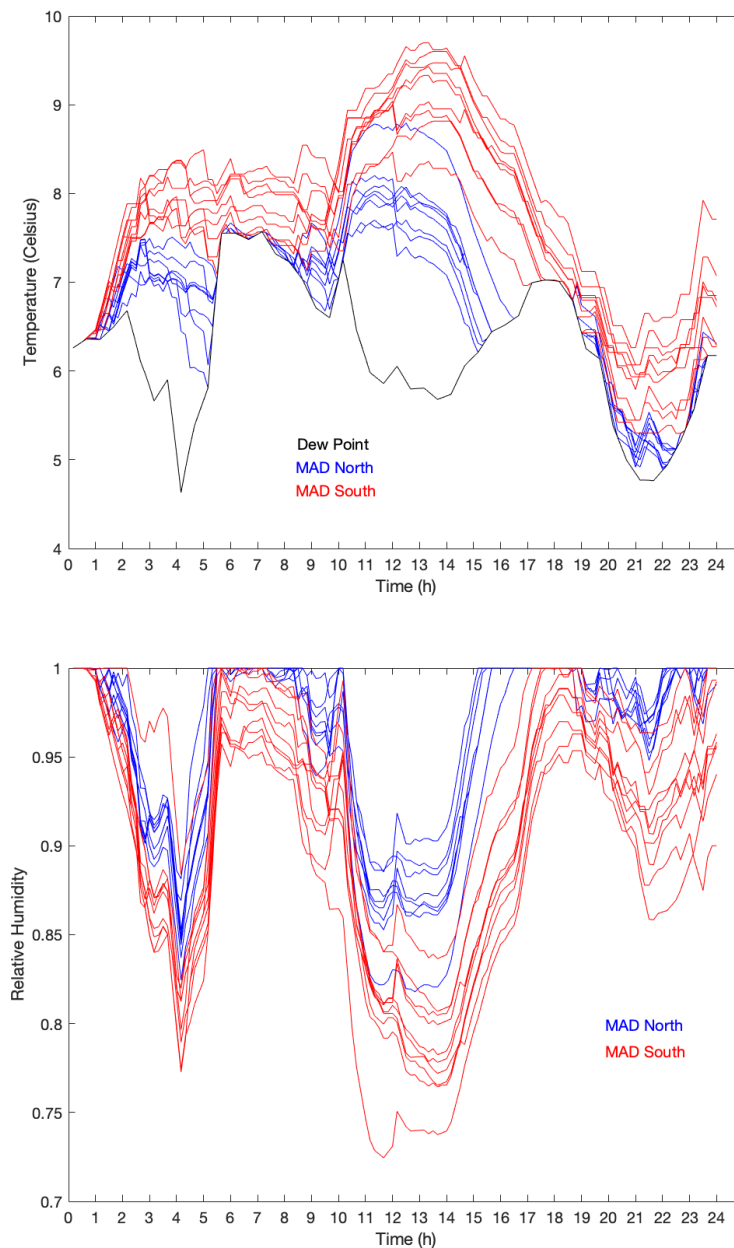


Fig. 4 Profilo Meteorologico n. 10

La **Tabella 1** riassume i risultati ottenuti, espressi in energia (kWh) media consumata per l'accensione e lo spegnimento delle scaldiglie, utilizzando le politiche di gestione native del simulatore ed eseguendo 100, 1.000 e 10.000 tracce di simulazione per ognuno dei profili n. 0, 3 e 10.

Profilo Meteorologico	Tracce di Simulazione	Consumo Medio (kWh)	Varianza
0	100	247,1238	7.842
3	100	242,2923	10.773
10	100	23,9522	4,2062
0	1.000	245,3509	8.257,9
3	1.000	233,5540	10.124
10	1.000	23,9606	4,7404
0	10.000	244,2943	8.366
3	10.000	233,6451	10.323
10	10.000	23,9436	3.9943

Tabella 1 Consumo medio di Energia (kWh) e varianza ricavati dai risultati ottenuti eseguendo 100, 1.000 e 10.000 tracce di simulazione con i profili meteorologici n. 0, 3 e 10.

