

Ricerca di Sistema elettrico



Implementazione e testing di modelli per la valutazione dell'adeguatezza di reti elettriche in funzione dell'affidabilità del sistema e dei suoi componenti

Roberto Ciavarella, Maria Valenti

Implementazione e testing di modelli per la valutazione dell'adeguatezza di reti elettriche in funzione dell'affidabilità del sistema e dei suoi componenti.

R. Ciavarella (ENEA), M. Valenti (ENEA)

Dicembre 2024

Report Ricerca di Sistema Elettrico

Accordo di Programma Ministero dell'Ambiente e della Sicurezza Energetica -ENEA Piano Triennale di Realizzazione 2022-2024

Obiettivo 2: Digitalizzazione ed evoluzione delle reti

Progetto 2.3: Evoluzione, pianificazione, gestione ed esercizio delle reti elettriche

Linea di attività: 1.5

Responsabile del Progetto: Maria Valenti

Responsabile Linea di Attività: Roberto Ciavarella

Mese inizio previsto: luglio 2023

Mese inizio effettivo: luglio 2023

Mese fine previsto: dicembre 2024

Mese fine effettivo: dicembre 2024

Indice

1	Risultati attesi	4
2	Risultati ottenuti.....	4
2.1	Avanzamento della ricerca rispetto allo stato dell'arte internazionale con riferimento ai risultati ottenuti.....	4
3	Prodotti attesi	5
4	Prodotti ottenuti.....	5
5	Analisi degli scostamenti su attività e risultati.....	6
6	Sintesi delle attività svolte	6
7	Dettaglio delle attività svolte.....	7
7.1	Adequacy tool: definizione della metodologia per la valutazione dell'adeguatezza	7
7.2	Adequacy tool: definizione del modello e codifica in Python	9
7.3	Adequacy tool: realizzazione del modulo per l'integrazione in ARS tool	10
7.4	Adequacy tool: simulazioni su base caso studio	11
7.5	Optimal Storage Tool: realizzazione del modulo per l'integrazione in ARS tool	11
8	Contributo delle eventuali consulenze alle attività sopra descritte.....	13
9	Pubblicazioni scientifiche.....	13
10	Eventi di disseminazione	14
11	Descrizione dei risultati ottenuti	14
11.1	Adequacy Tool: codice Python per l'implementazione del modello di valutazione dell'adeguatezza in funzione dell'affidabilità e delle anomalie di componente e corrispondente software opensource.....	14
11.2	Adequacy tool: risultati delle simulazioni su base caso studio	24

Indice delle figure

Figura 1: Approccio probabilistico proposto da ENTSO-E.....	7
Figura 2: Rappresentazione schematica della metodologia proposta.....	9
Figura 3: Schema di integrazione con tool della LA1.15.....	10
Figura 4: Modello concettuale dell'Optimal Storage Tool definito nella LA1.4	12
Figura 5: Schermata di output dell'Optimal Storage tool	13
Figura 6: Rete test	24
Figura 7: Affidabilità dei componenti e di fornitura	31
Figura 8: Adeguatezza del sistema nei tre casi studio	32

Indice delle Tabelle

Tabella 1: Nodi AC	25
Tabella 2: Trasformatori.....	25
Tabella 3: Linee AC	26
Tabella 4: Carichi AC	26
Tabella 5: PWM	27
Tabella 6: Nodi DC.....	27
Tabella 7: Carichi DC	28
Tabella 8: Batterie DC	28
Tabella 9: Convertitori DC-DC.....	28
Tabella 10: Fotovoltaico DC-DC.....	29
Tabella 11: Eolici DC.....	29
Tabella 12: Eolici DC	29
Tabella 13: Linee DC	29
Tabella 14: Valori di LOLE e EENS nei 3 casi studio.....	30

1 Risultati attesi

I risultati attesi previsti da capitolato vigente sono:

- implementazione e testing del modello sviluppato per la valutazione dell'adeguatezza di sistemi elettrici ad alta penetrazione di risorse distribuite, anche di tipo rinnovabile non programmabile, tenendo conto dell'affidabilità della fornitura o di componente.
- Implementazione del modello definito nella LA1.4

2 Risultati ottenuti

Le attività svolte nella presente LA hanno riguardato:

- la definizione e lo sviluppo di un modello per la valutazione dell'adeguatezza applicabile anche su scala locale in grado di considerare aspetti come l'affidabilità della fornitura e di componente.
- Implementazione del codice di integrazione in ARS tool del modello sviluppato nella LA1.4.

Con riferimento al primo risultato, si evidenzia che la metodologia proposta tiene conto, non solo dell'affidabilità, come previsto da capitolato, ma anche di possibili comportamenti anomali dei componenti di rete. Il risultato ottenuto, pertanto, rappresenta un risultato non previsto dal capitolato,

I risultati ottenuti, oltre a soddisfare pienamente gli obiettivi attesi, potranno comportare benefici per il sistema elettrico nazionale e quindi per i suoi utenti. Infatti, nei sistemi energetici futuri, la generazione distribuita, anche di tipo rinnovabile non programmabile, dovrà contribuire in percentuali sempre più significative all'adeguatezza di sistema che potrà anche divenire una specifica da garantire a livello locale. In tale scenario risulta cruciale sviluppare nuovi modelli per la valutazione dell'adeguatezza che siano applicabili anche su scala locale e che tengano conto di aspetti come l'affidabilità della fornitura o di componente e dei possibili comportamenti anomali dei componenti di rete come richiamato nel §2.1.

2.1 Avanzamento della ricerca rispetto allo stato dell'arte internazionale con riferimento ai risultati ottenuti

Molti articoli in letteratura propongono metodologie per valutare l'adeguatezza del sistema elettrico. Mentre le incertezze legate all'utilizzo delle energie rinnovabili sono generalmente considerate in quasi tutti i lavori, nessuna delle metodologie proposte considera aspetti connessi al comportamento anomalo dei componenti di rete. Come evidenziato nel rapporto

tecnico RdS_PTR 22-24_PR 2.3_LA1.3_154, un'anomalia è un'osservazione o una sequenza di osservazioni che si discostano in modo pronunciato dalla distribuzione generale dei dati considerati, formando solo una piccola parte dei dati considerati (Braei & Wagner, 2020). Nelle reti e microreti elettriche, diversi componenti possono manifestare comportamenti anomali, ovvero, nelle serie temporali delle variabili monitorate (es.: potenza prodotta, corrente, tensione, ecc.), una porzione dei dati osservati può variare rispetto alla sua sequenza generale. Tenere conto anche di questa piccola porzione di dati osservati anomali può contribuire a ottenere valori di adeguatezza di sistema più rappresentativi delle condizioni reali di funzionamento.

Infine, sebbene le metodologie di calcolo attualmente più utilizzate per la valutazione dell'adeguatezza siano efficaci negli attuali sistemi energetici, in cui l'adeguatezza è assicurata principalmente dalle fonti termoelettriche tradizionali, potrebbero non essere adeguate per gli scenari futuri. Infatti, se oggi, le fonti termoelettriche contribuiscono a garantire l'adeguatezza del sistema in misura quasi dieci volte superiore rispetto alle fonti rinnovabili variabili, negli scenari energetici futuri, la situazione potrebbe invertirsi, soprattutto a livello locale. Pertanto, occorrono metodologie che tengano conto anche delle risorse distribuite a livello locale e non prevalentemente delle risorse che operano a livello di trasmissione.

In tali premesse, la metodologia proposta nella LA1.5 rappresenta un avanzamento rispetto allo stato dell'arte in quanto consente sia di valutare l'adeguatezza delle risorse distribuite appartenenti a porzioni di rete locali sia perché tiene conto dell'affidabilità e di possibili comportamenti anomali delle risorse di rete.

3 Prodotti attesi

I prodotti attesi, come da capitolato, per la LA1.5 sono:

- Un software opensource per la valutazione dell'adeguatezza in funzione dell'affidabilità dei componenti di rete tenendo conto anche dei loro possibili comportamenti anomali. Il software prodotto è stato integrato in ARS tool nell'ambito della LA1.15.
- Un software per la valutazione dei sistemi di accumulo e delle FER per il supporto dell'adeguatezza di sistemi elettrici decarbonizzati basato sul modello e sul codice sviluppato nella LA1.4 e integrato in ARS tool (LA1.15).

4 Prodotti ottenuti

I prodotti ottenuti nell'ambito della LA1.5 sono:

- Un software opensource per la valutazione dell'adeguatezza in funzione dell'affidabilità dei componenti di rete tenendo conto anche dei loro possibili comportamenti anomali. Il software prodotto è stato integrato in ARS tool nell'ambito della LA1.15 e nel prosieguo del documento verrà identificato con il nome "Adequacy tool".
- Un software per la valutazione dei sistemi di accumulo e delle FER per il supporto dell'adeguatezza di sistemi elettrici decarbonizzati basato sul modello e sul codice sviluppato nella LA1.4. Il software prodotto è stato integrato in ARS tool nell'ambito della LA1.15 e nel prosieguo del documento verrà identificato con il nome "Optimal Storage tool".

5 Analisi degli scostamenti su attività e risultati

Non si evidenziano scostamenti tecnici/economici sull'attività svolta né sui risultati previsti da capitolato.

6 Sintesi delle attività svolte

Le attività svolte possono essere suddivise in cinque fasi distinte.

