



UNIVERSITÀ DEGLI STUDI GUGLIELMO MARCONI
FACOLTÀ DI INGEGNERIA
CORSO DI LAUREA IN INGEGNERIA INFORMATICA

Tesi di Laurea Magistrale
in Ingegneria e Qualità del Software

«Explainable Artificial Intelligence applicata a
Reti Neurali Long Short-Term Memory per la
stima della capacità di batterie agli ioni di litio»

Relatore
Prof. Erasmo Purificato

Candidato
Leonardo Dal Ronco
Matr.: 0025098

ANNO ACCADEMICO 2022/2023

Ringraziamenti

È per me un piacere e un dovere ringraziare le persone che hanno contribuito alla crescita e alla realizzazione di questa tesi.

In primis ringrazio il mio relatore, il professor Erasmo Purificato, per avermi seguito con attenzione e dedizione durante tutto il percorso di sviluppo della tesi, fornendomi preziosi consigli e utili nozioni nel momento del bisogno, e per avermi dato la possibilità di cimentarmi in un progetto stimolante e affascinante.

Un ringraziamento speciale va infine alla mia famiglia, ai miei genitori, a Paola, a Nicoletta ed Alberto, per avermi sostenuto durante questi anni di Università; ai miei amici, e a tutte le persone che mi hanno spronato e incoraggiato a dare sempre il massimo.

Indice

Abstract	5
1. Introduzione	7
2. Stato dell'arte	13
2.1 Stima della State Of Health	13
2.1.1 Experience-based models.....	13
2.1.2 Physics (model)-based models.....	13
2.1.3 Data-driven models	14
2.2 Explainable Artificial Intelligence	19
2.3 Domande di ricerca	22
3. Metodi e Tecnologie utilizzate	23
3.1 Nasa Battery Dataset	23
3.2 RUL e SOH: stima della capacità di una batteria	34
3.3 Single-Channel Input profile, Multi-Channel Input profile	35
3.4 Recurrent Neural Network (RNN)	36
3.4.1 Long Short-Term Memory (LSTM).....	39
3.5 Explainability	40
3.5.1 SHapley Additive exPlanations (SHAP).....	41
3.5.2 Deep Learning Important FeaTures (DeepLIFT).....	41
3.5.3 Saliency Map.....	41
3.5.4 Layer-wise Relevance Propagation (LRP).....	42
4. Progettazione e implementazione del sistema	43
4.1 Pulizia del Dataset	43
4.2 Normalizzazione dei dati	45
4.3 Rete Neurale LSTM	45
4.3.1 Campionamento sistematico	46
4.3.2 Architettura della rete LSTM	49

4.4 Setup sperimentale	51
4.5 Training	51
4.6 Metriche di accuratezza	51
4.6.1 B0005	52
4.6.2 B0006	53
4.6.3 B0007	54
4.6.4 B0018	55
4.6.5 Confronto tra metriche di accuratezza	56
5. Esperimenti	61
5.1 Setup sperimentale	61
5.2 Explainability	62
5.2.1 SHAP	63
5.2.1.1 B0005	64
5.2.1.2 B0006	68
5.2.1.3 B0007	72
5.2.1.4 B0018	76
5.2.2 DeepLIFT	80
5.2.3 Saliency Map	83
5.2.3.1 B0005	84
5.2.3.2 B0006	88
5.2.3.3 B0007	92
5.2.3.4 B0018	96
5.2.4 LRP	99
6. Sviluppo applicazione web	103
6.1 Streamlit	103
6.2 Applicazione web	104
6.2.1 Nasa Battery Dataset	105
6.2.2 LSTM Capacity Prediction	107
6.2.3 Explainability	109
7. Valutazione dei risultati, limitazioni e sviluppi futuri	111

7.1 Valutazione dei risultati.....	111
7.2 Limitazioni del lavoro svolto.....	113
7.3 Sviluppi futuri	114
8. Conclusioni	117
Nomenclatura	121
Bibliografia	123

Abstract

La *Prognostic and Health Management* (PHM) svolge un ruolo fondamentale all'interno di un sistema complesso, in quanto si occupa di intraprendere azioni preventive rispetto al verificarsi di un guasto, anziché limitarsi a intraprendere solo azioni di risposta dopo che il guasto si è già verificato. Per questo motivo, la PHM sta diventando sempre più rilevante dal punto di vista strategico all'interno dei sistemi di accumulo di energia. In quest'ottica, la stima accurata e affidabile della capacità di una batteria all'interno di un sistema di accumulo di energia risulta indispensabile, in modo da monitorare costantemente lo stato di salute della batteria stessa.

Ogni batteria è provvista di un *Battery Management System* (BMS), un componente elettronico che ha il compito di gestire i diversi aspetti del ciclo di vita della batteria stessa. Fra le sue funzionalità, il BMS ha il compito di utilizzare i principi della PHM per monitorare la capacità e dunque lo stato di salute della batteria, rilevando in maniera preventiva un eventuale degrado di prestazioni che potrebbe portare poi a un guasto all'interno del sistema.

Questa tesi si è quindi focalizzata sul raggiungimento di due obiettivi.

Il primo obiettivo è stato quello di sviluppare una rete neurale *Long Short-Term Memory* (LSTM) per la stima della capacità di una batteria agli ioni di litio, in grado di fornire risultati accurati come quelli trovati in letteratura (Park et al., 2020) (Choi et al., 2019), ma utilizzando un'architettura più semplice in termini di numero di parametri su cui effettuare l'addestramento (Ansari et al., 2021a), in modo che tale rete possa essere distribuita in futuro su BMS con minime risorse computazionali a disposizione. Una rete più snella non solo sarà più veloce nell'elaborazione dell'output, ma sarà più facile da spiegare tramite le tecniche di Explainability. Per la fase di training e di testing del modello, è stato utilizzato il dataset *Li-ion Battery Aging Dataset* messo a disposizione dal *NASA Prognostics Center of Excellence Data Set Repository*¹, in cui sono riportati i dati dei cicli di carica-scarica di 4 batterie agli ioni di litio.

¹ NASA Li-ion Battery Aging Dataset: <https://www.nasa.gov/content/prognostics-center-of-excellence-data-set-repository>. Visitato il 21 Luglio 2023

Le metriche di accuratezza *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE), *Mean Absolute Percentage Error* (MAPE), *Mean Absolute Error* (MAE) calcolate sul modello LSTM proposto si sono rivelate migliori rispetto alle metriche calcolate sui modelli LSTM presi come riferimento.

Il lavoro è dunque proseguito su questa strada, con l'obiettivo di verificare l'affidabilità del modello LSTM implementato per la stima della capacità di una batteria al litio, utilizzando alcune fra le più note tecniche di Explainability all'interno di un ambiente di sviluppo recente ed aggiornato, riportando per ogni tecnica non solo i risultati ottenuti, ma anche gli eventuali limiti legati a implementazioni datate e non più mantenute di tali tecniche.

Infine, è stata sviluppata e pubblicata un'applicazione web Streamlit² contenente tutto il lavoro di ricerca e sviluppo svolto in questa tesi, in modo da permettere a chiunque di esplorare il dataset *Li-ion Battery Aging Dataset* utilizzato per l'addestramento della rete neurale LSTM proposta, di interagire con il modello LSTM proposto verificandone l'accuratezza e di sperimentare le tecniche di Explainability testate sulla rete neurale proposta verificandone in autonomia l'affidabilità, in modo tale che questo possa servire come base di partenza per lavori successivi di esplorazione e approfondimento su tecniche di Explainability applicate al dominio della stima della capacità delle batterie agli ioni di litio su reti LSTM.

² Applicazione web Streamlit: <https://battery-lstm-explainability.streamlit.app/>.

CAPITOLO PRIMO

1. Introduzione

A causa della popolazione sempre più numerosa, della rapida crescita economica e dello sviluppo delle economie di mercato, la domanda globale di energia sta crescendo rapidamente. Tuttavia, l'energia necessaria per soddisfare i bisogni di industrie e famiglie al giorno d'oggi viene generata (e dunque dipende) in gran parte dai combustibili fossili. L'obiettivo dei fornitori di energia è dunque quello di spostarsi sempre di più verso fonti di energie rinnovabili (*Renewable Energy Sources*, RES). In particolare eolico e fotovoltaico sono le tecnologie su cui sono poste le maggiori aspettative di una transizione energetica verso il cosiddetto dopo-fossile.

Le fonti di energie rinnovabili presentano per loro natura due limiti principali: da un lato la variabilità e l'intermittenza della loro disponibilità nella produzione di energia, dall'altro la possibilità di immagazzinamento dell'energia da loro prodotta.

L'energia generata da RES, in particolare da impianti eolici e fotovoltaici, è per definizione condizionata dalla contestualità di una serie di condizioni climatiche favorevoli, atte ad innescare e mantenere il processo di produzione di energia elettrica. La produzione di energia rinnovabile dipende in gran parte dall'energia giornaliera o stagionale: ad esempio, l'energia generata dal vento dipende dalla velocità del vento e dalla densità dell'aria, mentre la produzione di energia da impianti fotovoltaici dipende dalla quantità di luce solare che cade sui pannelli in un dato luogo e momento. Questa intermittenza durante la produzione di energia può causare quindi una mancata corrispondenza con la domanda di energia da parte degli utenti. Per questo, le fonti di energie rinnovabili appena citate sono definite anche fonti intermittenti, in quanto non garantiscono una produzione continuativa di energia. In questo scenario, se un'abitazione privata fosse alimentata esclusivamente da pannelli solari, e per un'intera settimana ci dovesse essere maltempo, l'abitazione si ritroverebbe senza energia elettrica, in quanto i pannelli fotovoltaici a causa di condizioni meteo avverse non riuscirebbero a

soddisfare il fabbisogno energetico della famiglia residente in quell'abitazione.

In secondo luogo, l'energia elettrica prodotta da questi tipi di impianti non è un'energia che si riesce economicamente ad immagazzinare ed archiviare in grandi quantità e per lunghi periodi, come avviene invece con petrolio, carbone e gas naturale.

È stato quindi fondamentale lo sviluppo delle cosiddette *Smart Grid*, o Reti Intelligenti. Per *Smart Grid* si intende un insieme di reti di informazione e di distribuzione energetica in grado di integrare le azioni di tutti gli utenti collegati, in modo da restituire forniture elettriche sostenibili, economiche e sicure. Le *Smart Grid* nascono dunque come evoluzione del sistema elettrico tradizionale.

Negli ultimi anni, con il repentino affermarsi delle energie rinnovabili, le reti intelligenti sono diventate un'ottima soluzione su cui investire, dato che il loro scopo principale è quello di ottimizzare la distribuzione dell'energia elettrica, de-centralizzando le centrali di produzione dell'energia. Grazie alle *Smart Grid* l'utente non ha più il ruolo passivo di consumatore, ma svolge un'azione più attiva, essendo capace di produrre energia elettrica e di condividerla eventualmente con altri utenti.

All'interno delle *Smart Grid* i sistemi di accumulo di energia ricoprono un ruolo sostanziale per un funzionamento efficiente sostenibile dell'intero sistema. In generale, si parla di sistemi di stoccaggio efficienti (*Efficient Energy Storage System*, ESS), ovvero sistemi in grado di immagazzinare l'energia per un uso futuro della stessa.

Tali sistemi di accumulo riducono drasticamente la problematica dell'intermittenza della generazione di energia delle RES, riuscendo ad immagazzinare l'energia prodotta in eccesso nei periodi di punta, quando le condizioni climatiche ed ambientali sono favorevoli alla produzione di energia, e rendendola disponibile per il futuro, in momenti in cui la produzione di energia scarseggia a causa, ad esempio, di condizioni climatiche avverse o di guasti all'interno della rete di fornitura dell'energia elettrica.