3 Realizzazione di una politica di regolazione dell'attività delle scaldiglie

3.1 La politica di accensione e spegnimento

Come descritto brevemente nell'Introduzione di questo documento, l'obiettivo secondario del progetto di tesi è quello di sviluppare una politica alternativa per regolare l'attività delle scaldiglie. L'elaborazione e lo studio dei dati necessari per la creazione dei profili meteorologici hanno portato alla naturale intuizione dell'utilizzare la grande mole di informazione contenuta in questi ultimi e sfruttarla per la realizzazione di una nuova politica di gestione delle scaldiglie. I dati meteorologici giocano un ruolo fondamentale nella scelta dei comandi da impartire a questi dispositivi, per questo, conoscere la loro storia passata risulta molto utile per prevedere quella futura e quindi i comandi da impartire in un futuro prossimo.

L'insieme dei dati utilizzati per la creazione dei profili meteorologici può essere visto come una serie storica. Una serie storica è un insieme di variabili casuali, ordinate rispetto al tempo, che esprimono la dinamica di un certo fenomeno nel tempo, in questo caso l'evoluzione dei parametri meteorologici di temperatura e umidità. Analizzando le serie storiche possono essere individuati componenti di trend, di ciclicità e di stagionalità che possono essere utilizzati per prevedere il loro andamento futuro. In questo caso i dati disponibili presentano una forte componente stagionale in quanto ognuno di essi è riconducibile ad una giornata ed ognuna di queste presenta dei tratti comuni, come gli andamenti delle temperature che si sviluppano seguendo lo stesso schema di giornata in giornata. La giornata in questo caso rappresenta la *stagione*. Si può quindi affermare che i dati a disposizione presentino le caratteristiche necessarie per poter essere sfruttati nel prevedere l'andamento futuro dei fenomeni che essi descrivono. I metodi statistici che si possono utilizzare per ricavare informazioni utili dai dati a disposizione hanno comunque dei limiti, essi possono essere sfruttati per ottenere un numero limitato di previsioni dei valori successivi. In questo caso ciò non rappresenta un problema. Infatti, dato un valore di temperatura o umidità all'istante t , sarà necessario prevedere al più il valore di temperatura o umidità relativo all'istante $t+1$.

La previsione del valore futuro di temperatura e umidità andrebbe ad aggiungersi a quelli già presenti, che descrivono la condizione meteorologica delle scaldiglie all'istante corrente. Conoscere il valore futuro di temperatura e umidità, o individuare il trend che queste potrebbero seguire nel breve intervallo di tempo compreso tra l'ultima osservazione e quella successiva, risulta un'informazione molto utile per impartire i comandi alle scaldiglie in maniera intelligente, in modo da evitare possibili sprechi di energia causati, ad esempio, dall'accensione delle scaldiglie nel

momento in cui le temperature si trovano di poco al di sotto della soglia minima consentita e allo stesso tempo esse stanno seguendo un trend positivo che porterà entro pochi istanti la temperatura sopra la soglia minima. Un altro modo per sfruttare la previsione della temperatura successiva per evitare sprechi di energia potrebbe essere quello di spegnere le scaldiglie in maniera anticipata nel caso in cui il valore di temperatura attuale e quello predetto indicassero anche in questo caso un trend positivo, assicurandosi ovviamente che la temperatura all'istante in cui si decide di spegnere la scaldiglia sia già ad un livello tale da rendere impossibile la formazione di ghiaccio sullo scambio.

Vediamo più nel dettaglio come potrebbe essere utilizzato il meccanismo predittivo per realizzare la nuova politica.

Una delle politiche già presenti all'interno del Simulatore prevede l'accensione delle scaldiglie una volta che le temperature scendono sotto i 5°C . La nuova politica abbassa questa soglia a 4°C gestendo le temperature comprese tra i 4°C e i 5°C in maniera diversa. Data una temperatura T , rilevata all'istante t e compresa in questo intervallo, viene eseguita la funzione $\text{predictNext}(T)$ per prevedere la temperatura T' all'istante t' , con $t' > t$. Nel caso in cui le condizioni

- $T' > T$
- $T' > 4.5^{\circ}\text{C}$

risultassero entrambe verificate, la scaldiglia rimarrebbe spenta, altrimenti verrebbe accesa per contrastare la formazione di ghiaccio.