- La prima fase ha riguardato la definizione della metodologia per la valutazione dell'adeguatezza di un sistema elettrico applicabile anche su scala locale e in grado di considerare aspetti come l'affidabilità della fornitura e di componente ed i comportamenti anomali dei componenti di rete.
- La seconda fase ha riguardato la definizione del modello corrispondente alla metodologia definita e la relativa codifica Python.
- La terza fase è stata focalizzata sullo sviluppo del software e, in particolare sulla realizzazione del modulo di integrazione del modello (codici Python) con il software ARS tool. Tale fase ha previsto la progettazione e la realizzazione dell'interfaccia di input da parte degli utenti e dell'interfaccia di visualizzazione dei risultati.
- La quarta fase ha riguardato la simulazione di 3 casi studio mediante il software. Fissata la rete test, nel primo caso studio, è stata calcolata l'adeguatezza del sistema senza considerare comportamenti anomali e affidabilità; nel secondo caso studio, si è tenuto conto dell'affidabilità ma non di anomalie di comportamento dei componenti; nel terzo caso studio, sono stati considerati sia i comportamenti anomali che l'affidabilità.
- Nella quinta fase, acquisito il codice Python per la valutazione dei sistemi di accumulo e delle FER per il supporto dell'adeguatezza di sistemi elettrici decarbonizzati dalla LA1.4, si è proceduto allo sviluppo software del modulo di integrazione nel software ARS tool. In particolare, è stata effettuata la progettazione e la realizzazione dell'interfaccia

di input da parte degli utenti e l'implementazione dell'interfaccia di visualizzazione dei risultati.

7 Dettaglio delle attività svolte

Come evidenziato nella sezione 6 del documento, le attività della presente LA possono essere distinte in cinque fasi diverse. Per ciascuna di esse, di seguito è riportato il dettaglio delle attività svolte.

7.1 Adequacy tool: definizione della metodologia per la valutazione dell'adeguatezza

Al fine di definire una metodologia per la valutazione dell'adeguatezza di un sistema elettrico applicabile anche su scala locale in grado di considerare aspetti come l'affidabilità della fornitura e di componente ed i comportamenti anomali dei componenti di rete, si è fatto riferimento alla metodologia ENTSO-E basata su approccio probabilistico che viene considerata come riferimento per il calcolo dell'adeguatezza delle reti elettriche a livello Nazionale. Tale metodologia è mostrata in figura 1.

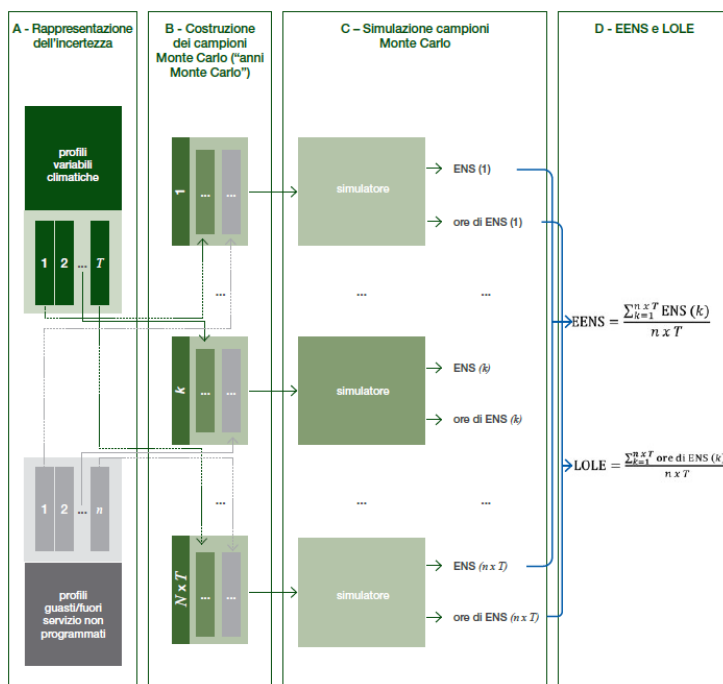


Figura 1: Approccio probabilistico proposto da ENTSO-E¹

¹ Fonte: European Resource Adequacy Assessment 2022 Edition

La metodologia proposta da ENTSO-E è adatta allo scenario attuale in cui le fonti termoelettriche tradizionali mediamente forniscono un contributo all'adeguatezza quasi 10 volte più elevato rispetto a quello fornito dalle fonti rinnovabili non programmabili². Tuttavia, come già evidenziato in precedenza, nei futuri scenari energetici la generazione distribuita a livello locale contribuirà in maniera sempre più importante a garantire l'adeguatezza del sistema elettrico. Per tale motivo, è necessario adottare nuove metodologie applicabili anche su scala locale. In tali premesse, nell'ambito della presente LA è stata definita una metodologia per la valutazione dell'adeguatezza, rappresentata in maniera schematica in figura 2. In particolare, la valutazione dell'adeguatezza viene condotta applicando il metodo Monte Carlo denominato State Sampling. Nello specifico, per valutare l'adeguatezza del sistema, vengono generati in maniera casuale M "Anni Monte Carlo".

Ogni anno Monte Carlo è suddiviso in 8760 stati del sistema estratti in maniera casuale. Uno stato casuale del sistema è ottenuto estraendo casualmente un'ora (h-esima ora) nell'orizzonte temporale di analisi e considerando, in corrispondenza dell'h-esima ora estratta, le informazioni sull'affidabilità di componente e di fornitura (quadrato in basso a sinistra nella Figura 2), anomalie dei componenti (quadrato in basso a destra nella Figura 2), previsioni di generazione e carico (quadrato in alto a sinistra nella Figura 2).

Più nello specifico:

- A.1.** per ciascuno stato, il modello implementato esegue le seguenti operazioni:
1. Legge le informazioni di input: previsioni a lungo termine per la domanda e la generazione ottenute mediante i modelli di predizione delle risorse di generazione e carico definiti nella LA1.2; affidabilità di componente e di fornitura valutata mediante la funzione già sviluppata per il PTR 2019-2021 progetto 2.7; profili anomali annuali dei componenti ottenuti dai modelli sviluppati nella LA1.3 di progetto.
 2. Estrae casualmente la h-esima ora nell'orizzonte temporale di analisi.
 3. Genera un numero casuale U contenuto nella distribuzione uniforme $[0, 1]$. In corrispondenza dell'h-esima ora, per ciascun carico presente nel sistema elettrico in esame, viene valutata la copertura della domanda in funzione della affidabilità della fornitura. Nello specifico, se U risulta essere maggiore o uguale al valore dell'affidabilità di fornitura del carico, il carico corrispondente è considerato non coperto.
 4. In corrispondenza della h-esima ora, individua i comportamenti anomali dei componenti di rete in base ai loro profili di anomalia.
 5. In corrispondenza della h-esima, valuta le probabili ore di mancata fornitura della domanda (LOLE) e l'energia attesa non fornita (EENS).
- A.2.** Il modello ripete le suddette operazioni per 8760 volte (stati del sistema) e calcola il LOLE e l'EENS medio per l'anno Monte Carlo generato.
-

A.3. Il modello ripete le suddette operazioni per gli M-anni Monte Carlo considerati nel periodo di riferimento e calcola il LOLE e l'EENS medio per il periodo analizzato (valor medio dei LOLE e degli EENS degli anni Monte Carlo ottenuti seguendo i passi precedenti come da formula riportata in Figura 2).

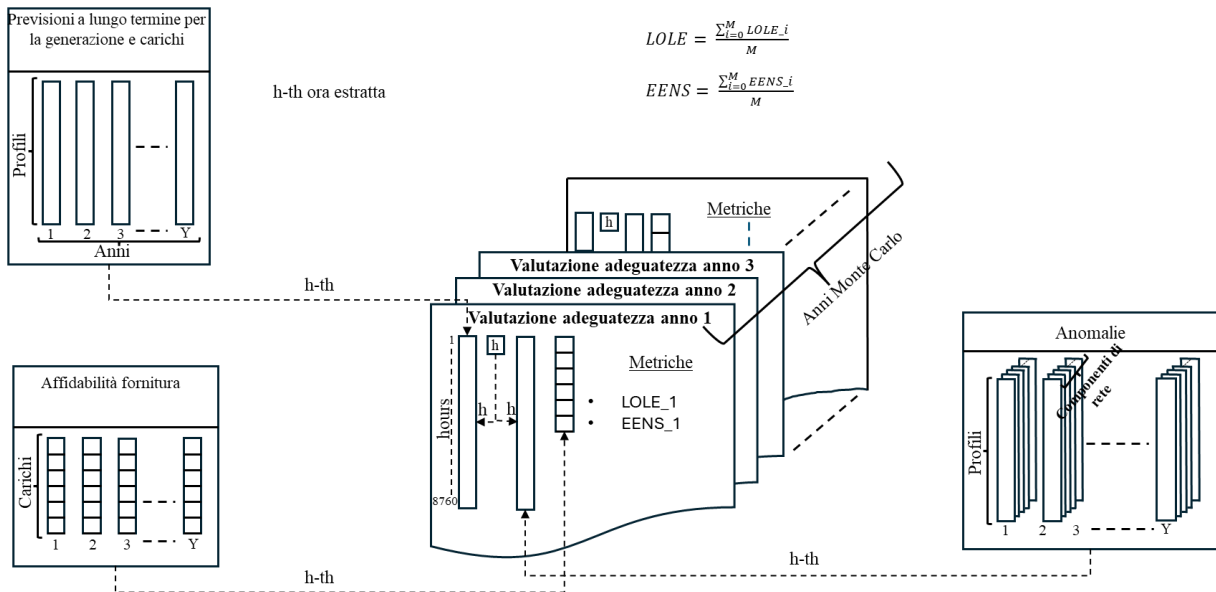


Figura 2: Rappresentazione schematica della metodologia proposta

7.2 Adequacy tool: definizione del modello e codifica in Python

La metodologia descritta al paragrafo §7.1 è stata tradotta in un codice scritto in linguaggio python. Tale codice è costituito essenzialmente da una classe principale denominata "Adeguatezza" contenente le funzioni "State_Sampling" e "generation_adequacy".