Esistono diverse tipologie di ESS, all'interno delle quali troviamo sistemi di stoccaggio *Superconducting Magnetic Energy Storage* (SMES), impianti di stoccaggio dell'energia tramite aria compressa (*Compressed Air Energy Storage*, CAES), impianti *Cryogenic Energy Storage* (CES), sistemi di accumulo *Flywheel Energy Storage* (FES), e sistemi di accumulo basati su batterie (*Battery Energy Storage System*, BESS) utilizzati per immagazzinare l'energia elettrica. Questi ultimi, in particolare, sono dispositivi elettrochimici in grado di convertire l'energia elettrica in energia chimica o viceversa, a seconda della loro modalità operativa: carica o scarica. Nella pratica, i BESS sono sistemi di accumulo di energia che si basano su batterie, che possono essere caricate e scaricate più volte.

I BESS presentano numerosi vantaggi: grazie alle dimensioni di stoccaggio ridotte, alla loro semplicità di manutenzione e sostituzione, alla loro rapida risposta in caso di richiesta di energia, alla lunga durata dei cicli di carica/scarica, alla loro elevata capacità e, in generale, alla loro efficienza, si stanno rapidamente diffondendo in tutto il mondo.

A livello globale, i BESS hanno ricevuto una considerazione significativa nell'affrontare i problemi relativi alle emissioni di carbonio e ai cambiamenti climatici (Hannan et al., 2019) (Yang et al., 2021). Fra le tecnologie BESS, ricoprono un ruolo fondamentale le batterie agli ioni di litio, una tipologia particolare di batteria ricaricabile che utilizza la riduzione reversibile degli ioni di litio per immagazzinare energia. Le batterie agli ioni di litio, rispetto ad altri sistemi di accumulo di energia elettrica, presentano diversi vantaggi: offrono un'elevata densità di energia, basse cadute di tensione, elevata durata e ampie temperature di esercizio. Pertanto, hanno ottenuto un'ampia accettazione non solo all'interno di sistemi di gestione dell'energia e comunicazione (*Smart Grid*), ma anche nei veicoli elettrici, in ambito industriale e aerospaziale.

Data la centralità dei BESS all'interno di un sistema complesso, diventa essenziale prevedere lo stato di salute e la durata residua della batteria stessa, in modo da garantirne un utilizzo sicuro e affidabile.

La *Prognostic and Health Management* (PHM) è un campo ingegneristico il cui obiettivo è quello di fornire agli utenti un'analisi approfondita delle

condizioni di salute di una macchina e dei suoi componenti. Per fare questo, essa utilizza tecniche e strumenti di raccolta dati, nozioni di statistica e di fisica per rilevare un eventuale guasto (*anomaly detection*) nel sistema, classificarlo assegnandogli una tipologia specifica (*diagnostic*) e prevedere per quanto tempo la macchina sarà in grado di funzionare in presenza di questo difetto (*prognostic*).

Uno degli obiettivi della PHM è presto diventato quello di intraprendere azioni preventive rispetto al verificarsi di un guasto, anziché limitarsi a intraprendere solo azioni di risposta dopo che il guasto si è già verificato.

Più in generale, la PHM consente di segnalare preventivamente (*just in time*) quali componenti all'interno di un sistema sono afflitti da perdita di prestazioni, in modo tale che tecnici e sistemisti possano intervenire e sostituire i componenti in questione, semplificando il paradigma di manutenzione convenzionale, ed evitando dunque che tali componenti vadano a logorare le prestazioni complessive del sistema.

La PHM svolge un ruolo importante per la previsione dei guasti in diverse applicazioni del mondo reale in campo manifatturiero, industriale, aerospaziale, negli apparecchi elettronici in generale, e dunque anche nei BESS (Liu et al., 2018) (Saha et al., 2008).

Man mano che una batteria viene ripetutamente caricata e scaricata, le sostanze chimiche contenute al suo interno subiscono un fenomeno di invecchiamento, e dunque lo stato di salute della batteria (*State Of Health*, SOH) diminuisce, così come la sua capacità. Questo si traduce in un degrado generale delle prestazioni del sistema che utilizza tale batteria, aumentando la probabilità del verificarsi di un guasto all'interno del sistema stesso.

Per un corretto funzionamento di un sistema che utilizza al suo interno batterie a ioni di litio, la stima accurata del SOH delle batterie è un aspetto critico, sia per garantire un funzionamento sicuro ed efficiente dell'intero sistema, sia per ridurre al minimo i costi di manutenzione e di sostituzione delle batterie effettivamente logore.

Ogni batteria al litio è perciò provvista di un *Battery Management System* (BMS), un componente elettronico che ha il compito di gestire i diversi aspetti del ciclo di vita della batteria stessa. Fra le sue funzionalità, il BMS

ha il compito di utilizzare i principi della PHM per monitorare la SOH della batteria, rilevando in maniera preventiva un eventuale degrado di prestazioni che potrebbe portare poi a un guasto all'interno del sistema.

Questa tesi si è quindi focalizzata sul raggiungimento di due obiettivi.

Il primo obiettivo è stato quello di sviluppare una rete neurale per la stima della capacità e dunque della SOH di una batteria al litio, in grado di fornire risultati accurati come quelli trovati in letteratura (Park et al., 2020) (Choi et al., 2019), ma utilizzando un'architettura più semplice in termini di numero di parametri su cui effettuare l'addestramento (Ansari et al., 2021a), in modo che tale rete possa essere distribuita in futuro su BMS con minime risorse computazionali a disposizione. Una rete più snella non solo sarà più veloce nell'elaborazione dell'output, ma sarà più facile da spiegare tramite le tecniche di Explainability. Per la fase di training e di testing del modello, è stato utilizzato il dataset *Li-ion Battery Aging Dataset* messo a disposizione dal *NASA Prognostics Center of Excellence Data Set Repository*³, in cui sono riportati i dati dei cicli di carica-scarica di 4 batterie agli ioni di litio.

Dopo aver implementato ed addestrato la nuova rete neurale, sono state calcolate le metriche per valutare l'accuratezza della rete stessa, quali *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE), *Mean Absolute Percentage Error* (MAPE), *Mean Absolute Error* (MAE), per verificare che i risultati ottenuti fossero comparabili con quelli riportati in letteratura (Park et al., 2020) (Choi et al., 2019).

Una volta ottenuto il modello per la stima della capacità di una batteria al litio, è stato ritenuto fondamentale valutarne l'affidabilità, cercando di comprendere se questo fosse in grado di comportarsi in modo ragionevole se distribuito in un ambiente di produzione, quando avrà a che fare con dati di input inediti, in modo da garantire un funzionamento sicuro ed efficiente dell'intero sistema, e in modo da ridurre al minimo i costi di manutenzione e di sostituzione delle batterie effettivamente logore. È stata quindi utilizzata l'*Explainable Artificial Intelligence* (XAI), un insieme di metodi e tecniche che hanno lo scopo di motivare, matematicamente o graficamente, le

³ NASA Li-ion Battery Aging Dataset: <https://www.nasa.gov/content/prognostics-center-of-excellence-data-set-repository>. Visitato il 21 Luglio 2023

predizioni di un sistema di apprendimento automatico. Attraverso l'analisi delle metriche di accuratezza e dei risultati dei metodi di XAI, è possibile valutare il grado di fiducia che è possibile riporre nel sistema in questione.

Con queste premesse, il secondo obiettivo di questa tesi è stato quello di verificare l'affidabilità del modello implementato per la stima della capacità di una batteria al litio, utilizzando alcune fra le più note tecniche di Explainability all'interno di un ambiente recente ed aggiornato, riportando per ogni tecnica non solo i risultati ottenuti, ma anche gli eventuali limiti legati a implementazioni datate e non più mantenute di tali tecniche.

Infine, è stata sviluppata e pubblicata un'applicazione web Streamlit⁴ contenente tutto il lavoro di ricerca e sviluppo svolto in questa tesi, in modo da permettere a chiunque di interagire con il dataset *Li-ion Battery Aging Dataset* utilizzato per l'addestramento della rete neurale LSTM proposta, di esplorare il modello LSTM proposto verificandone l'accuratezza, e di sperimentare le tecniche di Explainability testate sulla rete neurale proposta verificandone in autonomia l'affidabilità, in modo tale che questo possa servire come base di partenza per lavori successivi di esplorazione e approfondimento su tecniche di Explainability applicate al dominio della stima della capacità delle batterie agli ioni di litio su reti LSTM.

⁴ Applicazione web Streamlit: <https://battery-lstm-explainability.streamlit.app/>.

CAPITOLO SECONDO

2. Stato dell'arte

In questo capitolo verranno descritte le tecniche utilizzate attualmente per effettuare previsioni sullo stato di salute di una batteria.

Verrà poi descritta l'Explainability, e verranno descritti gli approcci attualmente utilizzati nel dominio delle *Recurrent Neural Network*.

2.1 Stima della State Of Health

Le tecniche utilizzate per effettuare previsioni sullo stato di salute di una batteria possono essere raggruppate in 3 categorie:

- *Experience-based models*
- *Physics (model)-based models*
- *Data-driven models*

2.1.1 Experience-based models

I metodi basati sull'esperienza vengono applicati a sistemi meno complessi e stimano la SOH della batteria utilizzando la distribuzione stocastica del deterioramento. I modelli basati sull'esperienza richiedono conoscenze specialistiche e un'esperienza ingegneristica per la stima della SOH della batteria.

In questi modelli, l'esperienza ottenuta risolvendo il problema precedente viene applicata nella risoluzione di problemi simili o nuovi.

Questi modelli dipendono fortemente da un insieme di regole, che vengono definite da parte di esperti di dominio appositamente per il sistema in analisi. Infine, i modelli basati sull'esperienza hanno lo svantaggio di non riuscire ad effettuare il monitoraggio in tempo reale (Biagetti et al., 2004).

2.1.2 Physics (model)-based models

I modelli *Physics (model)-based* stimano la SOH della batteria costruendo una descrizione matematica del processo di degradazione della batteria stessa, utilizzando i dati monitorati in tempo reale sulle condizioni della batteria per aggiornare i parametri dello stesso modello (Li et al., 2014).

Il comportamento del sistema è significativamente caratterizzato dall'utilizzo di leggi matematiche e della fisica.

Sebbene sia possibile una previsione RUL accurata senza utilizzare un grande volume di dati, non esistono modelli fisici di errore ben consolidati.

Inoltre, i modelli basati sulla fisica non sono adatti a sistemi complessi a causa della mancanza di comprensione delle modalità di guasto della batteria.

Tuttavia, i modelli *Physics based* richiedono una grande quantità di dati per comprendere la curva di degrado della batteria (Liao et al., 2014).

Nel 2017, Lyu et al. hanno proposto un framework di filtraggio delle particelle (*Particle Filtering*, PF) per la previsione RUL delle batterie al piombo, incorporando un modello elettrochimico (Lyu et al., 2017).

Anche Liu et al. hanno presentato un framework di *Particle Learning* (PL) migliorato per la previsione della RUL (Liu et al., 2016).

Infine, Su et al. sono riusciti a prevedere lo stato di salute delle batterie agli ioni di litio sviluppando il meccanismo *Interacting Multiple Model Particle Filter* (IMMPF) (Su et al., 2017).