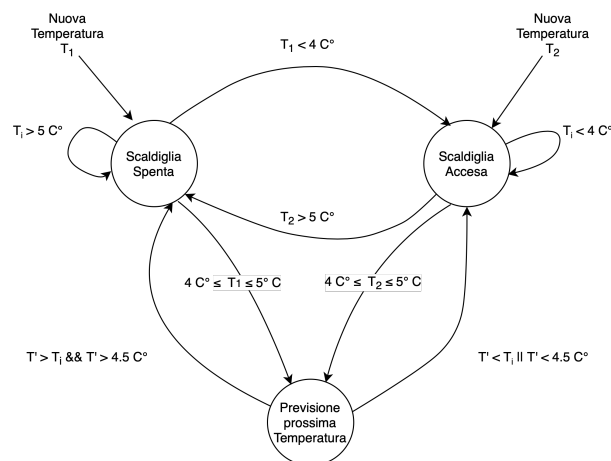


Fig. 5 Macchina a stati finiti che descrive l'attività della nuova politica.

Le temperature vengono predette dalla funzione $\text{predictNext}(\dots)$ che sfrutta un metodo statistico per l'analisi delle serie storiche, il metodo moltiplicativo di Holt-Winters (Triple

exponential smoothing). E' stato scelto questo metodo poiché permette di tenere conto della stagionalità dei dati. Infatti, in ogni insieme di temperature rilevate nell'arco di una giornata si ha un picco nelle rilevazioni effettuate tra le ore 12 e le ore 14. Ciò si ripete per ogni giornata presente nel data set, conferendo al data set stesso caratteristiche stagionali.

3.2 Il Metodo Moltiplicativo di Holt-Winters

Il metodo di Holt-Winters, nella sua versione moltiplicativa o in quella additiva, è un metodo statistico utilizzato per l'analisi di serie storiche i cui dati presentano tratti stagionali che si ripetono con una cadenza costante.

Supponiamo di avere una sequenza di osservazioni $\{y_t\}$, che inizino all'istante $t = 0$ con cicli stagionali di lunghezza L , il metodo di Holt-Winters si articola su tre equazioni:

$$\begin{cases} \ell_t = \alpha \frac{y_t}{s_{n-p}} + (1 - \alpha)(\ell_{t-1} + d_{t-1}) \\ d_t = \beta(\ell_t - \ell_{t-1}) + (1 - \beta)d_{t-1} \\ s_t = \gamma \frac{y_t}{\ell_t} + (1 - \gamma)s_{t-L} \end{cases}$$

La prima definisce il livello e può essere vista come una media ponderata tra il valore osservato y_t e la previsione calcolata al punto precedente $(\ell_{t-1} + d_{t-1})$. La seconda definisce il trend e può essere vista come la media ponderata tra la differenza tra le previsioni al tempo t e al tempo $t - 1$ $(\ell_t - \ell_{t-1})$ e la pendenza calcolata al tempo precedente (d_{t-1}) . Attribuire peso 1 a quest'ultima, cioè porre $(1 - \beta) = 1$, equivarrebbe ad assumere un trend lineare, ovvero una costanza nella pendenza. La terza equazione definisce la componente stagionale s_t , in cui L rappresenta la periodicità della stagione. La componente stagionale ha la funzione di pesare il valore del trend in base a dove l'istante considerato si colloca all'interno del ciclo stagionale. Sono presenti tre fattori di livellamento compresi tra 0 e 1, α , β e γ , il cui valore può essere stimato minimizzando i quadrati degli scarti, ovvero trovando i valori di α , β e γ per cui la funzione

$$RSS = \sum_{i=1}^n (y_i - F_i)^2$$

assume valore minimo. Dove F_i rappresenta la previsione, calcolata all'istante $i - 1$, per il valore osservato all'istante i , cioè y_i , e può essere calcolata come:

$$F_{t+m} = (\ell_t + m d_t) s_{t-L+1+(m-1) \bmod L}$$

Dove F_{t+m} rappresenta la stima del valore di y osservato all'istante $t + m$, $m > 0$ basata sui dati originali fino all'istante t .

3.3 Inizializzazione del metodo

I valori iniziali di livello, trend e componente stagionale (ℓ_0 , d_0 e s_0) devono essere inizializzati per permettere il calcolo dei valori successivi. Per fare ciò esistono diverse tecniche, quella scelta per l'implementazione della politica è quella che pone, per livello e trend:

$$\begin{aligned} \ell_0 &= y_0 \\ d_0 &= \frac{1}{L} \left(\frac{y_{L+1} - y_1}{L} + \frac{y_{L+2} - y_2}{L} + \dots + \frac{y_{L+L} - y_L}{L} \right) \end{aligned}$$

mentre per il vettore delle componenti stagionali si pone:

$$\begin{aligned} s_i &= \frac{1}{N} \sum_{j=1}^N \frac{y_{L(j-1)+i}}{A_j} \quad \forall i = 1, 2, \dots, L \\ A_j &= \frac{\sum_{i=1}^L y_{L(j-1)+i}}{L} \quad \forall j = 1, 2, \dots, N \end{aligned}$$

Dove N è il numero di cicli stagionali completi all'interno dei dati, ed A contiene, per ogni ciclo stagionale, la media dei valori osservati.