La funzione "State_Sampling" viene utilizzata per creare gli stati casuali del sistema, così come descritto al paragrafo §7.1, e, quindi, generare gli anni Monte Carlo. Per ciascun anno Monte Carlo vengono valutati i valori medi del LOLE [ore/anno MC] e dell'energia attesa non fornita EENS [kWh] e vengono prodotti i corrispondenti grafici. In particolare, la valutazione dell'adeguatezza viene eseguita considerando due casi:

- 1) valutazione dell'adeguatezza in funzione dell'affidabilità di componente e di fornitura del sistema in esame;
- 2) valutazione dell'adeguatezza considerando sia gli aspetti affidabilistici che i comportamenti anomali dei componenti di rete.

La funzione "generation_adequacy" è utilizzata per fare delle valutazioni sulla rete elettrica in esame. In particolare, vengono valutati: la potenza distribuita presente, la generazione interna totale, la generazione esterna, la domanda, il bilancio di potenza, eventuale domanda non soddisfatta (DNS).

Al paragrafo §11.1 è riportato il codice Python sviluppato.

7.3 Adequacy tool: realizzazione del modulo per l'integrazione in ARS tool

A partire dal codice sviluppato è stato implementato il software opensource per la valutazione dell'adeguatezza in funzione dell'affidabilità dei componenti di rete e dei loro possibili comportamenti anomali da integrare con il tool sviluppato nella LA1.15 di progetto. Tale implementazione ha richiesto la definizione di un dizionario e regole di interazione comuni con ARS tool. Lo schema di integrazione definito è riportato in figura 3.

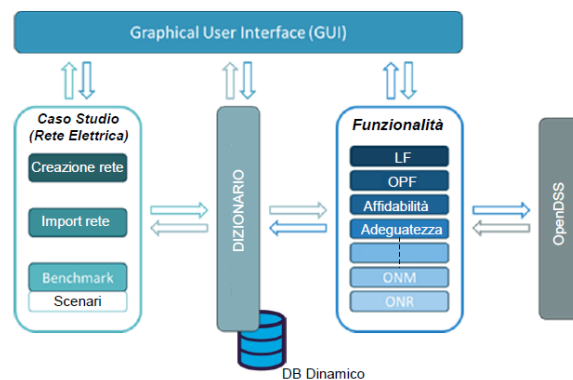


Figura 3: Schema di integrazione con tool della LA1.15

Come mostrato in Figura 3 il flusso dei dati dall'interfaccia di input (interfaccia utente) all'interfaccia di output (rappresentazione dei risultati ottenuti) viene gestito attraverso un dizionario che contiene tutte le informazioni, sia di natura elettrica (es.: potenza, tensione, corrente, ecc.) che topologica (es.: informazioni sulle connessioni tra i vari elementi di rete), della rete elettrica in esame.

Si precisa che, al fine di ottenere una rappresentazione immediata e chiara:

- i risultati sono stati raggruppati in quattro tipologie: potenza presente nella rete elettrica in [kW], possibili ore di mancata fornitura LOLE in [ore/anno], energia attesa non fornita EENA in [kWh], domanda non soddisfatta DNS, affidabilità di componente, affidabilità di fornitura distinguendo tra le ore diurne e notturne. Per ciascuna categoria di risultato è stato implementato il relativo codice python, integrato nel tool sviluppato nella LA1.15.

- Per ogni componente è stata definita una specifica interfaccia di input tramite la quale l'utente può inserire tutte le informazioni necessarie ad eseguire il calcolo dell'adeguatezza.

Esempi delle interfacce realizzate sono mostrati nelle simulazioni dei casi studio riportati nel paragrafo §11.3.

7.4 Adequacy tool: simulazioni su base caso studio

Al fine di valutare, l'impatto dell'affidabilità dei componenti e dei comportamenti anomali sull'adeguatezza, è stato condotto, utilizzando il software implementato, lo studio di una rete rappresentativa di un'area cittadina caratterizzata dalla presenza di una metropolitana, una zona residenziale, un'area per i servizi stradali e una zona con la presenza di impianti eolici e fotovoltaici. In particolare, la simulazione è stata condotta in corrispondenza di tre casi studio. Nel primo caso studio, è stata calcolata l'adeguatezza del sistema senza considerare comportamenti anomali e affidabilità; nel secondo, si è tenuto conto dell'affidabilità ma non di anomalie di comportamento dei componenti; nel terzo, sono stati considerati sia i comportamenti anomali che l'affidabilità.

Il dettaglio delle simulazioni e dei risultati è riportato nel §11.2. In questa sezione, si evidenzia che le simulazioni hanno confermato che, per lo specifico caso analizzato, pur in presenza di penetrazioni delle fonti rinnovabili che consentono di conservare un livello di adeguatezza in linea con le normative, si è osservato un peggioramento del LOLE e dell'EENS, quando lo studio dell'adeguatezza è stato condotto considerando i comportamenti anomali e/o l'affidabilità della generazione distribuita presente.

Tale aspetto, non considerato dagli attuali modelli consolidati per la valutazione dell'adeguatezza di un sistema elettrico, diverrà sempre più cruciale nei futuri scenari energetici decarbonizzati che, pertanto, dovranno avvalersi di metodologie analoghe a quella proposta nella presente LA.

7.5 Optimal Storage Tool: realizzazione del modulo per l'integrazione in ARS tool

Optimal Storage Tool è un software per la valutazione dei sistemi di accumulo e delle FER per il supporto dell'adeguatezza di sistemi elettrici decarbonizzati. Il software, fissati alcuni vincoli di sistema (es.: percentuale di FER, percentuale di elettrificazione, etc.) consente di stimare, per il sistema energetico analizzato, la quantità di energia da stoccare per rispettare i vincoli di sistema scelti, e di analizzarne l'impatto sull'adeguatezza del sistema.

La metodologia, il modello (Figura 4) e il codice Python del software sono stati sviluppati nell'ambito della LA1.4.

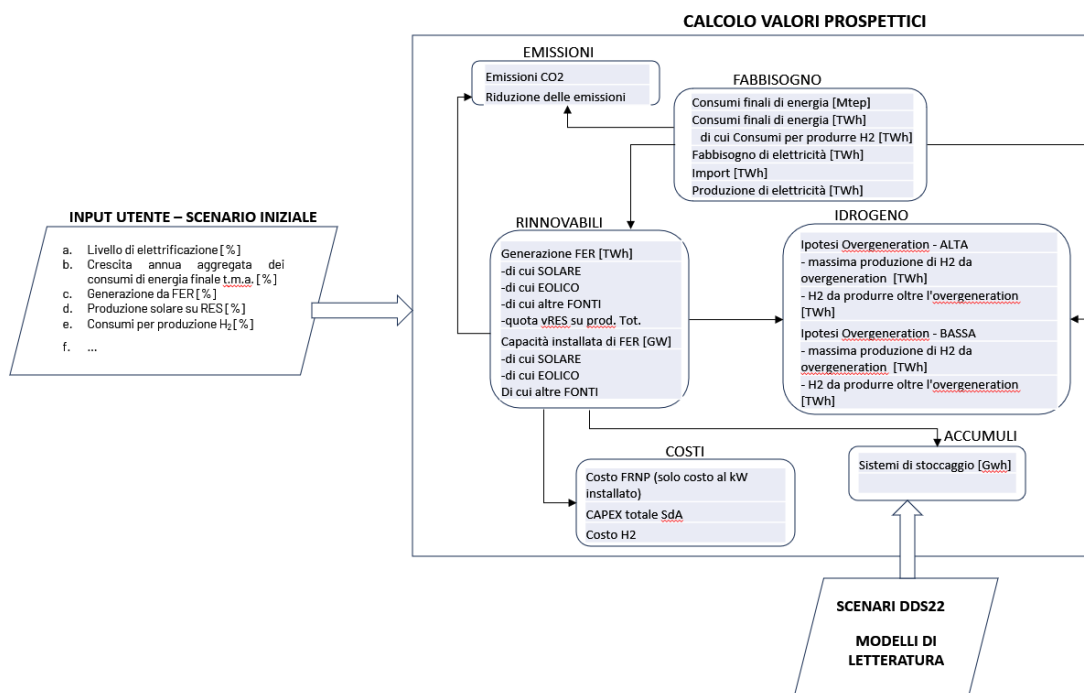


Figura 4: Modello concettuale dell'Optimal Storage Tool definito nella LA1.4

Nell'ambito della presente attività sono stati definiti il dizionario e regole di interazione comuni con ARS tool. Lo schema di integrazione definito è analogo a quanto già descritto nel §7.3 alla Figura 3.