Anche se le tecniche basate su modelli per la previsione RUL hanno compiuto progressi significativi negli ultimi tempi, esistono ancora degli svantaggi; ad esempio, non esiste un modello esatto che descriva l'andamento dell'invecchiamento delle batterie, che possa dunque fungere da base per prevedere lo stato di salute delle batterie. Infatti, l'accuratezza del PF più comunemente impiegato per la previsione RUL è compromessa dal problema della degenerazione delle particelle (Liu et al., 2016).

Fra i modelli *Physics based* degni di nota, troviamo infine modelli che utilizzano un filtro di Kalman per prevedere lo stato di salute di una batteria (Plett, 2004).

2.1.3 Data-driven models

I modelli *Data-driven* stimano la SOH della batteria basandosi esclusivamente sui dati storici costituiti da precedenti misurazioni di

tensione, corrente, temperatura, capacità, ed eventualmente altri parametri della batteria in questione.

I dati storici dei vari parametri della batteria solitamente vengono registrati ed archiviati direttamente dal BMS stesso della batteria.

Come possiamo intuire, i modelli *Data-driven* non richiedono modelli matematici espliciti o una conoscenza / esperienza pregressa sull'andamento del degrado della SOH della batteria.

I modelli *Data-driven* sono dunque più accurati, più veloci e meno complessi.

C'è stata una continua evoluzione, legata a un continuo miglioramento, dei metodi utilizzati per stimare la SOH delle batterie.

Inizialmente è stato presentato un metodo basato su *Ensemble Empirical Mode Decomposition* (EEMD) e *Auto-Regressive Integrated Moving Average* (ARIMA) per la stima della SOH di una batteria (Zhou et al., 2016). In questo metodo, l'EEMD viene utilizzato inizialmente per scomporre i dati in più componenti; successivamente, ARIMA viene utilizzato per effettuare la stima. Sebbene la previsione RUL nel metodo proposto sia soddisfacente, non è possibile ottenere la funzione di densità di probabilità (*Probability Density Function*, PDF) con ARIMA.

Successivamente è stato progettato un metodo basato su *Box-Cox Transformation* (BCT) e *Monte Carlo* (MC) per la stima della SOH della batteria (Zhang et al., 2018a). Tuttavia, per un sistema non lineare, i suddetti metodi si dimostrano insufficienti nel migliorare l'accuratezza delle previsioni, a causa delle loro architetture poco profonde, dimostrando quindi capacità prognostiche inadeguate.

Inoltre, alcuni ricercatori hanno implementato anche metodi basati su kernel come *Support Vector Machine* (SVM) e *Relevance Vector Machine* (RVM). Nuhic et al. hanno proposto una *Support Vector Machine* (SVM) per prevedere lo stato di salute e la *Remaining Useful Life* (RUL) della batteria (Nuhic et al., 2013).

Liu et al. hanno proposto un algoritmo *Relevance Vector Machine* (RVM) con un metodo di addestramento online per migliorare l'accuratezza della previsione della SOH (Liu et al., 2015).

Patil et al. hanno proposto una previsione della RUL basata su un algoritmo SVM, utilizzando la tensione e la temperatura della batteria come parametri chiave (Patil et al., 2015).

L'accuratezza e la robustezza di tali modelli però sono influenzate dalla non disponibilità di una grande quantità di dati per il training, in quanto viene fatto uso dei dati storici della curva di degrado della batteria, e quindi l'accuratezza della previsione della capacità della batteria di conseguenza è compromessa.

Attualmente, modelli *Data-driven* basati su reti neurali artificiali (*Artificial Neural Network*, ANN) sono piuttosto popolari tra i ricercatori a causa della loro elevata velocità di calcolo, bassa complessità e alta efficienza.

Wu et al. hanno sviluppato una rete neurale feed-forward (*Feed-Forward Neural Network*, FFNN) per la previsione dello stato di salute e della RUL della batteria. L'autore ha utilizzato come input le tensioni della batteria durante i vari cicli di carica per prevedere la SOH e la RUL (Wu et al., 2016).

Ansari et al. hanno sviluppato una rete neurale in cascata (*Cascade Forward Neural Network*, CFNN) per la previsione RUL sotto diversi profili di input. È stato condotto uno studio comparativo per valutare le prestazioni dell'algoritmo proposto utilizzando diversi profili di ricarica (Ansari et al., 2021c).

Recentemente, è stata poi sviluppata una tecnica di intelligenza artificiale basata su una rete neurale ricorrente (*Recurrent Neural Network*, RNN) per prevedere le prestazioni dei veicoli elettrici (*Electric Vehicle*, EV) (Małek et al., 2021).

Anche Liu et al. hanno introdotto una rete neurale ricorrente per la previsione della SOH e della RUL delle batterie (Liu et al., 2010). Utilizzando i dati delle batterie della NASA sulla degradazione degli ioni di litio, gli autori hanno verificato che l'RNN è migliore sia dell'RVM che delle tecniche di *Particle Filter* (PF).

In generale, le reti neurali ricorrenti hanno ricevuto sempre più attenzione ed interesse nell'ambito della previsione dello stato di salute delle batterie rispetto ad altre architetture di reti neurali, grazie alle loro migliori prestazioni di apprendimento, alla loro elevata precisione e alla loro robustezza (Lipu et al., 2018) (Hannan et al., 2020).

Tuttavia, le RNN hanno il problema della dipendenza a lungo termine (Bengio et al., 1994) man mano che il numero di cicli di carica-scarica aumenta.

Per far fronte a questo problema, recentemente Zhang et al. hanno utilizzato reti neurali *Long Short-Term Memory* (LSTM) per memorizzare i pattern a lungo termine circa la curva di degrado della capacità di una batteria (Zhang et al., 2018b).

Sono stati poi proposti modelli “ibridi”, composti cioè dalla combinazione di LSTM e altre architetture di *machine learning* per la previsione dello stato di salute di una batteria basati solamente sullo storico dei valori di capacità della batteria (Li et al., 2019). Tuttavia, questi modelli basati su LSTM non sono stati in grado di catturare il fenomeno della rigenerazione della capacità, (aumento irregolare della capacità durante i cicli di carica-scarica), in quanto questo fenomeno non può essere previsto in maniera corretta utilizzando solo i dati sulla capacità.

In generale, le tecniche ANN presentate finora hanno riportato risultati soddisfacenti nella stima dello stato di salute delle batterie, ma mancavano di un adeguato tuning degli iperparametri, producendo quindi un errore elevato dal punto di vista delle metriche di accuratezza. Inoltre, anche se i lavori sopra citati hanno prodotto risultati soddisfacenti, è necessario un adeguato volume di dati critici per addestrare la rete in modo efficiente.

Inoltre, l'utilizzo di più parametri del profilo operativo delle batterie agli ioni di litio non è stato preso in considerazione nei lavori precedenti, a causa della mancanza di capacità di addestramento degli algoritmi.

Pertanto, è importante studiare i parametri critici dei profili operativi, in modo da addestrare una rete neurale che consenta di ottenere risultati migliori per la stima dello stato di salute di una batteria.

I lavori presentati fino a qui hanno utilizzato modelli di previsione *Single-Channel Input profile* (SCI), ovvero modelli che utilizzano il dataset di una sola batteria, ripartendolo in percentuale tra dataset di training e di testing; inoltre, tali modelli ricevono come input un solo parametro della batteria per effettuare il training.

L'utilizzo di profili SCI, come abbiamo visto, comporta valori di errore elevati per quanto riguarda le metriche di accuratezza del modello, a causa di una scarsa dimensionalità dei dati utilizzati per addestrare la rete.

Pertanto, è stato essenziale lo sviluppo di una metodologia di training tramite un profilo di ingresso multicanale (*Multi-Channel Input profile*, MCI), utilizzando dataset di diverse batterie per addestrare la rete.

Nel 2021 Ansari et al. hanno proposto un'ANN per la stima dello stato di salute di una batteria al litio, utilizzando un profilo MCI (Ansari et al., 2021b). Gli autori hanno utilizzato un dataset messo a disposizione dalla NASA per effettuare il training e la valutazione del modello tramite *Systematic Sampling*. È stato riscontrato dagli autori che la metodologia MCI ha permesso di ottenere risultati migliori dal punto di vista dell'accuratezza rispetto alle metodologie SCI.

Il lavoro sopra citato è stato poi ripreso ed ampliato dagli stessi autori, che nello stesso anno hanno proposto una RNN basata su un profilo MCI per la stima dello stato di salute di una batteria al litio, comparando questa rete neurale con reti *Feed Forward Neural Network* (FFNN), *Back Propagation Neural Network* (BPNN), *Function Fitting Neural Network*, e *Cascade Forward Neural Network* (CFNN), e dimostrando che la RNN proposta supera in termini di accuratezza tutte le altre reti testate (Ansari et al., 2021a). Nel 2020 Park et al. hanno proposto una rete neurale LSTM basata su un profilo MCI per la stima della RUL di una batteria al litio, dimostrando che l'architettura LSTM raggiunge risultati in termini di accuratezza migliori rispetto a RNN classiche, a modelli *Gated Recurrent Unit* (GRU) e a modelli

Simple Recurrent Unit (SRU) nel dominio delle batterie al litio (Park et al., 2020). Le architetture messe a confronto sono però architetture complesse, con molti parametri da apprendere durante il training.

Nel 2019 Choi et al. hanno confermato la supremazia dell'architetture LSTM su un profilo MCI per la stima dello stato di salute di una batteria al litio, confrontando tale architettura con reti *Feed-Forward Neural Network* (FFNN) e *Convolutional Neural Network* (CNN) (Choi et al., 2019).

Da quanto visto finora sulla previsione della capacità di una batteria al litio e sulla stima del suo stato di salute si può dedurre che, in questo dominio di applicazione:

- Poiché SOH e RUL di una batteria dipendono dalla stima della capacità della batteria stessa (Ansari et al., 2021a) (Park et al., 2020) (Choi et al., 2019), sarà fondamentale concentrarsi sulla stima della capacità di una batteria al litio.
- I modelli *Data-driven* sono i più accurati e i più efficienti in questo dominio di applicazione, e dunque sono da preferire rispetto a modelli *Experience-based* e *Physics (model)-based*.
- Reti neurali basate su profili MCI danno risultati migliori in termini di accuratezza rispetto a reti neurali basate su profili SCI nel dominio della stima della capacità delle batterie al litio.
- Fra le varie architetture di reti neurali, le RNN (Ansari et al., 2021a), in particolare le LSTM (Park et al., 2020) (Choi et al., 2019), danno risultati migliori in termini di accuratezza in questo dominio.
- Le reti LSTM proposte presentano architetture complesse (Park et al., 2020) (Choi et al., 2019), con molti parametri da apprendere durante il training, il che rende più lenta l'elaborazione dell'output da parte della rete stessa, in ottica poi di distribuire tale rete neurale all'interno di sistemi BMS con limitate risorse di calcolo.

2.2 Explainable Artificial Intelligence

L'apprendimento automatico è al centro di molti recenti progressi della scienza e della tecnologia. Sfortunatamente, l'importante ruolo degli esseri

umani è un aspetto spesso trascurato nel settore. Indipendentemente dal fatto che gli esseri umani utilizzino direttamente modelli per risolvere determinati problemi, rimane una preoccupazione fondamentale: se gli utenti non si fidano di un modello, o di una previsione da esso effettuata, non utilizzeranno mai quel modello in produzione.

È importante distinguere tra due diverse (ma correlate) definizioni di fiducia (Ribeiro et al., 2016):

1. *Fidarsi di una previsione*, ovvero quando un utente si fida di una singola previsione in maniera sufficiente da intraprendere un'azione basata su di essa;
2. *Fidarsi di un modello*, ovvero quando l'utente si fida del fatto che un modello riesca a comportarsi in modo ragionevole se distribuito in un ambiente di produzione, anche nel momento in cui avrà a che fare con dati di input mai visti prima.