3.4 Implementazione del modello

La creazione dei profili meteorologici e l'implementazione del modello per la previsione delle temperature fin qui descritto, si incontrano nell'implementazione, in quanto i dati su cui dovrà essere applicato il modello durante le simulazioni saranno proprio quelli generati durante la creazione del profilo meteorologico. In quest'ultimo, come descritto in 2.2, vengono inseriti soltanto i dati relativi ad una giornata, ci sono però alcune componenti del modello statistico che acquisiscono maggiore accuratezza se i loro valori iniziali sono calcolati utilizzando tutti i dati disponibili nel data set. Questo è il caso delle componenti stagionali, che se calcolate soltanto

utilizzando i dati inseriti nel profilo meteorologico vengono influenzate da un solo ciclo stagionale degli N disponibili. Per questo, durante la fase di elaborazione dei dati per il profilo meteorologico, viene aggiunta una nuova fase che si occupa di creare il vettore delle componenti stagionali. Questo avviene una volta che il data set originale è stato importato ed elaborato a seconda dei valori delle variabili globali *onlyWinter* e *correctionFactor* producendo il data set finale dal quale verranno estrapolati i dati per il profilo meteorologico. Questo data set finale manterrà la proprietà di stagionalità di quello iniziale e verrà utilizzato per produrre le componenti stagionali utilizzate nell'applicazione del modello di Holt-Winters.

L'inizializzazione delle componenti stagionali viene effettuata al momento della creazione del profilo meteorologico attraverso la funzione `initSeasonalComponents(...)`, implementata anch'essa nel programma *wpg*. Come le altre componenti di ogni profilo meteorologico, questa viene scritta all'interno del file *params_def_in.h* attraverso la funzione `writeWeatherProfile(...)` e diventa quindi accessibile al simulatore. Questo vettore rimarrà statico e non verranno calcolate nuove componenti in quanto quelle già presenti possono essere sfruttate per il calcolo di L previsioni, cioè per predire le temperature di una giornata intera. Poiché, all'interno del simulatore, le temperature e gli altri valori meteorologici si sviluppano per la durata di una sola giornata, sarebbe inutile produrre nuove componenti stagionali per poi utilizzarle nella predizione di valori meteorologici delle giornate successive, in quanto questo non sarebbero utilizzate nella simulazione. Quindi, una volta calcolato il vettore *wp_seasonalComponents*, relativo alle componenti stagionali di un determinato profilo meteorologico *wp*, esso rimane costante per tutta la durata della simulazione. In seguito è riportata l'implementazione del calcolo del vettore A e del vettore delle componenti stagionali.

```
for (int j = 0; j < nCycles; j++) {  
    sum = 0;  
    for (int i = 0; i < seasonLenght; i++)  
        sum = sum + data[seasonLenght*j+i];  
    sum = sum / seasonLenght;  
    avg[j] = sum;  
}
```

Codice 2 Calcolo del vettore A .

```

for (int i = 0; i < seasonLenght; i++) {
    sum = 0;
    for (int j = 0; j < nCycles; j++)
        sum = sum + data[seasonLenght*j+i] / avg[j];
    seasonalComponents[i] = sum / nCycles;
}

```

Codice 3 Calcolo del vettore seasonalComponents.

Come specificato nel paragrafo precedente, il valore iniziale del livello ℓ_0 assume il valore della prima delle osservazioni disponibili y_0 .

L'inizializzazione del trend necessita dei valori delle ultime due stagioni, quindi delle ultime due giornate. Questi dati non sono inclusi nel profilo meteorologico, per questo motivo vengono aggiunti due ulteriori vettori oltre a quello relativo alle componenti stagionali, i vettori *pastDaysT* e *pastDaysH*. Queste informazioni, scritte insieme al profilo meteorologico nel file *params_def_in.h*, vengono sfruttate all'interno delle simulazioni per inizializzare il trend attraverso la funzione *initialTrend(...)*.