Nello specifico, è stata creata una classe python denominata "OptStorWGT" all'interno della quale sono state definite le funzioni: "calculation", "scen_selected", "sysStor_calc", "barchart". La funzione "calculation" è utilizzata per leggere i dati di input (vincoli del sistema e scenario energetico selezionato dall'utente tramite la funzione "scen_selected") e calcolare, in base agli pseudocodici definiti nella LA1.4, i valori prospettici relativi alle categorie "fabbisogno", "Rinnovabili", "Idrogeno", "Costi", "Emissioni" mostrati in Figura 4.

Per la valutazione dei sistemi di accumulo, è stata definita la funzione "sysStor_calc". A partire dal valore della generazione da FER, calcolata nella funzione "calculation", e dalla curva di interpolazione definita nella LA1.4, viene determinato il valore prospettico, in funzione dei vincoli del sistema e dello scenario energetico scelti dall'utente, dei sistemi di accumulo espresso in [GWh].

Infine, con la funzione "barchart" sono stati generati i grafici a barre (vedi Figura 5) utili per confrontare i valori prospettici ottenuti con i valori dello scenario selezionato.

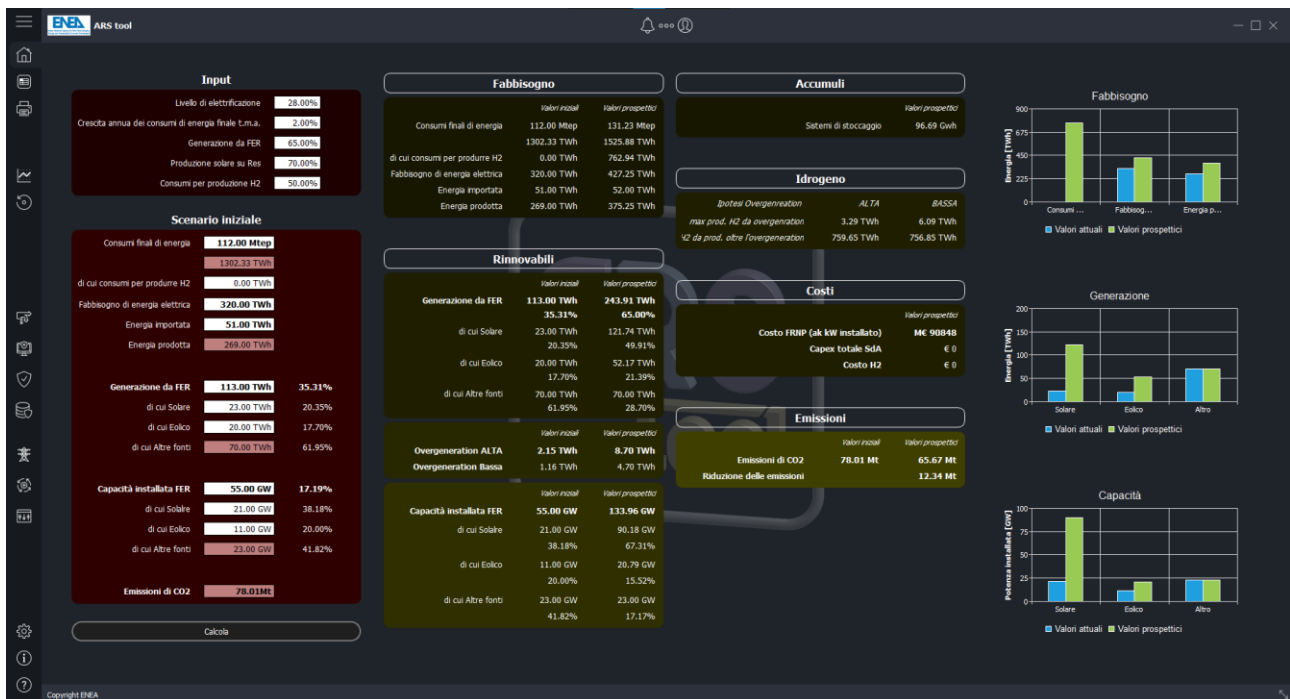


Figura 5: Schermata di output dell'Optimal Storage tool

Il codice Python di "integrazione" è riportato, insieme al codice generale del software, nel Rapporto Tecnico ENEA RdS_PTR 22-24_PR 2.3_LA1.4_156.

8 Contributo delle eventuali consulenze alle attività sopra descritte

Nella LA1.5 non sono state attivate consulenze.

9 Pubblicazioni scientifiche

- Ciavarella R., Buonanno A., Valenti M., Sforza G., Graditi G. **"An Innovative Methodology for Evaluating Power System Adequacy, Integrating Analyses of Both the Reliability and Anomalous Behaviour of Components"** (2024) Proceedings - 24th IEEE International Conference on Environment and Electrical Engineering and 8th CPS Industrial and Commercial Power Systems Europe, IEEEIC/I and CPS Europe 2024, DOI: 10.1109/IEEEIC/ICPEurope61470.2024.10751225

- Adinolfi G., Ciavarella R, Graditi G., Ricca A., Valenti M., **“Innovative Method for Reliability Assessment of Power Systems: From Components Modeling to Key Indicators Evaluation”** (2023) Electronics. <https://doi.org/10.3390/electronics13020275>

10 Eventi di disseminazione

- 1) 2024 IEEE Industrial and Commercial Power Systems Europe (IEEE ICPS Europe), Roma, Italia, 17-20 Giugno 2024
- 2) Evento Finale di disseminazione dei risultati del Progetto di Ricerca 2.3 “Evoluzione, pianificazione, gestione ed esercizio delle reti elettriche” (Piano triennale di realizzazione 2022-2024), C. R. ENEA, Portici (NA), Italia, 3 Dicembre 2024.

11 Descrizione dei risultati ottenuti

11.1 Adequacy Tool: codice Python per l’implementazione del modello di valutazione dell’adeguatezza in funzione dell’affidabilità e delle anomalie di componente e corrispondente software opensource

```
class Adeguatezza:
    def __init__(self):
        self.graphs = dict()

        self.MC_years = 10 # anni monte carlo
        self.statesampling = 8760 # stati del sistema per ogni anno monte carlo
        # estraiamo tutti i componenti di rete presenti nel dizionario
        self.buss = [componente for componente in v.keys() if v[componente]['category'] != 'ExternalGrid' if
v[componente]['category'] == 'AC-Node']
        self.trasformatori = [componente for componente in v.keys() if v[componente]['category'] == '2W-Transformer' and
v[componente]['par']['out-of-service'] == False]
        self.convertitori_dc = [componente for componente in v.keys() if v[componente]['category'] == 'DC-DC-Converter' and
v[componente]['par']['out-of-service'] == False]
        self.pwm = [componente for componente in v.keys() if v[componente]['category'] == 'Inverter' and
v[componente]['par']['out-of-service'] == False]
        self.carichi_ac = [componente for componente in v.keys() if v[componente]['category'] == 'AC-Load' and
v[componente]['par']['out-of-service'] == False]
        self.carichi_dc = [componente for componente in v.keys() if v[componente]['category'] == 'DC-Load' and
v[componente]['par']['out-of-service'] == False]
        self.link_dc = [componente for componente in v.keys() if v[componente]['category'] == 'DC-Line' and
v[componente]['par']['out-of-service'] == False]
        self.link_ac = [componente for componente in v.keys() if v[componente]['category'] == 'AC-Line' and
v[componente]['par']['out-of-service'] == False]
```

```

self.PV_AC = [componente for componente in v.keys() if v[componente]['category'] == 'AC-PV' and
v[componente]['par']['out-of-service'] == False]
self.PV_DC = [componente for componente in v.keys() if v[componente]['category'] == 'DC-PV' and
v[componente]['par']['out-of-service'] == False]
self.BESS_AC = [componente for componente in v.keys() if v[componente]['category'] == 'AC-BESS' and
v[componente]['par']['out-of-service'] == False]
self.BESS_DC = [componente for componente in v.keys() if v[componente]['category'] == 'DC-BESS' and
v[componente]['par']['out-of-service'] == False]
self.WIND_AC = [componente for componente in v.keys() if v[componente]['category'] == 'AC-Wind' and
v[componente]['par']['out-of-service'] == False]
self.WIND_DC = [componente for componente in v.keys() if v[componente]['category'] == 'DC-Wind' and
v[componente]['par']['out-of-service'] == False]
self.ext_grid = [componente for componente in v.keys() if v[componente]['category'] == 'ExternalGrid']
self.generazione_distribuita_senza_BESS = self.PV_DC + self.PV_AC + self.WIND_AC + self.WIND_DC
self.domanda = self.carichi_ac + self.carichi_dc
self.BESS = self.BESS_AC + self.BESS_DC
self.generazione_distribuita_con_BESS = self.generazione_distribuita_senza_BESS + self.BESS
self.PV = self.PV_AC + self.PV_DC
# verifico l'orizzonte temporale delle curve di previsione e carico leggendo l'output
# di un componente qualsiasi
self.horizon_time = len(v[self.generazione_distribuita_senza_BESS[0]]['f']['p'])

def somma(self,v1, v2):
    n = len(v1)
    w = []
    for i in range(n):
        w.append(v1[i] + v2[i])
    return w

def prodotto(self,v1, v2):
    n = len(v1)
    w = []
    for i in range(n):
        w.append(v1[i] * v2[i])
    return w

def differenza(self,v1, v2):
    n = len(v1)
    w = []
    for i in range(n):
        w.append(v1[i] - v2[i])
    return w

def generation_adequacy(self):
    generazione_distribuita_senza_BESS = np.zeros(self.horizon_time)
    generazione_da_external_grid = np.zeros(self.horizon_time)
    domanda_load = np.zeros(self.horizon_time)
    Pbess_generazione_totale = np.zeros(self.horizon_time)
    Pbess_domanda_totale = np.zeros(self.horizon_time)
    DNS = np.zeros(self.horizon_time)
    DNS_somma = 0
    LOLP = 0
    EDNS = 0
    LOLE = 0