Entrambe le definizioni di fiducia viste sopra sono direttamente influenzate da quanto l'essere umano comprende il comportamento di un modello, invece di vederlo come una scatola nera.

Valutare la fiducia delle previsioni e del modello diventa fondamentale nel momento in cui il modello viene utilizzato all'interno di un processo decisionale. Quando infatti si utilizzano tecniche di *machine learning* per task critici all'interno di un sistema, gli utenti che supervisionano tale sistema non possono e non devono fidarsi ciecamente delle stime e delle previsioni effettuate dal modello, in quanto le conseguenze potrebbero essere catastrofiche sotto diversi aspetti (economico, ambientale, morale, etc.)

In altre parole, affidarsi solamente alle metriche di accuratezza per valutare l'affidabilità di un modello non è sufficiente.

L' Explainable Artificial Intelligence (XAI) consiste dunque in un insieme di metodi, tecniche e processi che consentono agli utenti di comprendere e considerare attendibili i risultati e l'output creati dagli algoritmi di *machine learning*. La XAI è fondamentale per un'organizzazione nello sviluppo della fiducia e della sicurezza quando vengono messi in produzione modelli di intelligenza artificiale.

Per quanto riguarda il dominio delle batterie al litio, è fondamentale valutare l'affidabilità di un modello che ha come obiettivo la stima della capacità di una batteria, cercando di comprendere se questo è in grado di comportarsi in modo ragionevole se distribuito in un ambiente di produzione, quando avrà a che fare con dati di input mai visti prima, in modo da garantire un funzionamento sicuro ed efficiente dell'intero sistema, ed in modo tale da ridurre al minimo i costi di manutenzione e di sostituzione delle batterie effettivamente logore.

La letteratura, per quanto riguarda l'Explainability di reti RNN, è sfortunatamente molto scarna: pochi sono infatti i riferimenti in questo ambito, segno che l'Explainability sulle RNN è un argomento ancora da esplorare ed approfondire nel dettaglio.

Nel 2019, Schlegel et al. hanno cercato di valutare in maniera approfondita l'efficacia dei metodi XAI su modelli che lavorano su serie temporali di dati (Schlegel et al., 2019). Sono state considerate due architetture durante gli esperimenti: un'architettura basata su reti neurali convoluzionali (*Convolutional Neural Network*, CNN) ed una basata su reti neurali ricorrenti (*Recurrent Neural Network*, RNN).

È stato dimostrato che, per quanto riguarda dati di tipo *time-series* (serie temporali di dati):

- La tecnica di Explainability *LIME* (Ribeiro et al., 2016) riporta risultati poco soddisfacenti in ogni tipo di rete, molto probabilmente a causa della grande dimensionalità dei dati e del classificatore lineare impiegato.
- Le performance delle tecniche di Explainability *Saliency Map* (Simonyan et al., 2019) e *SHAP* (Lundberg et al., 2017) superano le performance delle altre tecniche per quanto riguarda la RNN, ma sono state riscontrate diminuzioni della metrica di qualità, il che è in qualche modo inaspettato; ciò implica dunque la necessità di un'ulteriore esplorazione dei metodi di XAI nelle RNN.
- *SHAP* funziona in modo affidabile per tutti i modelli, mentre altre tecniche di Explainability come *DeepLIFT* (Shrikumar et al., 2017),

LRP (Bach et al., 2015) e *Saliency Map* funzionano meglio con architetture specifiche.

- Anche *DeepLIFT* e *LRP* hanno mostrato buoni risultati.

È possibile dedurre comunque che le tecniche di Explainability più diffuse, come *SHAP*, *Saliency Map*, *DeepLIFT* e *LRP*, applicate a reti RNN riportano in generale risultati soddisfacenti, anche se nello specifico possono variare a seconda dell'architettura utilizzata.

2.3 Domande di ricerca

Le domande di ricerca che hanno motivato la scrittura di questa tesi, e a cui si vuole rispondere tramite il lavoro effettuato sono le seguenti:

- **RQ1** (*Research Question 1*): È possibile ottenere una rete neurale LSTM con la stessa struttura della rete descritta da Ansari (Ansari et al., 2021a), più semplice delle strutture descritte da Park (Park et al., 2020) e Choi (Choi et al., 2019), ma in grado di ottenere gli stessi risultati in termini di accuratezza, in modo dunque da guadagnare una rete più snella, più veloce nell'elaborazione dell'output, e dunque più adatta e versatile per essere distribuita all'interno di BMS con limitate risorse di calcolo?
- **RQ2** (*Research Question 2*): Quali tecniche di Explainability allo stato dell'arte permettono di ottenere buoni risultati in termini di spiegabilità se applicate alla rete LSTM implementata?

CAPITOLO TERZO

3. Metodi e Tecnologie utilizzate

In questo capitolo verranno descritti i metodi e le tecnologie utilizzate in questo lavoro di tesi.

Verrà descritto ed analizzato il dataset *Li-ion Battery Aging Dataset* messo a disposizione dal *NASA Prognostics Center of Excellence Data Set Repository*, utilizzato per l'addestramento del modello LSTM proposto ed implementato in questo lavoro di tesi. Verranno poi introdotti formalmente i concetti di *Remaining Useful Life* (RUL) e *State Of Health* (SOH) di una batteria, analizzandone la dipendenza con il valore di capacità della batteria stessa. Saranno poi descritte nel dettaglio le *Recurrent Neural Network* (RNN) e le reti *Long Short-Term Memory* (LSTM). Infine, verranno elencate le tecniche di Explainability applicate sul modello LSTM proposto ed implementato in questo lavoro di tesi.

3.1 Nasa Battery Dataset

Per stimare la capacità di una batteria al litio, è stato utilizzato il dataset *Li-ion Battery Aging Dataset* messo a disposizione dal *NASA Prognostics Center of Excellence Data Set Repository*⁵, in cui sono riportati i dati dei cicli di carica-scarica di quattro batterie agli ioni di litio.

In particolare, questo gruppo di batterie agli ioni di litio (B0005, B0006, B0007, B0018), ognuna con capacità nominale di 2 Ah, è stato sottoposto e monitorato sotto tre diversi profili operativi a temperatura ambiente: carica, scarica e impedenza.

La carica delle batterie è stata effettuata in modalità a corrente costante (*Constant Current*, CC) a 1,5 A fino a quando la tensione della batteria ha raggiunto 4,2 V ed è stata poi continuata in modalità a tensione costante (*Constant Voltage*, CV) fino a quando la corrente di carica è scesa a 20 mA. La scarica delle batterie è stata effettuata a un livello di corrente costante

⁵ NASA Li-ion Battery Aging Dataset: <https://www.nasa.gov/content/prognostics-center-of-excellence-data-set-repository>. Visitato il 21 Luglio 2023

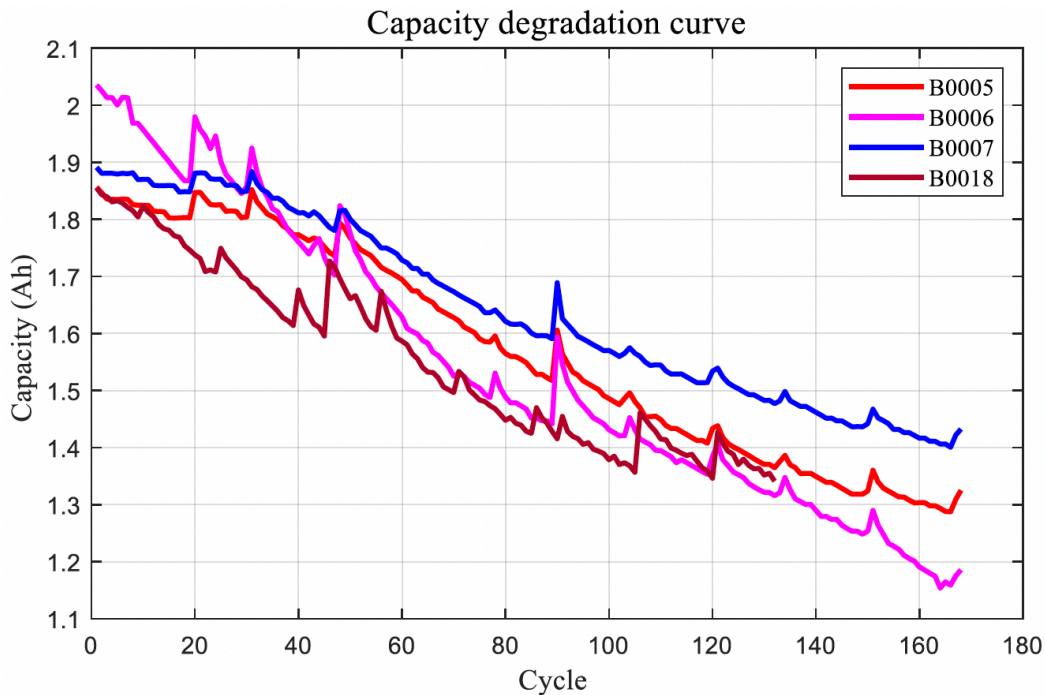
(CC) di 2 A fino a quando la tensione della batteria è scesa rispettivamente a 2,7 V, 2,5 V, 2,2 V e 2,5 V per le batterie B0005, B0006, B0007 e B0018.

La misurazione dell'impedenza, ovvero della facilità con cui la corrente alternata attraversa il circuito elettrico, è stata effettuata tramite una scansione di frequenza con la tecnica della spettroscopia di impedenza elettrochimica (*Electrochemical Impedance Spectroscopy*, EIS) da 0,1 Hz a 5 kHz.

Cicli ripetuti di carica e scarica comportano un invecchiamento accelerato delle batterie, mentre le misurazioni dell'impedenza forniscono informazioni sui parametri interni della batteria che cambiano con il progredire dell'invecchiamento.

La figura 3.1 riporta la curva di degrado della capacità delle varie batterie, dovuta ai cicli ripetuti di carica – scarica effettuati sulle stesse.

Figura 3.1: Curva del degrado della capacità delle batterie B0005, B0006, B0007, B0018 all'aumentare dei cicli di carica – scarica.



Andremo quindi ad utilizzare i dati presenti in questo dataset per addestrare un modello in grado di stimare la capacità di una batteria.

È importante considerare che durante il processo di scarica è difficile misurare o calcolare con precisione i parametri interni di una batteria perché la corrente varia rapidamente nel tempo; inoltre, i parametri di un processo di scarica in uno scenario reale potrebbero variare da contesto a contesto, in base alla modalità specifica d'uso del proprietario e utilizzatore della batteria in questione.

Al contrario, il processo di carica si basa solitamente su protocolli e parametri preimpostati dal costruttore della batteria stessa (parametri che risiedono nel dispositivo di carica della batteria), per cui le prestazioni della batteria possono essere facilmente misurate e comparate durante diversi processi di carica.

Per questi motivi, per comprendere come cambiano i parametri interni della batteria durante il suo invecchiamento, verranno sfruttati principalmente i dati relativi a tensione, intensità di corrente e temperatura della batteria misurati durante i cicli di carica.

Nei paragrafi successivi viene riportato rispettivamente l'andamento dei valori di tensione, intensità di corrente e temperatura delle batterie B0005, B0006, B0007 e B0018 registrati durante la fase di carica, all'aumentare dei cicli di carica – scarica.

Batteria B0005

Figura 3.2: andamento della tensione nel tempo, per i cicli 2, 21, 41, 61, 81, 102, 122 142 della batteria B0005.