Le informazioni necessarie all'inizializzazione del modello sono quindi generate con il profilo meteorologico e messe a disposizione del simulatore che ne farà uso in caso le temperature rilevate dai MAD in un preciso istante si trovassero nell'intervallo $[4^{\circ}\text{C}, 5^{\circ}\text{C}]$. In caso esse si trovassero in questo intervallo, per fare in modo che le scaldiglie si comportino come descritto nel paragrafo 3.1, sono state modificate due funzioni già presenti all'interno del file *SAN_functions.h* del simulatore, le funzioni *Alg2LogicOn(T* TemperatureAtMAD)* e *Alg2LogicOff(T* TemperatureAtMAD)*. Il comportamento di queste due funzioni è stato modificato in modo da invocare la funzione *predictNext(...)* in caso le temperature si trovassero nell'intervallo sopracitato.

La funzione *predictNext(...)* si occupa, prima di tutto, di inizializzare livello, trend e componenti stagionali sfruttando i dati e le funzioni descritte precedentemente. Successivamente, all'interno della stessa funzione sono implementate le equazioni del modello per il calcolo dei nuovi valori di livello e trend, sfruttati poi per il calcolo della predizione. In seguito è riportato il codice dell'implementazione.

```

level = smoothing_factor*(tempAtMAD)/seasonalComponents[tm+1] +
        (1-smoothing_factor)*(level+trend);
trend = trend_factor*(level - lastLevel) + (1-trend_factor)*trend;
prediction = (level + trend)*seasonalComponents[tm+1];

```

Codice 4 Calcolo di level, trend e valore della previsione

Dove `smoothing_factor` e `trend_factor` sono rispettivamente α e β , la variabile `tm` rappresenta il tempo trascorso all'interno della simulazione e `tempAtMAD` rappresenta l'ultima osservazione della temperatura sullo specifico MAD. Mentre `lastLevel` è il livello relativo al passo precedente. Come specificato precedentemente, la terza equazione relativa al calcolo dei nuovi componenti stagionali non è stata implementata poiché non necessaria in questo caso. La previsione viene calcolata sommando `level` e `trend` e moltiplicando il risultato per il componente stagionale relativo all'istante per cui sto effettuando la previsione. Se le temperature dei MAD sono state osservate fino all'istante `tm`, la previsione si occuperà di stimare la temperatura all'istante `tm+1`.

3.5 Stima dei parametri

I parametri α e β sono stati stimati sfruttando tutti i dati disponibili. Si è trattato l'intero data set come una serie storica di $N = \text{dimDataset}/L$ cicli, dove `dimDataset` è il numero di rilevazioni totali contenute nel data set ed L è il numero di rilevazioni contenute in una giornata, ovvero la dimensione di una stagione. Inizializzati `level`, `trend` e `seasonal components`, e partendo dalla prima delle temperature, si sono calcolate le previsioni per ogni temperatura contenuta nel data set e per ogni coppia di (*valoreOsservato*, *previsione*) si è calcolato l'errore espresso come quadrato della differenza tra le due componenti. La somma di questi errori, per ogni coppia, equivale alla funzione RSS da minimizzare per ottenere i valori ottimali dei parametri. Effettuando molteplici iterazioni di questo processo, ognuna delle quali con valori di α e β diversi, si sono individuati i valori di questi parametri per cui RSS assume i valori minori. Questi valori sono $\alpha = 0.9781$ e $\beta = 0.2458$.

3.6 Analisi dei risultati

Al fine di testare l'efficacia della nuova politica di gestione delle scaldiglie, sono state effettuate diverse simulazioni variando di volta in volta il profilo meteorologico utilizzato e confrontando i risultati, in termini di consumo medio di energia, ottenuti con le politiche standard e con la nuova politica basata sulla previsione delle temperature. Ad esempio, prendendo in

considerazione il Profilo Meteorologico n. 3, di cui sono riportati in Fig. 3 gli andamenti di temperatura e umidità, e tenendo fisso il valore del numero di misurazioni $N = 10.000$, si è effettuata una analisi delle frequenze in modo da capire quali siano le fasce di consumo energetico in cui si collocano le misurazioni rilevate con la politica standard e con la politica predittiva. All'interno del Simulatore è possibile selezionare diversi “studi” per l'esecuzione della simulazione, essi variano l'uno dall'altro per il valore di due parametri, c e w , che determinano il *failure rate* del canale di comunicazione. Per testare il comportamento della nuova politica si è fatto uso dello studio numero 0 e dello studio numero 40. In seguito sono riportati i grafici delle analisi delle frequenze ottenuti a partire dal Profilo Meteorologico n. 3.