```

```

for bess in self.BESS:
    Pbess = np.array(v[bess]['f']['p'])
    Pbess1 = Pbess.copy()
    Pbess2 = Pbess.copy()
    index0 = np.where(Pbess > 0)

    index1 = np.where(Pbess < 0)

    for i in range(0, len(index0[0])):
        Pbess1[index0[0][i]] = 0
    for i in range(0, len(index1[0])):
        Pbess2[index1[0][i]] = 0
    Pbess_generazione_totale = self.somma(np.abs(Pbess1), Pbess_generazione_totale)
    Pbess_domanda_totale = self.somma(np.abs(Pbess2), Pbess_domanda_totale)
    for generatore in self.generazione_distribuita_senza_BESS:
        generazione_distribuita_senza_BESS =
(self.somma(np.abs(v[generatore]['f']['p']), generazione_distribuita_senza_BESS))
    for domanda in self.domanda:
        domanda_load = self.somma(np.abs(v[domanda]['f']['p']), domanda_load)

    generazione_totale_distribuita = self.somma(Pbess_generazione_totale, generazione_distribuita_senza_BESS)
    domanda_totale = self.somma(Pbess_domanda_totale, domanda_load)

    external_grid = v['source']['f']['p'].copy()

    for i in range(len(external_grid)):
        external_grid[i] = -external_grid[i]
        if external_grid[i] > 0:
            generazione_da_external_grid[i] = external_grid[i]
    generazione_interna_totale = self.somma(generazione_da_external_grid, generazione_totale_distribuita)
    bilancio_potenza = self.somma(external_grid, generazione_totale_distribuita)
    for i in range(0, len(generazione_interna_totale)):
        if generazione_interna_totale[i] < domanda_totale[i]:
            DNS[i] = 1
            DNS_somma = domanda_totale[i] - generazione_interna_totale[i] + DNS_somma

    LOLP = sum(DNS) * 100 / self.horizon_time # Loss of Load Probability (probabilità di avere un load shedding involontario)
    EDNS = DNS_somma / self.horizon_time # [valore in potenza]Expected Demand Not Supplied
    LOLE = (LOLP / 100) * self.horizon_time # [ore/anno] Loss of load expectation

    return generazione_totale_distribuita, domanda_totale, DNS_somma, LOLP, EDNS, LOLE, DNS,
generazione_interna_totale, external_grid, bilancio_potenza

def State_Sampling(self, generazione_totale_distribuita, domanda_totale):
    durata_timestep = 1 # durata timestep in ore
    timehorizon = len(generazione_totale_distribuita) # orizzonte temporale in ore
    numerotimestep = timehorizon/durata_timestep
    SAIDI = 2.5 # ORE DI INTERRUZIONE MEDIE ANNUE PREVISTE DA ENEL NEL 2025
    LOLE = np.zeros(self.statesampling)
    LOLE_medio = np.zeros(self.statesampling)
    EENS = np.zeros(self.statesampling)
    EENS_medio = np.zeros(self.statesampling)
    LOLE_anomalies = np.zeros(self.statesampling)
    LOLE_medio_anomalies = np.zeros(self.statesampling)
    EENS_anomalies = np.zeros(self.statesampling)

```

```

EENS_medio_anomalies = np.zeros(self.statesampling)
LOLE_medio_anno_MC = np.zeros(self.MC_years)
EENS_medio_anno_MC = np.zeros(self.MC_years)
LOLE_medio_anomalies_anno_MC = np.zeros(self.MC_years)
EENS_medio_anomalies_anno_MC = np.zeros(self.MC_years)
generazione_totale_distribuita_copy = generazione_totale_distribuita.copy()
for i in range(0,self.MC_years):
    timestep_estratti = [ ]
    for j in range(0,self.statesampling):
        CV_LOLE = 0 # coefficiente di variazione
        CV_LOLE_anomalies = 0 # coefficiente di variazione
        CV_EENS = 0 # coefficiente di variazione
        LOLE_carico = 0
        EENS_carico = 0
        LOLE_carico_anomalies = 0
        EENS_carico_anomalies = 0
        LOLE_carico_degrado = 0
        EENS_carico_degrado = 0
        timestep = rand.randrange(0, numerotimestep-1) # estraggo il time step nell'orizzonte temporale che sto analizzando
        # Valuto LOLE ed ENS considerando la sola affidabilità della fornitura
        for carico in self.domanda :
            u = rand.uniform(0, 1)
            ttr = (-1/SAIDI)* math.log(u) # tempo di ripristino in ore
            fraz, int = modf(ttr / durata_timestep)
            if v[self.PV[0]][ 'f' ][ 'p' ][ timestep ] > 0 : # siamo in ore diurne
                load_rel = v[carico][ 'rel' ][ 'results' ][ 'load_rel' ]
            else:
                load_rel = v[carico][ 'rel' ][ 'results' ][ 'load_rel1' ]
            if u > load_rel: # il carico non è alimentato
                #calcolo le ore di mancata fornitura al carico
                if timestep+round(int) < numerotimestep:
                    LOLE_carico = LOLE_carico + ttr
                else:
                    LOLE_carico = LOLE_carico + numerotimestep-timestep
                # calcolo l'energia non fornita al carico
                if ttr <= durata_timestep:
                    EENS_carico = EENS_carico + v[carico][ 'f' ][ 'p' ][ timestep ]*ttr
                else:
                    if timestep + round(int)< numerotimestep:
                        for t in range(0,round(int)):
                            EENS_carico = EENS_carico + v[carico][ 'f' ][ 'p' ][ timestep+t ] * durata_timestep
                        EENS_carico = EENS_carico + v[carico][ 'f' ][ 'p' ][ timestep + round(int) ] * fraz
                    else:
                        for t in range(0, numerotimestep - 1):
                            EENS_carico = EENS_carico + v[carico][ 'f' ][ 'p' ][ timestep + t ] * durata_timestep

        ##Valuto LOLE ed ENS considerando sia l'affidabilità della fornitura che le anomalie presenti
        LOLE_carico_anomalies = LOLE_carico
        EENS_carico_anomalies = EENS_carico

    for generatore in self.generazione_distribuita_con_BEES:
        a_dict = v[generatore][ 'anom' ][ 'res' ][ 'a_dict' ]
        a_vct = v[generatore][ 'anom' ][ 'res' ][ 'a_vct' ]
        xi = v[generatore][ 'anom' ][ 'res' ][ 'xi' ]
        betai = v[generatore][ 'anom' ][ 'res' ][ 'beta' ]

```

```

        if (v[generatore]['anom']['par'].get('Hourly_Degradation') and a_vct[timestep] == 'normal' and timestep not in
timestep_estratti) or (a_vct[timestep] != 'normal' and timestep not in timestep_estratti):
            timestep_estratti.append(timestep)
            generazione_totale_distribuita_copy[timestep] = generazione_totale_distribuita[timestep] -
abs(v[generatore]['lf']['p'][timestep]) + (abs(v[generatore]['lf']['p'][timestep]) * xi[timestep] + betai[timestep])
            if generazione_totale_distribuita_copy[timestep] < domanda_totale[timestep] and a_vct[timestep] == 'normal' :
# dobbiamo considerare il naturale degrado
                EENS_carico_degrado = abs(v[generatore]['lf']['p'][timestep]) -
(abs(v[generatore]['lf']['p'][timestep]) * xi[timestep] + betai[timestep]) + EENS_carico_degrado
                LOLE_carico_degrado = LOLE_carico_degrado + 1 # il degrado dura tutto il timestep cha vale 1 ora

        if generazione_totale_distribuita_copy[timestep] < domanda_totale[timestep] and a_vct[timestep] != 'normal':
            for evento in range(0, len(a_dict.keys())):
                if timestep in a_dict[a_vct[timestep]][evento]['event' + str(evento+1)]['idx']:
                    orig_start = a_dict[a_vct[timestep]][evento]['event' + str(evento + 1)]['orig_start']
                    orig_end = a_dict[a_vct[timestep]][evento]['event' + str(evento + 1)]['orig_end']
                    if timestep == math.floor(orig_start): # il timestep coincide con il punto di partenza dell'evento
                        EENS_carico_anomalies = EENS_carico_anomalies + abs(v[generatore]['lf']['p'][timestep] -
(abs(v[generatore]['lf']['p'][timestep]) * xi[timestep] + betai[timestep])) * (orig_start - min(orig_end, math.ceil(orig_start)))) *
durata_timestep
                        LOLE_carico_anomalies = LOLE_carico_anomalies + (orig_start - min(orig_end, math.ceil(orig_start))) *
durata_timestep
                    if timestep > orig_start and timestep < math.floor(orig_end):
                        EENS_carico_anomalies = EENS_carico_anomalies + abs(v[generatore]['lf']['p'][timestep] -
abs(v[generatore]['lf']['p'][timestep]) * xi[timestep] + betai[timestep]) * durata_timestep
                        LOLE_carico_anomalies = LOLE_carico_anomalies + durata_timestep
                    if timestep == math.floor(orig_end):
                        EENS_carico_anomalies = EENS_carico_anomalies + abs(v[generatore]['lf']['p'][timestep] -
(v[generatore]['lf']['p'][timestep]) * xi[timestep] + betai[timestep]) * (orig_end - timestep) * durata_timestep
                        LOLE_carico_anomalies = LOLE_carico_anomalies + (orig_end - timestep) * durata_timestep

        LOLE_degrado_anomalie = LOLE_carico_degrado + LOLE_carico_anomalies
        EENS_degrado_anomalie = EENS_carico_degrado + EENS_carico_anomalies

        LOLE[j] = LOLE_carico
        EENS[j] = EENS_carico
        LOLE_anomalies[j] = LOLE_degrado_anomalie
        EENS_anomalies[j] = EENS_degrado_anomalie

        LOLE_medio_anno_MC[i] = sum(LOLE)/self.statesampling
        EENS_medio_anno_MC[i] = sum(EENS)/self.statesampling
        LOLE_medio_anomalies_anno_MC[i] = sum(LOLE_anomalies)/self.statesampling
        EENS_medio_anomalies_anno_MC[i] = sum(EENS_anomalies)/self.statesampling

        generazione_totale_distribuita_copy.clear()
        return LOLE_medio_anno_MC, EENS_medio_anno_MC, LOLE_medio_anomalies_anno_MC,
EENS_medio_anomalies_anno_MC

def generation_adequacy_plot(self, adequacy_result, i0=0, i1=23):
    (generazione_totale_distribuita, domanda_totale, DNS_somma, LOLP, EDNS, LOLE, DNS, generazione_interna_totale,
external_grid, bilancio_potenza) = adequacy_result

    ###PLOTTO LO STUDIO SULLA GENERATION ADEQUACY
    sample = np.zeros(len(generazione_interna_totale))
    for i in range(0, len(generazione_interna_totale[0:23])):