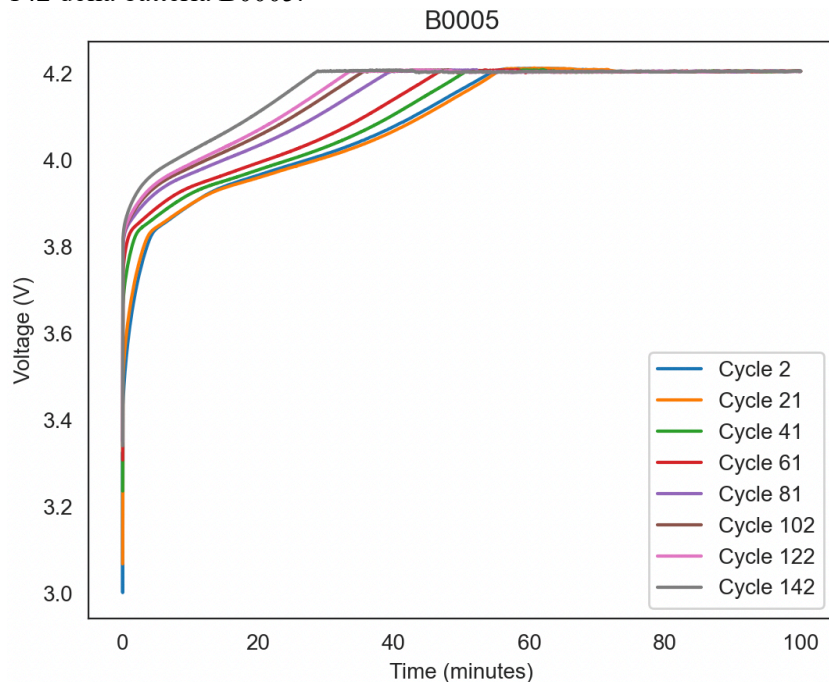


Figura 3.3: andamento dell'intensità corrente nel tempo per i cicli 2, 21, 41, 61, 81, 102, 122 e 142 della batteria B0005.

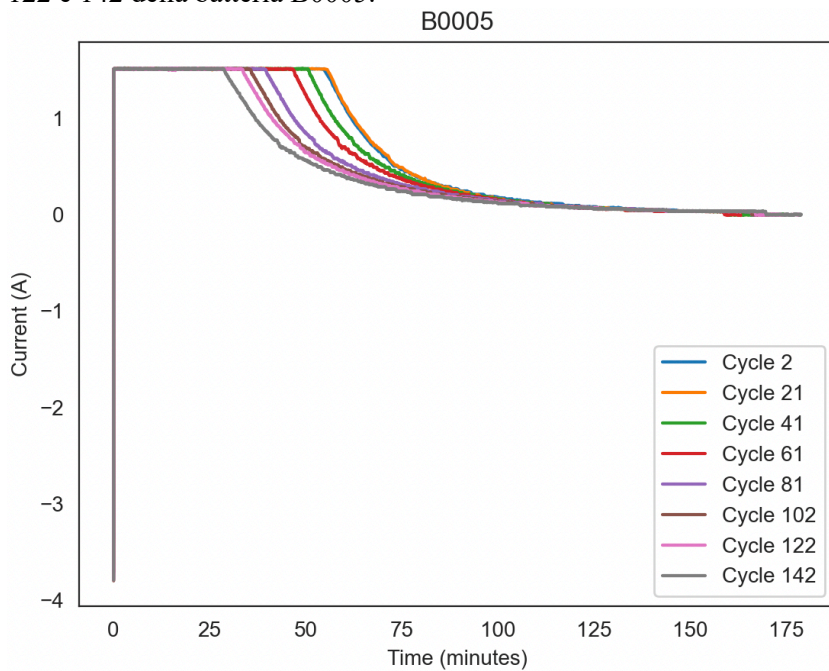
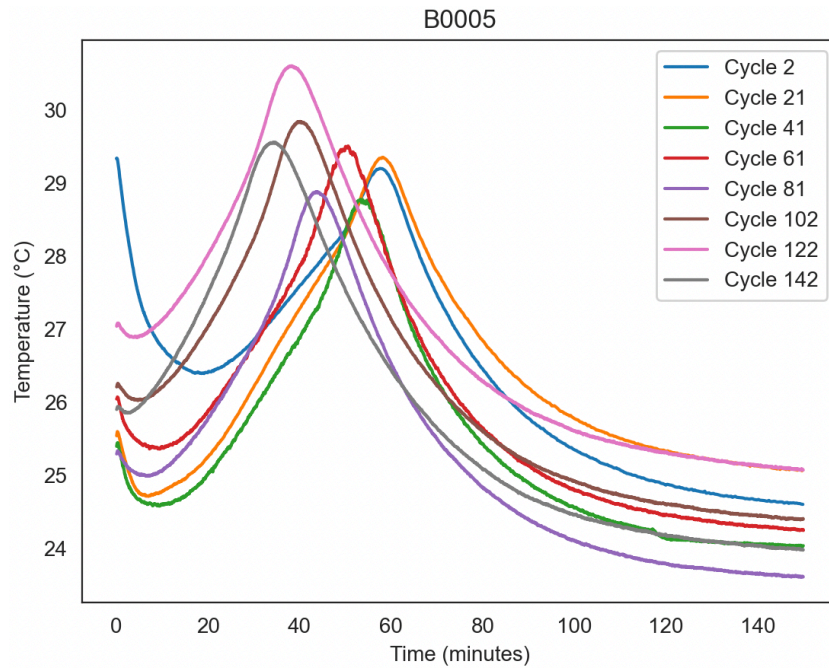


Figura 3.4: andamento della temperatura nel tempo per i cicli 2, 21, 41, 61, 81, 102, 122 e 142 della batteria B0005.



Batteria B0006

Figura 3.5: andamento della tensione nel tempo per i cicli 2, 21, 41, 61, 81, 102, 122 e 142 della batteria B0006.

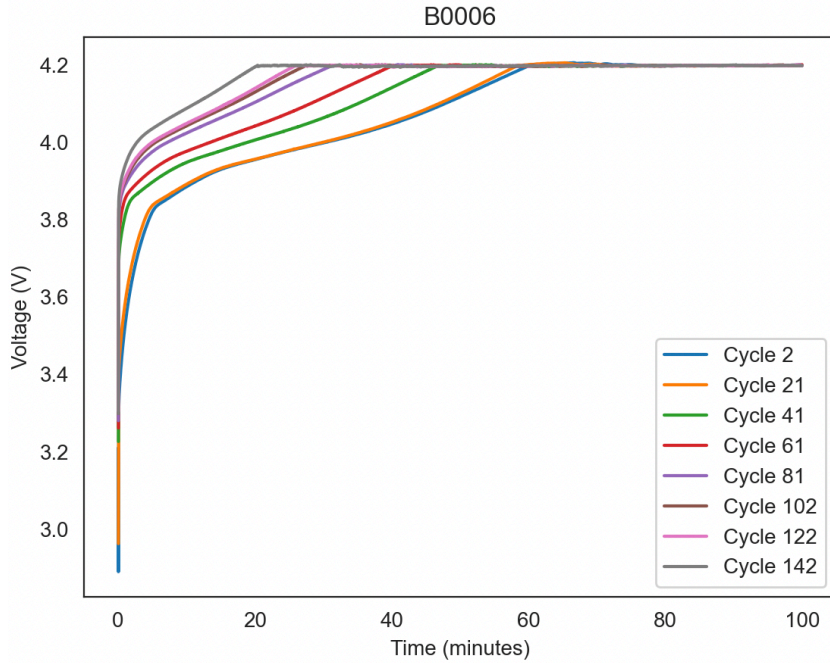


Figura 3.6: andamento dell'intensità di corrente nel tempo per i cicli 2, 21, 41, 61, 81, 102, 122 e 142 della batteria B0006.

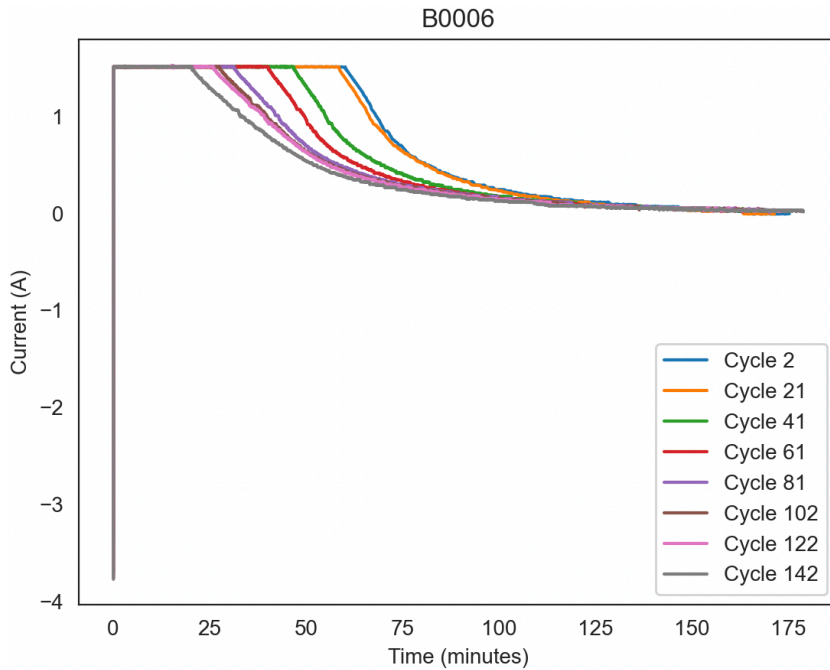
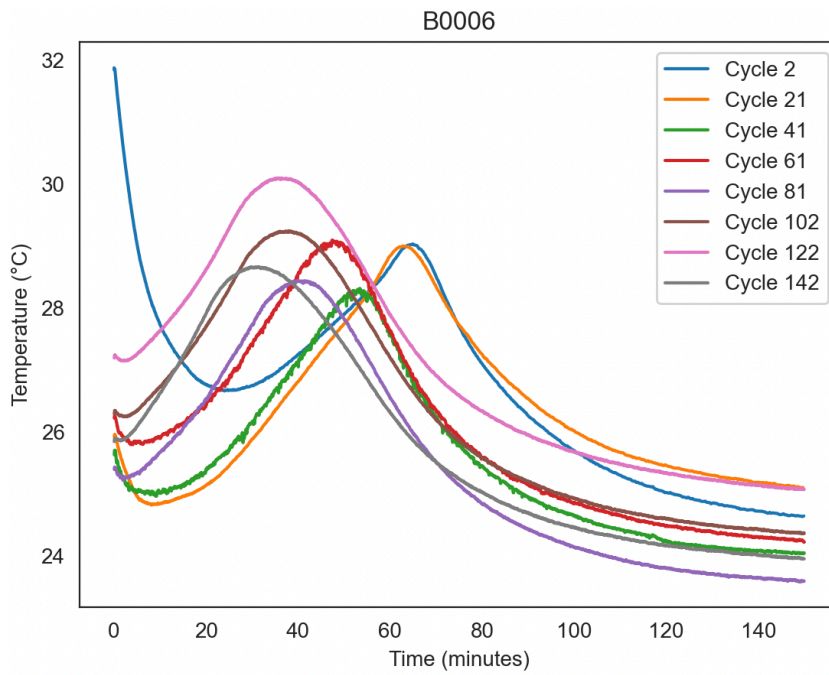


Figura 3.7: andamento della temperatura nel tempo per i cicli 2, 21, 41, 61, 81, 102, 122 e 142 della batteria B0006.