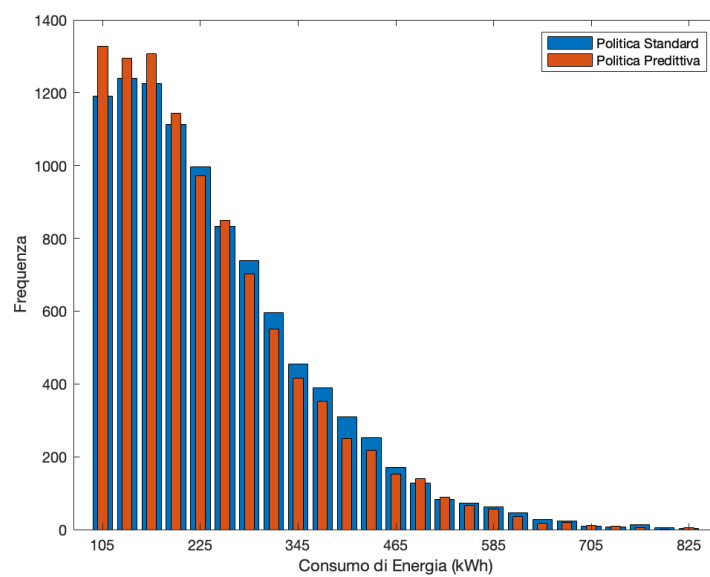


Fig. 6 Analisi delle frequenze effettuata utilizzando lo “studio” 0 con $N = 10.000$ e il Profilo Meteorologico 3.

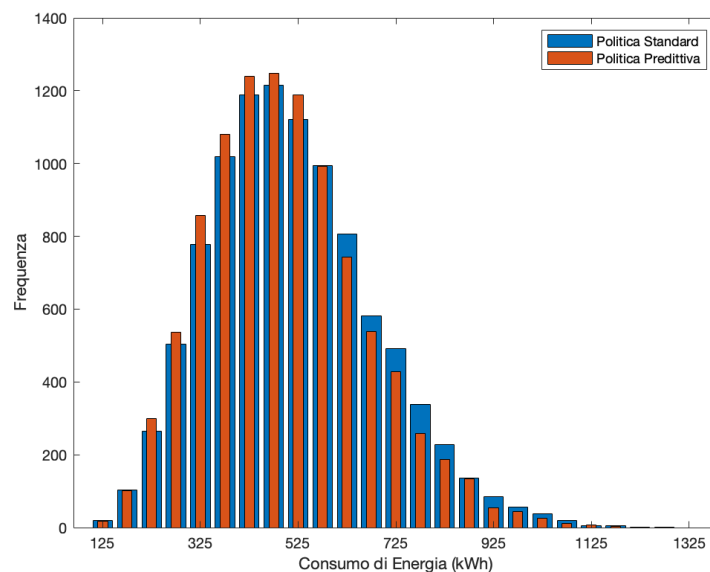


Fig. 7 Analisi delle frequenze effettuata utilizzando lo “studio” 40 con $N = 10.000$ e il Profilo Meteorologico 3.

Da questi risultati si può notare come le misurazioni relative alle simulazione lanciate con la Politica Predittiva, si collochino maggiormente nelle fasce di consumo più basse, mentre quelle relative alla Politica Standard sono in maggioranza nelle fasce di consumo più alte.

Per dimostrare ulteriormente la correttezza dei risultati sono stati effettuati altri test, questa volta sfruttando il Profilo Meteorologico 1, un profilo con temperature medie più alte rispetto al quello precedentemente utilizzato, di cui per completezza sono riportati anche i grafici di temperatura e umidità.