```

```

    sample[i] = i
figure, axis = plt.subplots(2, 1)
axis[0].plot(sample[i0:i1], generazione_totale_distribuita[i0:i1], color="yellow", label="GENERAZIONE DISTRIBUITA")
axis[0].plot(sample[i0:i1], generazione_interna_totale[i0:i1], color="red", label="GENERAZIONE INTERNA TOTALE")
axis[0].plot(sample[i0:i1], domanda_totale[i0:i1], color="blue", label="DOMANDA")
axis[0].plot(sample[i0:i1], external_grid[i0:i1], color="black", label="GENERAZIONE ESTERNA")
axis[0].plot(sample[i0:i1], bilancio_potenza[i0:i1], color="green", label="BILANCIO DI POTENZA")
axis[0].set_title("ADEQUACY")
axis[0].legend(frameon=False, loc='upper left', fontsize=5)
axis[0].set_xlabel('Tempo [h]', fontsize=5)
axis[0].set_ylabel('Potenza [kW]', fontsize=5)

axis[0].text(15, 250, "% Gen.distr./Gen. interna tot.:" +
str(round(sum(generazione_totale_distribuita)*100/sum(generazione_interna_totale),2))+"%", style='italic')
plt.xlabel('ADEQUACY')

axis[1].plot(sample[i0:i1], DNS[i0:i1], color="red", label="DNS")
axis[1].set_title("DNS")
axis[1].legend()

strFile = "generation_adequacy.png"

savepath = 'C:/Users/anton/PycharmProjects/ARStool/Functionalities/Adequacy/___images___/'

if os.path.isfile(savepath + strFile):
    os.remove(savepath + strFile) # Opt.: os.system("rm "+strFile)
filename1 = os.path.join(savepath, strFile)
plt.savefig(filename1)
plt.close()

figure1, ax1 = plt.subplots()

self.graphs[0] = {
    'title': "",
    'x-axis': 'Tempo [h]',
    'y-axis': 'Potenza [kW]',
    'ratio': 1.5,
    'x': sample[i0:i1],
    'y': {
        0: generazione_totale_distribuita[i0:i1],
        1: generazione_interna_totale[i0:i1],
        2: domanda_totale[i0:i1],
        3: external_grid[i0:i1],
        4: bilancio_potenza[i0:i1],
    },
    'colors': {
        0: 'yellow',
        1: 'red',
        2: '#00ffdd',
        3: 'white',
        4: 'green',
    },
    'labels': {
        0: 'GENERAZIONE DISTRIBUITA',
        1: 'GENERAZIONE INTERNA TOTALE',

```

```

    2: 'DOMANDA',
    3: 'GENERAZIONE ESTERNA',
    4: 'BILANCIO DI POTENZA',
  },
}

ax1.plot(sample[i0:i1], generazione_totale_distribuita[i0:i1], color="yellow",
         label="GENERAZIONE DISTRIBUITA")
ax1.plot(sample[i0:i1], generazione_interna_totale[i0:i1], color="red", label="GENERAZIONE INTERNA TOTALE")
ax1.plot(sample[i0:i1], domanda_totale[i0:i1], color="blue", label="DOMANDA")
ax1.plot(sample[i0:i1], external_grid[i0:i1], color="black", label="GENERAZIONE ESTERNA")
ax1.plot(sample[i0:i1], bilancio_potenza[i0:i1], color="green", label="BILANCIO DI POTENZA")
ax1.set_title("ADEQUACY")
ax1.legend(frameon=False, loc='upper center', bbox_to_anchor=(0.5, -0.15), fancybox=True, shadow=True, ncol=2,
          fontsize=8)
ax1.set_xlabel('Tempo [h]', fontsize=10)
ax1.set_ylabel('Potenza [kW]', fontsize=10)

box = ax1.get_position()
ax1.set_position([box.x0, box.y0 + box.height * 0.15, box.width, box.height * 0.85])
box1 = ax1.get_position()

self.x_gen_distr = sum(generazione_totale_distribuita) * 100 / sum(generazione_interna_totale)

strFile = "1.png"
if os.path.isfile(savepath + strFile):
    os.remove(savepath + strFile) # Opt.: os.system("rm "+strFile)

# ----

figure2, ax2 = plt.subplots()

self.graphs[1] = {
    'title': "",
    'x-axis': 'Tempo [h]',
    'y-axis': 'DNS [-]',
    'ratio': 1.5,
    'x': sample[i0:i1],
    'y': {
        0: DNS[i0:i1],
    },
    'colors': {
        0: 'red',
    },
    'labels': {
        0: 'DNS',
    },
}

ax2.plot(sample[i0:i1], DNS[i0:i1], color="red", label="DNS")
ax2.set_title("DNS")
ax2.legend(fontsize=8)
ax2.set_xlabel('Tempo [h]', fontsize=10)
ax2.set_ylabel('Potenza [kW]', fontsize=10)

```

```

strFile = "2.png"
if os.path.isfile(savepath + strFile):
    os.remove(savepath + strFile) # Opt.: os.system("rm "+strFile)

def state_sampling_plot(self, generazione_totale_distribuita, domanda_totale):

    (LOLE_medio_anno_MC, CV_LOLE, EENS_medio_anno_MC, LOLE_medio_anomalies_anno_MC, CV_LOLE_anomalies,
    EENS_medio_anomalies_anno_MC) = self.State_Sampling(generazione_totale_distribuita, domanda_totale)

    ##PLOTTO LO STUDIO DELL'ADEQUACY CONSIDERANDO L'AFFIDABILITÀ DELLA FORNITURA E LE ANOMALIE

    sample = np.zeros(self.MC_years)
    for i in range(0, len(sample)):
        sample[i] = i
    figure1, axis1 = plt.subplots(1, 2)
    figure1.suptitle('ADEGUATEZZA')

    self.graphs[2] = {
        'title': 'Adeguatezza',
        'x-axis': 'Anni Monte Carlo',
        'y-axis': 'LOLE [ore/anno]',
        'ratio': 0.8,
        'x': sample,
        'y': {
            0: LOLE_medio_anno_MC,
            1: LOLE_medio_anomalies_anno_MC,
        },
        'colors': {
            0: 'green',
            1: 'red',
        },
        'labels': {
            0: "con Affidabilità dei componenti",
            1: "con Affidabilità e Anomalie",
        },
    }

    axis1[0].plot(sample, LOLE_medio_anno_MC, color="blue")
    axis1[0].plot(sample, LOLE_medio_anomalies_anno_MC, color="red")

    # Shrink current axis's height by 10% on the bottom
    box = axis1[0].get_position()
    axis1[0].set_position([box.x0, box.y0 + box.height * 0.1, box.width*0.9, box.height * 0.9])

    axis1[0].text(2.5, 0.000625, "% LOLE con affidabilità fornitura:" + str(LOLE_medio_anno_MC[-1]), style='italic')
    axis1[0].text(2000, 0.4, "% LOLE con affidabilità fornitura e anomalie:" + str(LOLE_medio_anomalies_anno_MC[-1]),
    style='italic')
    axis1[0].yaxis.set_label_position("right")
    axis1.flat[0].set(xlabel='ANNI MONTE CARLO', ylabel='LOLE [ore/anno]')

    axis1[1].plot(sample, EENS_medio_anno_MC, color="blue")
    axis1[1].plot(sample, EENS_medio_anomalies_anno_MC, color="red")