Batteria B0007

Figura 3.8: andamento della tensione nel tempo per i cicli 2, 21, 41, 61, 81, 102, 122 e 142 della batteria B0007.

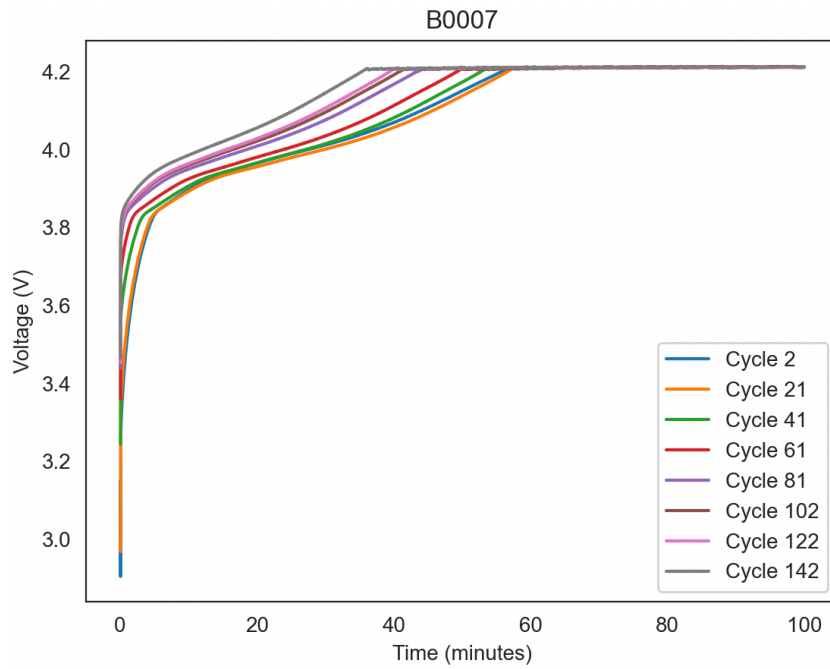


Figura 3.9: andamento dell'intensità di corrente nel tempo per i cicli 2, 21, 41, 61, 81, 102, 122 e 142 della batteria B0007.

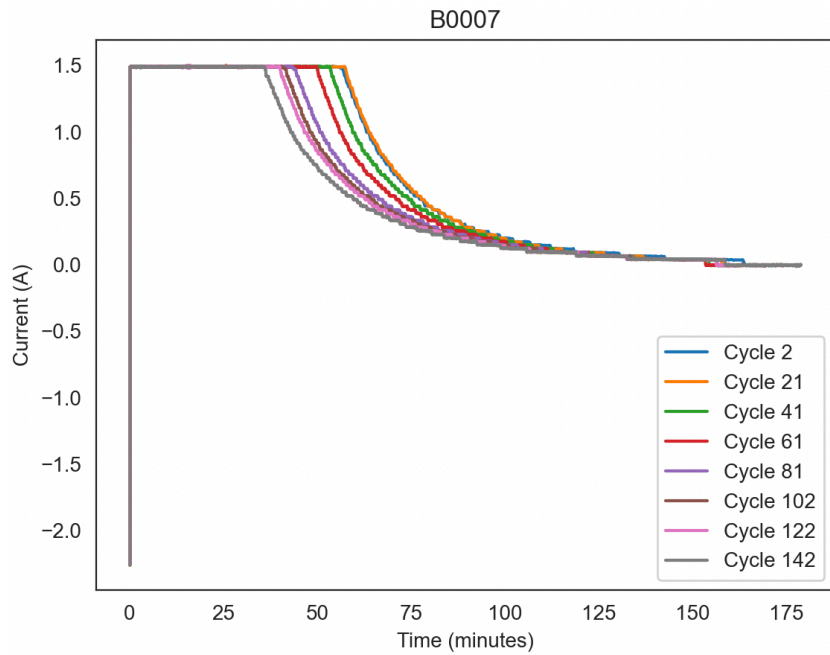
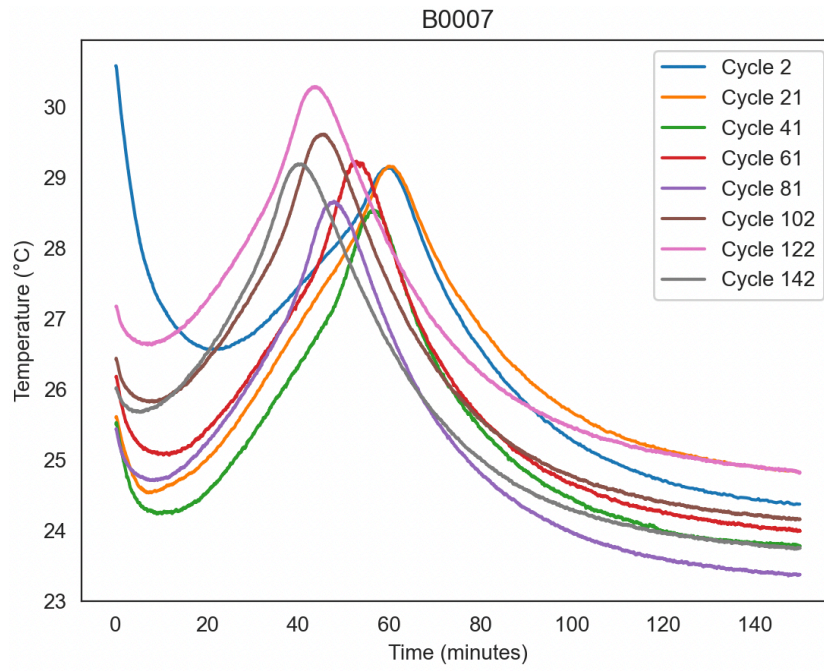


Figura 3.10: andamento della temperatura nel tempo per i cicli 2, 21, 41, 61, 81, 102, 122 e 142 della batteria B0007.



Batteria B0018

Figura 3.11: andamento della tensione nel tempo per i cicli 2, 21, 41, 61, 81, 101 della batteria B0018.

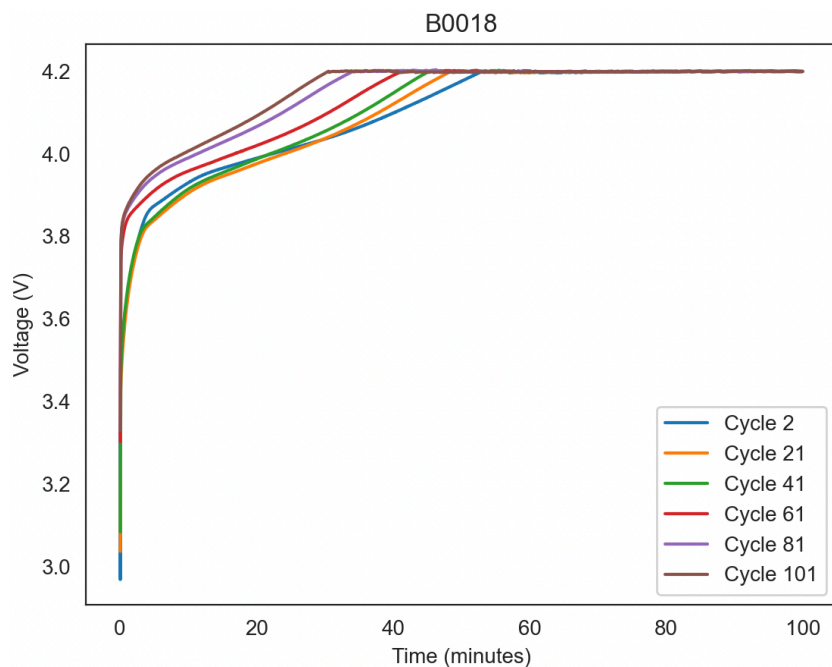


Figura 3.12: andamento dell'intensità di corrente nel tempo per i cicli 2, 21, 41, 61, 81, 101 della batteria B0018.

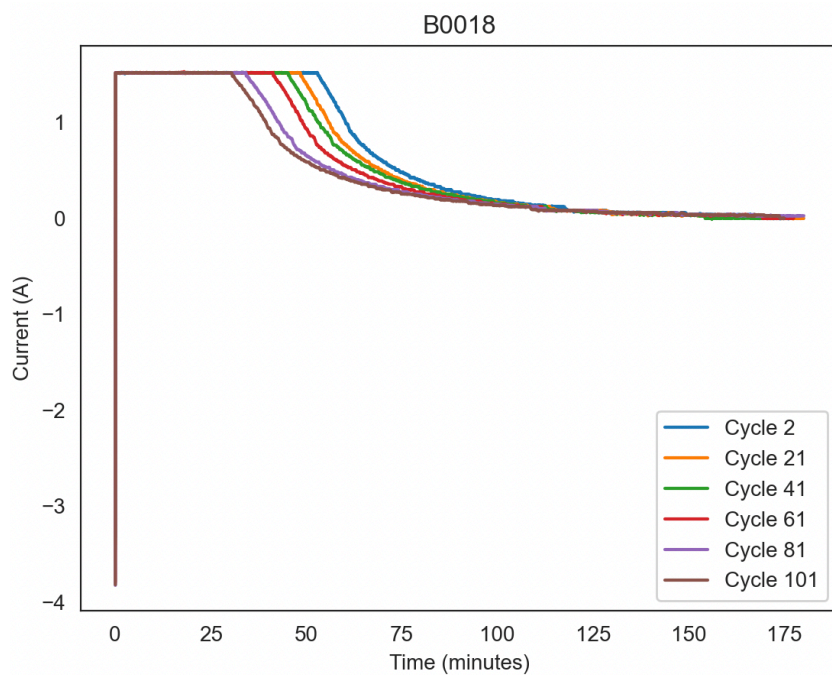
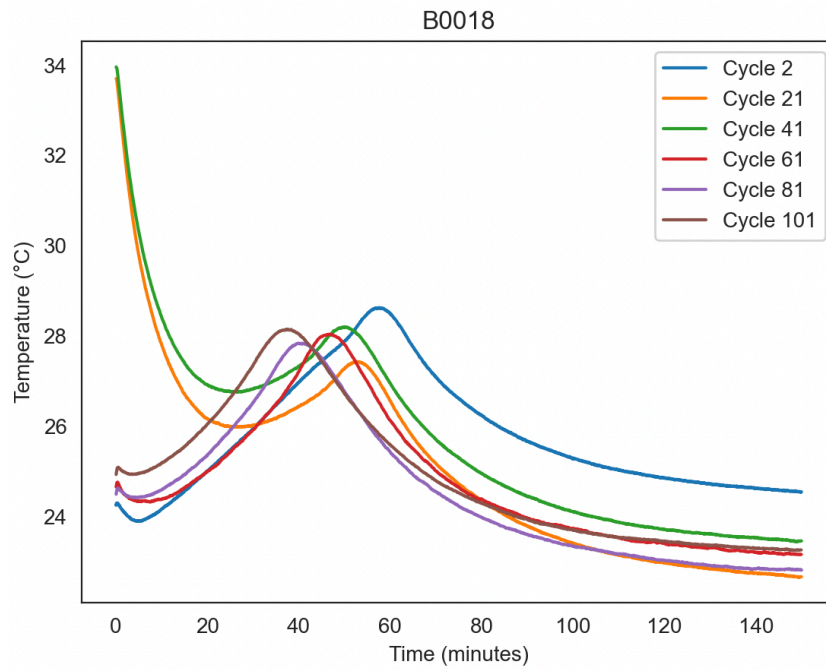


Figura 3.13: andamento della temperatura nel tempo per i cicli 2, 21, 41, 61, 81, 101 della batteria B0018.