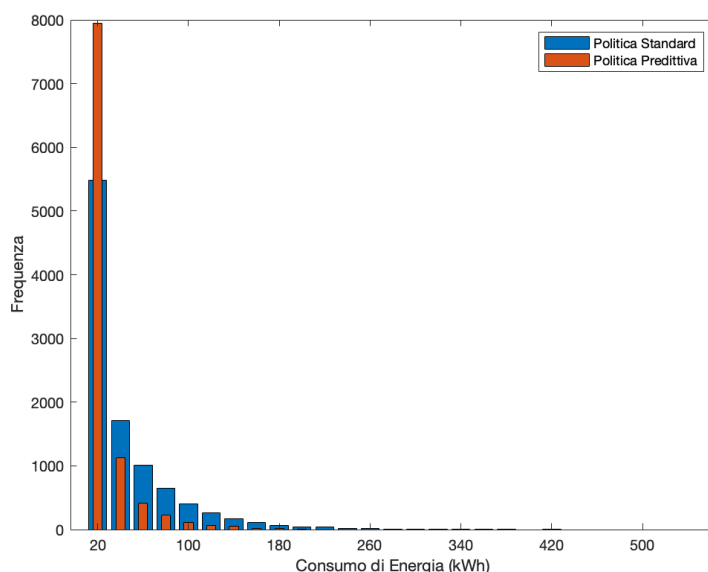


Fig. 8 Analisi delle frequenze effettuata utilizzando lo “studio” 0 con N = 10.000 e il Profilo Meteorologico 1.

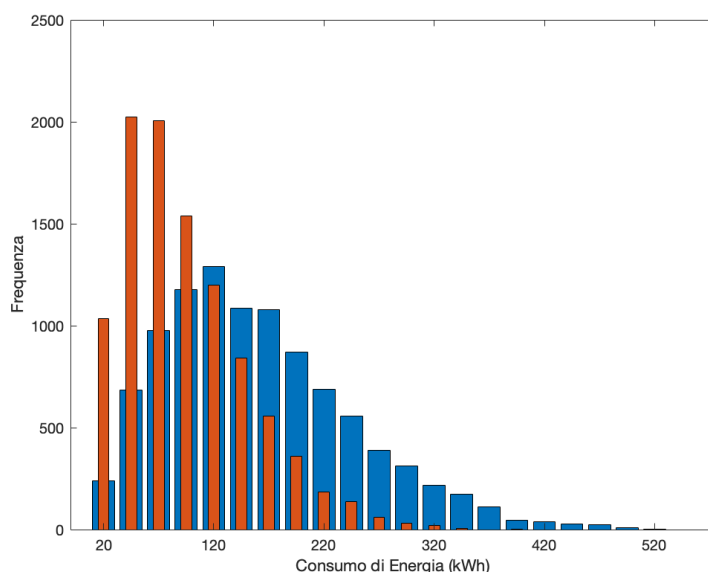


Fig. 9 Analisi delle frequenze effettuata utilizzando lo “studio” 40 con N = 10.000 e il Profilo Meteorologico 1.