```

```

axis1[1].yaxis.set_label_position("right")
axis1.flat[1].set(xlabel='ANNI MONTE CARLO', ylabel='EENS[KWh]')

# Shrink current axis's height by 10% on the bottom
box = axis1[1].get_position()
axis1[1].set_position([box.x0*1.1, box.y0 + box.height * 0.1, box.width, box.height * 0.9])

# Put a legend below current axis
figure1.legend(labels=["CONSIDERANDO L'AFFIDABILITA' DEI COMPONENTI", "CONSIDERANDO L'AFFIDABILITA' E
ANOMALIE DEI COMPONENTI"], loc="lower center", ncol=2)

strFile = "LOLE&EENS.png"

savepath = 'C:/Users/anton/PycharmProjects/ARStool/Functionalities/Adequacy/___images___/'

if os.path.isfile(savepath + strFile):
    os.remove(savepath + strFile) # Opt.: os.system("rm "+strFile)

filename1 = os.path.join(savepath, strFile)
plt1.savefig(filename1, dpi=1200)
plt1.close()

figure1 = plt.Figure(figsize=(4, 5))
ax1 = figure1.add_subplot()

ax1.plot(sample, LOLE_medio_anno_MC, color="blue", label='con Affidabilità dei componenti')
ax1.plot(sample, LOLE_medio_anomalies_anno_MC, color="red", label='con Affidabilità e Anomalie')

box = ax1.get_position()
ax1.set_position([box.x0 + box.width * 0.05, box.y0 + box.height * 0.05, box.width, box.height * 1])

ax1.set_xlabel('Anni Monte Carlo [y]', fontsize=10)
ax1.set_ylabel('Lole [h/y]', fontsize=10)
ax1.legend(frameon=False, loc='upper center', bbox_to_anchor=(0.5, -0.1), fancybox=True, shadow=True, ncol=1,
           fontsize=8)

self.avLoleFunnRel = LOLE_medio_anno_MC[-1]
self.avLoleAnom = LOLE_medio_anomalies_anno_MC[-1]
self.avEensFunnrel = EENS_medio_anno_MC[-1]
self.avEensAnom = EENS_medio_anomalies_anno_MC[-1]

strFile = "3.png"
if os.path.isfile(savepath + strFile):
    os.remove(savepath + strFile) # Opt.: os.system("rm "+strFile)
# -----

figure2 = plt.Figure(figsize=(4, 5))
ax2 = figure2.add_subplot()

self.graphs[3] = {
    'title': "",
    'x-axis': 'Anni Monte Carlo [y]',

```

```

'y-axis': 'EENS [kWh]',
'ratio': 0.8,
'x': sample,
'y': {
    0: EENS_medio_anno_MC,
    1: EENS_medio_anomalies_anno_MC,
},
'colors': {
    0: 'green',
    1: 'red',
},
'labels': {
    0: "con Affidabilità dei componenti",
    1: "con Affidabilità e Anomalie",
},
}

```

```

ax2.plot(sample, EENS_medio_anno_MC, color="blue", label='con Affidabilità dei componenti')
ax2.plot(sample, EENS_medio_anomalies_anno_MC, color="red", label='con Affidabilità e Anomalie')

```

```

box = ax1.get_position()
ax2.set_position([box.x0 + box.width * 0.05, box.y0 + box.height * 0.05, box.width, box.height * 1])

```

```

ax2.set_xlabel('Anni Monte Carlo [y]', fontsize=10)
ax2.set_ylabel('EENS [kWh]', fontsize=10)
ax2.legend(frameon=False, loc='upper center', bbox_to_anchor=(0.5, -0.1), fancybox=True, shadow=True, ncol=1,
          fontsize=8)

```

```

strFile = "4.png"
if os.path.isfile(savepath + strFile):
    os.remove(savepath + strFile)

```

11.2 Adequacy tool: risultati delle simulazioni su base caso studio

Le simulazioni sono state eseguite utilizzando la rete test mostrata in Figura 6 rappresentativa di un'area cittadina caratterizzata dalla presenza di una metropolitana, una zona residenziale, un'area per i servizi stradali e una zona con la presenza di impianti eolici e fotovoltaici.

In particolare, si è proceduto a condurre la simulazione di 3 casi studio mediante il software Adequacy tool. A parità di rete test (Figura 6), nel primo scenario, è stata calcolata l'adeguatezza del sistema senza considerare comportamenti anomali e affidabilità; nel secondo scenario, si è tenuto conto dell'affidabilità ma non di anomalie di comportamento dei componenti; nel terzo scenario, sono stati considerati sia i comportamenti anomali che l'affidabilità.

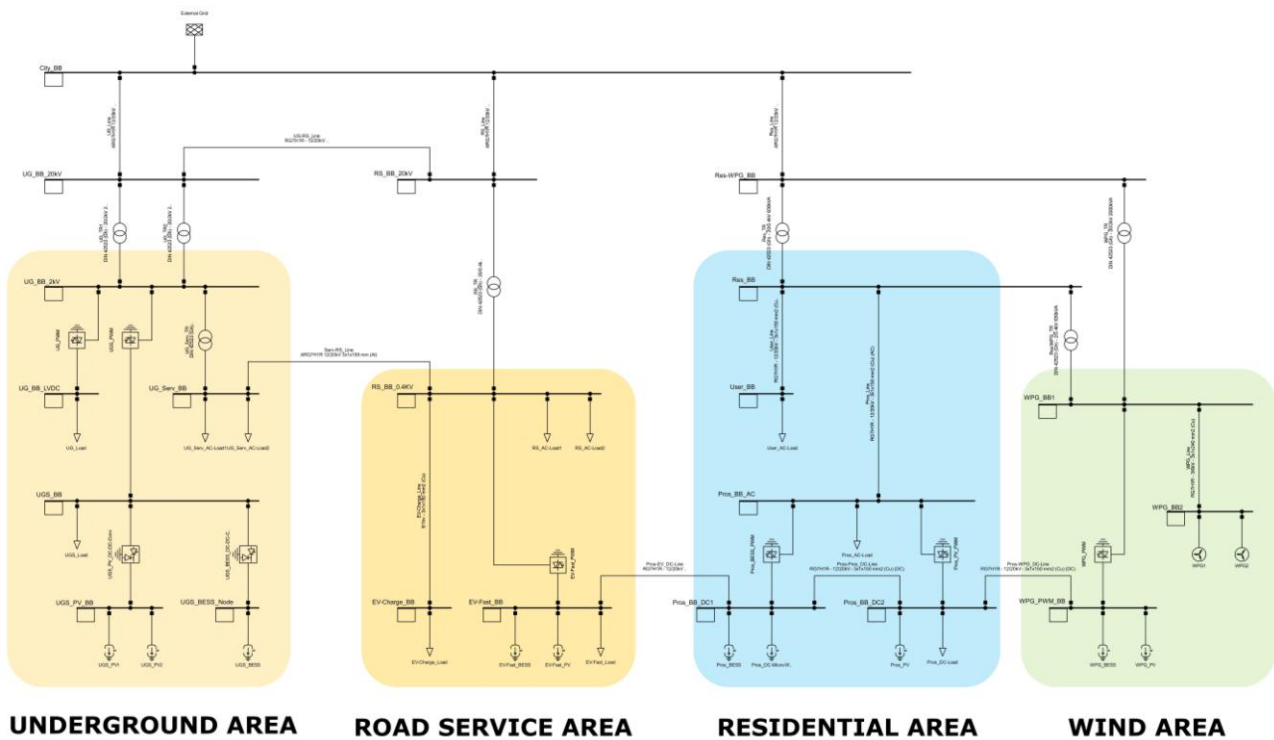


Figura 6: Rete test

Di seguito si riportano i principali parametri della rete test.

Nodi AC

Elemento	Vn [kV]
sourcebus	20.000
ug_20kv_bb	20.000
ug_2kv_bb	2.000
ug_serv_bb	0.400
rs_20kv_bb	20.000
rs_04kv_bb	0.400
ev-charge_bb	0.400
res-wpg_bb	20.000
res_bb	0.400
user_bb	0.400
pros_bb	0.400
wpg_1_bb	2.000
wpg_2_bb	2.000

Tabella 1: Nodi AC

Trasformatori

Elemento	Sr [kVA]	Vn HV [kV]	Vn LV [kV]	Rs0 [p.u.]	Rs1 [p.u.]	XHL [p.u.]	imag [p.u.]	out-of-serv.
ug_1_tr	2000.000	20.000	2.000	0.930	0.000	6.000	1.500	False
ug_2_tr	2000.000	20.000	2.000	0.930	0.000	6.000	1.500	False
ug_serv_tr	400.000	2.000	0.400	1.000	0.000	6.000	1.600	False
rs_tr	400.000	20.000	0.400	1.000	0.000	6.000	1.600	False
res_tr	630.000	20.000	0.400	1.000	0.000	6.000	1.600	False