Dall'analisi di tensione, intensità di corrente e temperatura misurate su una batteria durante la fase di carica, possiamo dedurre che, in una batteria in stato di invecchiamento avanzato, ovvero dopo numerosi cicli di carica – scarica:

1. Il valore della tensione di soglia 4.2 V viene raggiunto molto prima rispetto a una batteria più giovane, sulla quale sono stati effettuati meno cicli di carica – scarica;
2. Il valore dell'intensità di corrente diminuisce più rapidamente rispetto a una batteria più giovane, sulla quale sono stati effettuati meno cicli di carica – scarica;
3. Il valore di temperatura massima viene raggiunto più rapidamente rispetto a una batteria più giovane, sulla quale sono stati effettuati meno cicli di carica – scarica.

Da queste osservazioni, possiamo concludere che i valori di tensione, intensità di corrente e temperatura misurati durante la fase di carica di una batteria dipendono dallo stato di salute e dunque dal grado di invecchiamento della batteria stessa. Nei prossimi capitoli sfrutteremo dunque questa relazione, utilizzando questi dati per addestrare il modello per la stima della capacità di una batteria agli ioni di litio. Per comprendere come cambiano i parametri interni della batteria durante il suo invecchiamento, verrà considerata anche la capacità misurata durante il ciclo di scarica precedente.

3.2 RUL e SOH: stima della capacità di una batteria

Spesso in letteratura (Ansari et al., 2021b) (Ansari et al., 2021a) (Park et al., 2020) (Choi et al., 2019) lo stato di salute (*State Of Health*, SOH) e la vita utile residua (*Remaining Useful Life*, RUL) di una batteria vengono utilizzati come sinonimi; tuttavia, questi due termini presentano alcune differenze, sebbene abbiano anche dei punti in comune.

La stima della SOH della batteria si basa sulla stima della capacità della batteria stessa, in quanto viene calcolato a partire dalla capacità attuale della batteria, secondo la formula:

$$SOH = \frac{C_k}{C_0} \times 100$$

dove C_0 è la capacità nominale della batteria, e C_k è la capacità della batteria registrata al ciclo k -esimo.

Definito il criterio di fine vita (*End Of Life*, EOL) di una batteria come il valore di soglia di capacità superato il quale non è più garantito il corretto funzionamento della batteria stessa, la RUL di una batteria può essere definita come la differenza tra il numero totale di cicli di carica-scarica conteggiati quando la capacità effettiva della batteria scende al valore di soglia (N_{EOL}) e il numero di cicli di carica-scarica effettuati finora dalla batteria (N). La RUL esprime dunque il numero di cicli di carica-scarica rimanenti della batteria, superati i quali non sono più garantite le prestazioni e il corretto funzionamento della batteria stessa. In formule:

$$RUL = N_{EOL} - N$$

Pertanto, tutte le tecniche e i lavori presenti in letteratura (Ansari et al., 2021b) (Ansari et al., 2021a) (Park et al., 2020) (Choi et al., 2019) per la stima della SOH e della RUL di una batteria hanno come obiettivo principale quello della stima della capacità della batteria.

Per questo motivo, il modello che verrà creato avrà come obiettivo quello di stimare la capacità di una batteria al litio, in modo che tale informazione possa poi essere utilizzata per valutare lo stato di salute della batteria stessa.

3.3 Single-Channel Input profile, Multi-Channel Input profile

I modelli di *machine learning* che utilizzano un profilo a un solo canale di ingresso (*Single-Channel Input profile*, SCI), sono modelli che utilizzano il dataset di una sola batteria, ripartendolo in percentuale tra dataset di training e di testing; inoltre, tali modelli ricevono come input un solo parametro della batteria per effettuare il training.

Ad esempio, per quanto riguarda il nostro dominio di applicazione, un modello viene definito SCI se riceve come input solamente una serie storica di valori di tensione della batteria per andare a stimare lo stato di salute della batteria stessa, oppure se riceve come input solamente una serie storica di valori di capacità della batteria per andare a stimare lo stato di salute della batteria stessa.

L'utilizzo di profili SCI comporta valori di errore elevati per quanto riguarda le metriche di accuratezza del modello a causa di una scarsa dimensionalità dei dati utilizzati per addestrare la rete (Ansari et al., 2021a) (Park et al., 2020) (Choi et al., 2019).

Pertanto, è stato essenziale lo sviluppo di una metodologia di training tramite un profilo di ingresso multicanale (*Multi-Channel Input profile*, MCI), ovvero modelli che utilizzano dataset di diverse batterie per addestrare la rete; per ogni batteria, vengono utilizzati diversi parametri per il training, come la tensione, la corrente, la temperatura, e la capacità misurata durante il corrispondente ciclo di scarica della batteria stessa, in modo da riuscire a catturare il fenomeno della rigenerazione della capacità (aumento irregolare della capacità durante i cicli di carica-scarica).

Come abbiamo già visto dalla letteratura (Ansari et al., 2021a) (Park et al., 2020) (Choi et al., 2019), le metodologie MCI hanno consentito di ottenere risultati migliori dal punto di vista dell'accuratezza rispetto alle metodologie SCI. Il solo utilizzo dei dati sulla capacità di una batteria non riesce infatti a catturare in maniera corretta il fenomeno della rigenerazione della capacità, (aumento irregolare della capacità durante i cicli di carica-scarica) (Li et al., 2019).

Verrà quindi utilizzata una metodologia MCI per addestrare il modello proposto ed implementato in questo lavoro di tesi, in ottica di trovare una risposta affermativa alla RQ1.

3.4 Recurrent Neural Network (RNN)

Come abbiamo anticipato nel precedente capitolo, una rete neurale ricorrente (*Recurrent Neural Network*, RNN) è una tipologia di rete neurale in cui i neuroni appartenenti ad un determinato livello (*layer*) possono ammettere (Medsker et al., 1999):

- collegamenti all'interno dello stesso *layer*;
- collegamenti in avanti verso un *layer* successivo;
- collegamenti all'indietro verso *layer* precedenti.

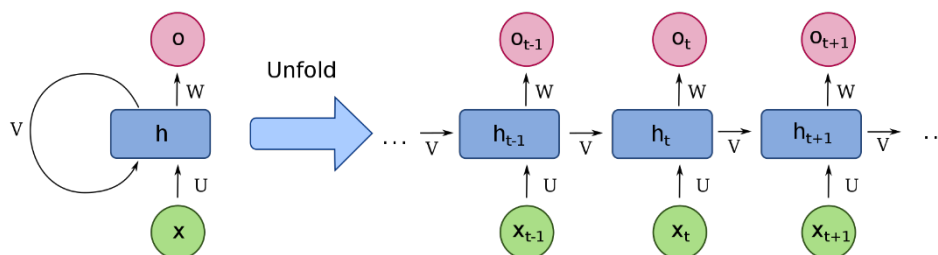
All'interno di una RNN possono quindi essere presenti dei cicli (*loop*). Il concetto di ricorrenza introdotto nelle RNN introduce intrinsecamente il concetto di memoria all'interno della rete, in quanto l'output di un nodo all'interno di una RNN dipende dagli output dei nodi precedenti all'interno della sequenza.

Per questi motivi, le RNN vengono utilizzate per attività in cui si ha a che fare con informazioni dinamiche nel tempo (sequenze o serie temporali di dati) in cui è necessario osservare e riconoscere i cambiamenti che avvengono nei dati di input al fine di generare gli output appropriati.

Una RNN può essere rappresentata da una singola cella in cui è presente un loop: si parla in questo caso di rappresentazione *folded*.

Entrando più nel dettaglio, possiamo espandere la rappresentazione *folded*, ottenendo la rappresentazione *unfolded* della rete neurale ricorrente. Tramite questa seconda rappresentazione, la RNN viene vista in maniera simile a una classica FFNN (figura 3.14).

Figura 3.14: struttura base di una RNN folded (rete di sinistra) e unfolded (rete di destra).



Fonte: https://en.wikipedia.org/wiki/Recurrent_neural_network

Nella rappresentazione *unfolded*, h (*history vector* o *state vector*) viene definito come segue:

$$h_t = f(x_t, h_{t-1})$$

dove:

- h_t = informazione temporale prodotta dall'*history vector* t .
- x_t = input all'istante t .

- h_{t-1} = informazione temporale prodotta dall'*history vector* t-1.

In altre parole, l'output della cella (*history vector*) t all'interno di un layer RNN dipende sia dall'input all'istante t, che dall'output della cella t-1.

Più formalmente, possiamo quindi esprimere h_t come:

$$h_t = \phi(Ux_t + Vh_{t-1} + b)$$

dove:

- ϕ è una funzione di attivazione.
- W è la matrice che pesa l'importanza di x_t .
- V è la matrice che pesa l'importanza di h_{t-1} .
- b è il bias, un valore che determina come traslare le due precedenti somme ponderate.

Generalmente, la funzione di attivazione ϕ utilizzata nelle RNN classiche è la tangente iperbolica, *tanh*, delimitata tra -1 e 1, estremi esclusi, definita dalla formula:

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^{2x} - 1}{e^{2x} + 1}$$

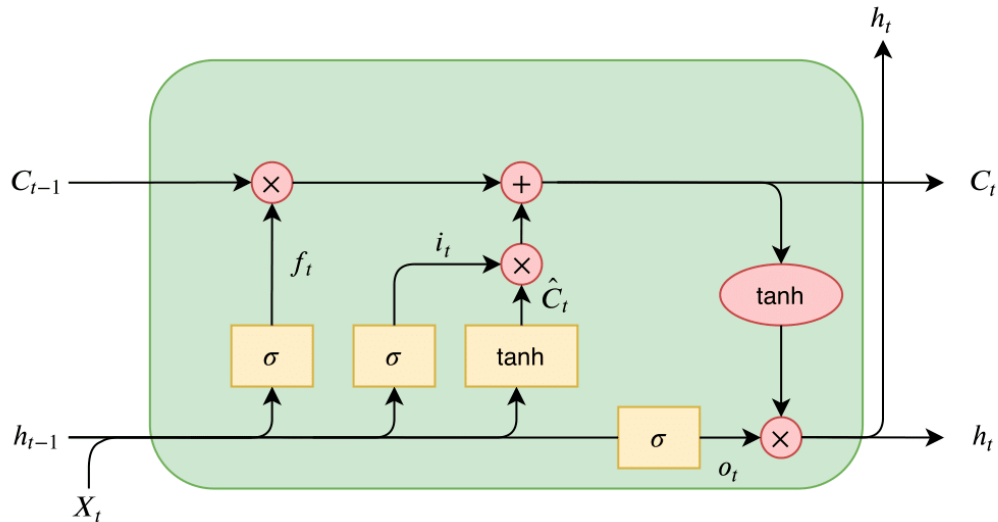
Uno dei maggiori problemi di questa rete è la scomparsa del gradiente (*vanishing gradient*), che fa sì che la struttura RNN di base diventi vulnerabile e poco efficace quando i dati delle serie temporali hanno una dipendenza a lungo termine (Hochreiter, 1998).

Per far fronte a questo ostacolo, sono stati progettati e introdotti modelli *Long Short-Term Memory* (Hochreiter et al., 1997).

3.4.1 Long Short-Term Memory (LSTM)

Le reti neurali *Long Short-Term Memory* (LSTM) sono una particolare tipologia di reti neurali ricorrenti progettate e introdotte per far fronte al problema del *vanishing gradient* tipico delle RNN.