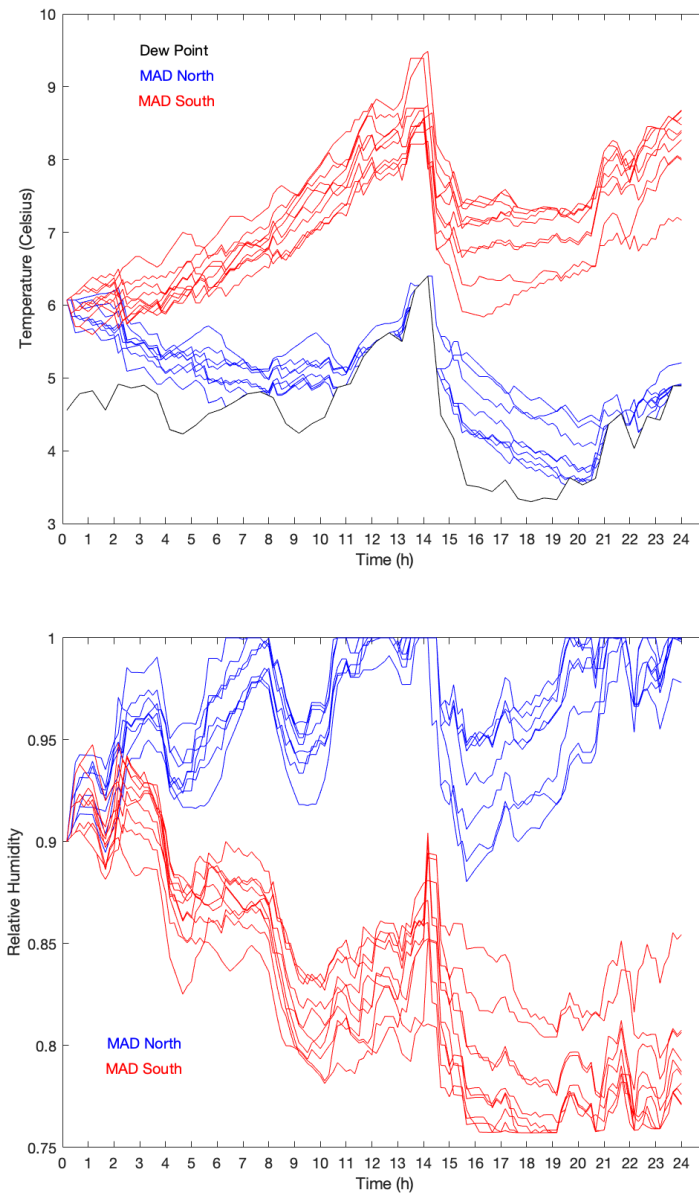


Fig. 10 Profilo Meteorologico 1

Anche in questo caso si può notare come il numero di misurazioni con valori di consumo energetico appartenente a fasce basse sia maggiore quando si utilizza la politica predittiva. Nella seguente tabella sono riportati i valori di consumo medio, ottenuto nelle varie combinazioni di simulazione, e il risparmio medio generato dell'uso della politica predittiva.

Profilo Meteorologico	Studio	Consumo Medio (kWh) con Politica Standard	Consumo Medio (kWh) con Politica Predittiva	Risparmio Medio %
1	0	52.8222	34.7437	34.22%
	40	172.5690	106.2818	38.41%
3	0	233.6451	226.9259	2.87%
	40	537.9057	523.9839	2.59%

Applicando la politica predittiva al Profilo Meteorologico 1 si ha un notevole risparmio. Ciò è dovuto al fatto che le temperature di questo profilo, come mostrato in Fig. 9, si trovano per lo più al di sopra della soglia dei 5°C (che esclude di default la scelta dell'accensione), oppure all'interno della fascia [4°C,5°C] in cui la decisione viene presa facendo uso della previsione. I casi in cui le scaldiglie vengono accese direttamente senza l'ausilio di una analisi predittiva, ovvero i casi in cui le temperature si trovano sotto i 4°C, sono relativamente pochi, per questo i consumi si mantengono bassi.

Eseguendo le simulazioni in modalità di debug è possibile osservare il processo decisionale che porta all'accensione o meno della scaldiglia *SH*.

```
...
Weather change at Time 530
POLICY BASED ON PREDICTION IS RUNNING:
LastTempAtMAD  NextTempPrediction
4.88919    4.93507
SH will not be switched on
POLICY BASED ON PREDICTION IS RUNNING:
LastTempAtMAD  NextTempPrediction
4.96854    5.01497
SH will not be switched on
...
```

In questo esempio si può notare come il sistema, ottenendo l'ultimo dato di temperatura disponibile, cioè LastTempAtMAD, riesca a produrre una previsione tramite la quale è in grado di decidere se accendere o meno la scaldiglia *SH*. In questo caso i valori di temperatura predetti soddisfano le condizioni descritte in 3.1 e viene quindi presa la decisione di non accendere le scaldiglie portando a un risparmio nel consumo di energia.

4 Conclusioni

Tramite l'implementazione del programma C++ *wpg* è stato possibile creare una libreria di profili meteorologici, con caratteristiche diverse l'uno dall'altro, utilizzabili all'interno del Simulatore di Sistemi Ferroviari in modo da migliorarne le prestazioni, differenziando le condizioni meteorologiche tra una simulazione e l'altra, e permettendo analisi più accurate per le politiche volte al miglioramento dell'efficienza energetica.

Con la nuova politica di gestione delle scaldiglie basata sulla previsione delle temperature si è proposto un approccio alternativo alla gestione di questi dispositivi. Questo tipo di tecnica può trarre beneficio da un contesto IoT come quello della Smart Station proposta dal progetto STINGRAY e i risultati ottenuti da questo progetto di tesi dimostrano che potrebbe essere un approccio utile per ridurre in maniera intelligente i consumi di energia.

Riferimenti

1. S. Chiaradonna, F. Di Giandomenico, G. Masetti, D. Basile.: A refined framework for model-based assessment of energy consumption in the railway sector.
2. G. Masetti, G. Spagnolo, C. Vallati, G. Anastasi, A. Bechini, C. Vincenzo, S. Chiaradonna, F. Di Giandomenico, S. Gnesi.: STINGRAY: SmarT station INtelliGent RAILwaY. (2019)
3. Lawrence, M.G.: The relationship between relative humidity and the dew point temperature in moist air: A simple conversion and applications. Bulletin of the American Meteorological Society 86(2), 255-234 (2005)
4. Kalehar, Prajakta S.: Time series Forecasting using Holt-Winters Exponential Smoothing.
5. R.J. Hyndman.: Initializing the Holt-Winters method.