Tabella 2: Trasformatori

Linee AC

Elemento	Lunghezza [m]	In [A]	R1 [Ohm/km]	X1 [Ohm/km]	B1 [uS/km]	R0 [Ohm/km]	X0 [Ohm/km]	B0 [uS/km]	out-of-serv.
ug_line	3.000	400.000	0.000	0.000	0.000	0.000	0.000	0.000	False
rs_line	3.000	400.000	0.000	0.000	0.000	0.000	0.000	0.000	False
ev-charge_line	0.050	400.000	0.000	0.000	0.000	0.000	0.000	0.000	False
res-wpg_line	2.000	400.000	0.000	0.000	0.000	0.000	0.000	0.000	False
user_line	0.100	400.000	0.000	0.000	0.000	0.000	0.000	0.000	False
pros_line	0.050	400.000	0.000	0.000	0.000	0.000	0.000	0.000	False
wpg_line	1.000	400.000	0.000	0.000	0.000	0.000	0.000	0.000	False

Tabella 3: Linee AC

Carichi AC

Elemento	P [kW]	cosPhi	Vn [kV]	out-of-serv.
ug_serv1_ac-load	80.000	0.900	0.400	False
ug_serv2_ac-load	120.000	0.900	0.400	False
rs_1_ac-load	40.000	0.900	0.400	False
rs_2_ac-load	60.000	0.900	0.400	False
ev-charge_ac-load	40.000	1.000	0.400	False
user_ac-load	200.000	0.900	0.400	False
pros_1_ac-load	150.000	0.900	0.400	False

Tabella 4: Carichi AC

PWM

Elemento	Sr [kVA]	Vn AC [kV]	Vn DC [kV]	out-of-serv.
ug_pwm	800.000	2.000	1.500	False
ugs_pwm	800.000	2.000	1.500	False
ev-fast_pwm	2000.000	0.400	0.600	False
pros_bess_pwm	400.000	0.400	0.600	False
pros_pv_pwm	400.000	0.400	0.600	False
wpg_pwm	2000.000	2.000	0.600	False

Tabella 5: PWM

Nodi DC

Elemento	Vn [kV]
ug_dc-bb	1.500
ugs_dc-bb	1.500
ugs_pv_dc-bb	0.750
ev-fast_dc-bb	0.600
ev-fast-load_dc-node	0.400
pros1_dc-bb	0.600
pros2_dc-bb	0.600
wpg_pv_dc-bb	0.600

Tabella 6: Nodi DC

Carichi DC

Elemento	P [kW]	Vn [kV]	out-of-serv.
ugs_dc-load	200.000	1.500	False
ugs_dc-load	200.000	1.500	False
ev-fast_dc-load	50.000	0.400	False
pros_dc-load	30.000	0.600	False

Tabella 7: Carichi DC

Batterie DC

Elemento	Capacità [kWh]	S.o.C. [p.u.]	P [kW]	eff [p.u.]	Vn [kV]	out-of-serv.
ugs_dc-bess	-20.000	50.000	-10.000	1.000	1.500	False
ev-fast_dc-bess	20.000	50.000	10.000	1.000	0.600	False
pros_dc-bess	-100.000	50.000	-50.000	1.000	0.600	False
wpg_dc-bess	-200.000	50.000	-100.000	1.000	0.600	False

Tabella 8: Batterie DC

Convertitori DC-DC

Elemento	Sr [kVA]	Vn HV [kV]	Vn LV [kV]	out-of-serv.
ugs_dc-dc-conv	50.000	1.500	0.750	False
ev-fast_dc-dc-conv	300.000	0.600	0.400	False

Tabella 9: Convertitori DC-DC

Fotovoltaico DC-DC

Elemento	Pn [kVA]	Vn [kV]	eff. [p.u.]	out-of-serv.
ugs_1_dc-pv	15.000	0.750	1.000	False
ugs_2_dc-pv	15.000	0.750	1.000	False
ev-fast_dc-pv	30.000	0.600	1.000	False
pros_dc-pv	50.000	0.600	1.000	False
wpg_dc-pv	40.000	0.600	1.000	False

Tabella 10: Fotovoltaico DC-DC

Eolici DC

Elemento	P [kW]	eff [p.u.]	Vn [kV]	out-of-serv.
pros_dc-micro-wind	10.000	1.000	0.600	False

Tabella 11: Eolici DC

Eolici AC

Elemento	P [kW]	cosPhi	eff [p.u.]	Vn [kV]	out-of-serv.
wpg_1_wind	140.000	140.000	1.000	2.000	False
wpg_2_wind	140.000	140.000	1.000	2.000	False

Tabella 12: Eolici AC

Linee DC

Elemento	Lunghezza [m]	In [A]	R [Ohm/km]	X [Ohm/km]	out-of-serv.
pros-ev_dc-line	0.020	400.000	0.123	0.100	False

Tabella 13: Linee DC

In tabella 14 sono riportati i valori di LOLE e EENS medi annui ottenuti simulando i 3 casi studio. Come è possibile osservare dal caso studio 1, senza considerare gli effetti dell'affidabilità dei componenti e le loro possibili anomalie, il sistema è perfettamente adeguato con un valore di DNS sempre nullo e la domanda sempre soddisfatta.

Dal confronto tra i valori in tabella 14 si evince che, quando si tiene conto dell'affidabilità dei componenti della rete (caso studio 2), si registra un leggero peggioramento rispetto al caso 1 sia del LOLE, che diviene pari a 0.01 ore/anno, sia dell'EENS, che diviene pari a 0.71 kWh.

Considerando anche gli effetti di possibili comportamenti anomali dei componenti di rete (caso studio 3), sia il LOLE che l'EENS peggiorano ulteriormente rispetto al caso studio 1. In particolare, il LOLE diviene pari a 0.63 ore/anno mentre l'EENS risulta di 0.99 kWh.

Tenendo conto che il valore obiettivo in termini di ore di inadeguatezza fissato dalla delibera ARERA 370/2021/R/EEL è pari a 3 ore/anno, possiamo concludere che in tutti e tre i casi studio analizzati il sistema risulta comunque adeguato.

È evidente, però, che i risultati "peggiorativi" ottenuti nei casi studio 2 e 3, mostrano che risulta cruciale nei futuri scenari energetici, caratterizzati da forte presenza di fonti rinnovabili, utilizzare strumenti che valutino l'adeguatezza considerando le anomalie di funzionamento e l'affidabilità di componente per ottenere valori maggiormente attendibili.

Valori di LOLE e EENS

Caso Studio	Descrizione	LOLE [ore/anno]	EENS [kWh]
1	Valutazione adeguatezza senza considerare comportamenti anomali e affidabilità	0	0
2	Valutazione adeguatezza considerando solo l'affidabilità dei componenti di rete	0.01	0.71
3	Valutazione adeguatezza considerando sia i comportamenti anomali che l'affidabilità dei componenti di rete	0.63	0.99

Tabella 14: Valori di LOLE e EENS nei 3 casi studio

Per completezza, si riportano in Figura 7 e Figura 8 le schermate output del tool. In particolare, la figura 7 mostra le affidabilità di componente e di fornitura ottenute nei tre casi. La figura 8, invece, mostra l'adeguatezza del sistema nei tre casi studio.

🔔 ⋮ 🔒

Affidabilità di Componente

elemento	Lambda	R	Pi_Si	MTBF (h)	MTBF (y)
ev-charge_line	3.5150E-06	0.811	0.0796	284497.62	32.4769
ev-fast_dc-bess	6.3076E-06	0.8017	0.1776	158540.1	18.09818
ev-fast_dc-dc-conv	6.3076E-06	0.8017	0.1776	158540.1	18.09818
ev-fast_dc-pv	6.9901E-06	0.75254	0.1776	143058.63	16.33089
ev-fast_pwm	6.3076E-06	0.8017	0.1776	158540.1	18.09818
pros-ev_dc-line	3.5282E-06	0.81035	0.0912	283427.03	32.35468
pros_bess_pwm	6.3076E-06	0.8017	0.1776	158540.1	18.09818
pros_dc-bess	6.3076E-06	0.8017	0.1776	158540.1	18.09818
pros_dc-micro-wind	7.5799E-06	0.70067	0.1776	131928.64	15.06035
pros_dc-pv	6.9901E-06	0.75254	0.1776	143058.63	16.33089
pros_line	3.5489E-06	0.80936	0.1093	281775.98	32.16621
pros_pv_pwm	6.3076E-06	0.8017	0.1776	158540.1	18.09818
res-wpg_line	3.5145E-06	0.81102	0.0791	284536.35	32.48132
res_tr	2.5865E-06	0.8857	0.5335	386628.98	44.13573
rs_line	3.5145E-06	0.81102	0.0791	284537.77	32.48148
rs_tr	2.5865E-06	0.8857	0.5335	386628.98	44.13573
ug_1_tr	2.5865E-06	0.8857	0.5335	386628.98	44.13573
ug_2_tr	2.5865E-06	0.8857	0.5335	386628.98	44.13573
ug_line	3.5145E-06	0.81102	0.0792	284531.55	32.48077

Affidabilità di Fornitura

elemento	R (day)	R (night)
ev-charge_ac-load	0.99971	0.99143
ev-fast_dc-load	0.99997	0.99752
pros_1_ac-load	0.99971	0.9914
pros_dc-load	0.99638	0.96235
rs_1_ac-load	0.99997	0.99752
rs_2_ac-load	0.99997	0.99752
ug_dc-load	0.97115	0.87361
ug_serv1_ac-load	0.98642	0.91856
ug_serv2_ac-load	0.98642	0.91856
ugs_dc-load	0.97115	0.87361
user_ac-load	0.99526	0.95591

Figura 7: Affidabilità dei componenti e di fornitura



Figura 8: Adeguatezza del sistema nei tre casi studio