Figura 3.15: struttura di una cella LSTM.



Fonte: https://thorirmar.com/post/insight_into_lstm/

Un'unità LSTM è composta dai seguenti elementi (figura 3.15) (Hochreiter et al., 1997):

- *Forget gate*, che ha il compito di selezionare quali informazioni possono essere rimosse dall'input x_t e dall'*history vector* $t-1$ perché non rilevanti:

$$f_t = \sigma(W_x^f x_t + W_h^f h_{t-1} + b_f)$$

- *Input gate*, che ha il compito di selezionare le informazioni rilevanti provenienti dall'input x_t e dall'*history vector* $t-1$:

$$i_t = \sigma(W_x^i x_t + W_h^i h_{t-1} + b_i)$$

- *Candidate gate*, che ha il compito di selezionare quali informazioni dovranno essere scritte nella cella:

$$\hat{C}_t = \tanh(W_x^c x_t + W_h^c h_{t-1} + b_c)$$

- *Output gate*, che ha il compito di selezionare le informazioni da dare in output:

$$o_t = \sigma(W_x^o x_t + W_h^o h_{t-1} + b_o)$$

Nelle definizioni precedenti:

- σ rappresenta la funzione sigmoide;
- l'operatore \odot rappresenta il prodotto *elementwise* (elemento per elemento).

Definiamo poi lo stato della cella all'istante t C_t come segue:

$$C_t = f_t \odot C_{t-1} + i_t \odot \hat{C}_t$$

L'*history vector* all'istante t è invece definito come:

$$h_t = o_t \odot \tanh(C_t)$$

Dalla letteratura (Park et al., 2020) (Choi et al., 2019) le LSTM hanno riportato risultati migliori nel dominio della stima della capacità di una batteria agli ioni di litio rispetto ad altre reti neurali, RNN comprese. Pertanto, il modello sviluppato in questa tesi per la stima della capacità di una batteria agli ioni di litio sarà implementato tramite un'architettura LSTM, in ottica di trovare una risposta affermativa alla RQ1.

3.5 Explainability

Possiamo raggruppare le tecniche di Explainability in due principali categorie:

- *Model-agnostic*: queste tecniche possono essere utilizzate su qualsiasi modello, e vengono applicate dopo che il modello è stato addestrato. Queste tecniche solitamente lavorano analizzando i dati di input e i risultati prodotti dal modello, considerando il modello stesso come una black-box. Questi metodi dunque non hanno accesso a informazioni interne del modello come pesi o dettagli strutturali.
- *Model-specific*: queste tecniche possono essere utilizzate solamente su determinate classi di modelli, a seconda della tecnica. Utilizzano informazioni interne del modello come pesi o dettagli strutturali per generare le loro spiegazioni.

Nei paragrafi successivi, verranno introdotte le tecniche di Explainability testate in questo lavoro di tesi.

3.5.1 SHapley Additive exPlanations (SHAP)

SHapley Additive exPlanations (SHAP) è una tecnica di Explainability che fornisce informazioni dettagliate sul contributo di ogni feature di input nell'elaborazione di una determinata stima di output da parte di un modello (Lundberg et al., 2017). Si basa sul concetto di valori Shapley, un metodo che consiste nell'assegnare credito ai singoli giocatori che partecipano ad un gioco cooperativo (Shapley, 1953). *SHAP* applica questo concetto alle feature di input di una rete neurale, calcolando il contributo medio che ogni feature dà nell'elaborazione dell'output di un modello.

In questo lavoro di tesi, verrà testata la tecnica model-agnostic *Kernel SHAP* (Lundberg et al., 2017), con l'obiettivo di rispondere alla RQ2.

3.5.2 Deep Learning Important FeaTures (DeepLIFT)

Deep Learning Important FeaTures (DeepLIFT) è una tecnica di Explainability model-agnostic, applicabile solamente a reti neurali, che consiste nell'andare a decomporre l'output generato da una rete neurale propagando all'indietro il contributo di ogni neurone della rete (Shrikumar et al., 2017). *DeepLIFT* applica questo concetto, andando a confrontare le attivazioni di ogni neurone con il loro valore di attivazione di riferimento, e assegnando un valore di contributo in base alla differenza tra i valori di attivazione confrontati. L'obiettivo è quello di ottenere il valore di importanza che ogni feature di input ha avuto nell'elaborazione di un determinato output da parte della rete.

In questo lavoro di tesi, verrà testata la tecnica di Explainability *DeepLIFT* con l'obiettivo di rispondere alla RQ2.

3.5.3 Saliency Map

Date delle feature di input e un output generato da un modello, una *Saliency Map* è una tecnica di explainability model-agnostic in grado di generare un'immagine che mette in risalto la porzione delle feature di input che ha

contribuito in maniera determinante nell'andare a generare l'output attraverso il modello preso in considerazione (Simonyan et al., 2019). Le *Saliency Map* vengono spesso adottate su reti neurali convoluzionali (*Convolutional Neural Network*, CNN), quando si ha a che fare con task di classificazione che prendono come input immagini.

In questo lavoro di tesi, verrà testata la generazione di una mappa di calore tramite la tecnica di Explainability *Saliency Map*, per andare a comprendere quali siano le feature che maggiormente hanno contribuito nell'andare a stimare la capacità della batteria, con l'obbiettivo di rispondere alla domanda RQ2.

3.5.4 Layer-wise Relevance Propagation (LRP)

Layer-wise Relevance Propagation (LRP) è una tecnica di Explainability model-agnostic, applicabile solo a reti neurali, che consiste nell'andare a decomporre l'output generato da una rete neurale propagando all'indietro il la rilevanza che ogni ogni neurone della rete ha avuto nella generazione dell'output stesso (Bach et al., 2015). *LRP* applica questo concetto con l'obbiettivo di ottenere, per ogni feature, il valore di rilevanza che questa ha avuto nell'elaborazione di un determinato output da parte della rete, esprimendo il tutto attraverso una mappa di calore.

In questo lavoro di tesi, verrà testata la tecnica di Explainability *LRP* con l'obbiettivo di rispondere alla RQ2.

CAPITOLO QUARTO

4. Progettazione e implementazione del sistema

In questo capitolo verranno descritte le operazioni di pulizia effettuate sul dataset *Li-ion Battery Aging Dataset* messo a disposizione dal *NASA Prognostics Center of Excellence Data Set Repository*. Successivamente, verrà descritto il modello proposto ed implementato per la stima della capacità di una batteria al litio.

L'obiettivo è stato quello di implementare una rete neurale LSTM con la stessa struttura della rete descritta da Ansari (Ansari et al., 2021a), più semplice delle strutture descritte da Park (Park et al., 2020) e Choi (Choi et al., 2019), ma in grado di ottenere gli stessi risultati in termini di accuratezza. Raggiungendo questo obiettivo, abbiamo guadagnato una rete più snella, più veloce nell'elaborazione dell'output, più adatta e versatile per essere distribuita all'interno di BMS con limitate risorse di calcolo e più facile da spiegare tramite le tecniche di Explainability.

Abbiamo quindi risposto in maniera affermativa alla RQ1.

4.1 Pulizia del Dataset

È stato analizzando il dataset *Li-ion Battery Aging Dataset* messo a disposizione dal *NASA Prognostics Center of Excellence Data Set Repository*⁶, utilizzato per il training della rete neurale LSTM proposta ed implementata in questo lavoro di tesi, con l'obiettivo di verificare la correttezza delle registrazioni dei cicli di carica - scarica di ogni batteria.

Una registrazione di un ciclo di carica – scarica corretta è composta da una registrazione di un ciclo di carica seguita da una registrazione di un ciclo di scarica.

Analizzando il dataset, sono emerse alcune incongruenze nei cicli di carica – scarica registrati; per questo motivo è stata effettuata un'operazione di pulizia, finalizzata a migliorare la correttezza dei dati utilizzati per il training della rete neurale LSTM proposta.

⁶ NASA Li-ion Battery Aging Dataset: <https://www.nasa.gov/content/prognostics-center-of-excellence-data-set-repository>. Visitato il 21 Luglio 2023.

In particolare:

- Per la batteria B0005:
 - l'11-esima operazione è stata eliminata, perché l'operazione successiva era anch'essa un'operazione di carica;
 - la 30-esima operazione è stata eliminata, perché l'operazione successiva era anch'essa un'operazione di carica;
 - la 90-esima operazione di scarica è stata eliminata, in quanto l'operazione precedente era anch'essa un'operazione di scarica.
- Per la batteria B0006:
 - l'11-esima operazione è stata eliminata, perché l'operazione successiva era anch'essa un'operazione di carica;
 - la 30-esima operazione è stata eliminata, perché l'operazione successiva era anch'essa un'operazione di carica;
 - la 90-esima operazione di scarica è stata eliminata, in quanto l'operazione precedente era anch'essa un'operazione di scarica.
- Per la batteria B0007:
 - l'11-esima operazione è stata eliminata, perché l'operazione successiva era anch'essa un'operazione di carica;
 - la 30-esima operazione è stata eliminata, perché l'operazione successiva era anch'essa un'operazione di carica;
 - la 90-esima operazione di scarica è stata eliminata, in quanto l'operazione precedente era anch'essa un'operazione di scarica.
- Per la batteria B0018:
 - la 45-esima operazione è stata eliminata, perché l'operazione successiva era anch'essa un'operazione di carica;
 - la 55-esima operazione è stata eliminata, perché l'operazione successiva era anch'essa un'operazione di carica.

Riassumendo:

- per la batteria B0005, sono stati considerati validi 167 cicli di carica – scarica;
- per la batteria B0006, sono stati considerati validi 167 cicli di carica – scarica;
- per la batteria B0007, sono stati considerati validi 167 cicli di carica – scarica;
- per la batteria B0018, sono stati considerati validi 132 cicli di carica – scarica.

4.2 Normalizzazione dei dati

Una volta estratte le feature tramite il campionamento sistematico, è stata effettuata un'operazione di normalizzazione con l'obiettivo di raggiungere migliori risultati in termini di accuratezza.

Tramite la normalizzazione, tutte le colonne dei dataset utilizzati sono state trasformate in modo che assumessero valori compresi nell'intervallo $[0, 1]$. Data la colonna $j = 1, \dots, n$ del nostro dataset, e supponendo che il nostro dataset abbia m righe, siano:

$$m_j = \min_{i=1, \dots, m} x_{ij}; \quad M_j = \max_{i=1, \dots, m} x_{ij}$$

rispettivamente il minimo e il massimo valore assunto dalla colonna j . La normalizzazione è stata effettuata attraverso la seguente mappa affine:

$$x'_{ij} = \frac{x_{ij} - m_j}{M_j - m_j}$$

4.3 Rete Neurale LSTM

In questa sezione verrà descritta la rete neurale LSTM implementata, descrivendo come sono state ottenute le feature di input tramite il metodo del campionamento sistematico, per poi arrivare all'architettura della rete neurale LSTM vera e propria.

Verranno poi estrapolate le metriche di accuratezza della rete LSTM implementata, e verranno confrontate con le metriche di accuratezza elaborate da Park (Park et al., 2020) e Choi (Choi et al., 2019).

4.3.1 Campionamento sistematico

Il campionamento sistematico è una tecnica di campionamento utilizzata in statistica e probabilità, che consiste nell'estrarre campioni a intervalli regolari a partire da una popolazione di riferimento ordinata.

Definiamo N come l'ampiezza della popolazione di riferimento, e stabiliamo n come ampiezza del campione (numero di campioni che verranno estratti), è possibile definire l'intervallo di campionamento k come:

$$k = \frac{N}{n}$$

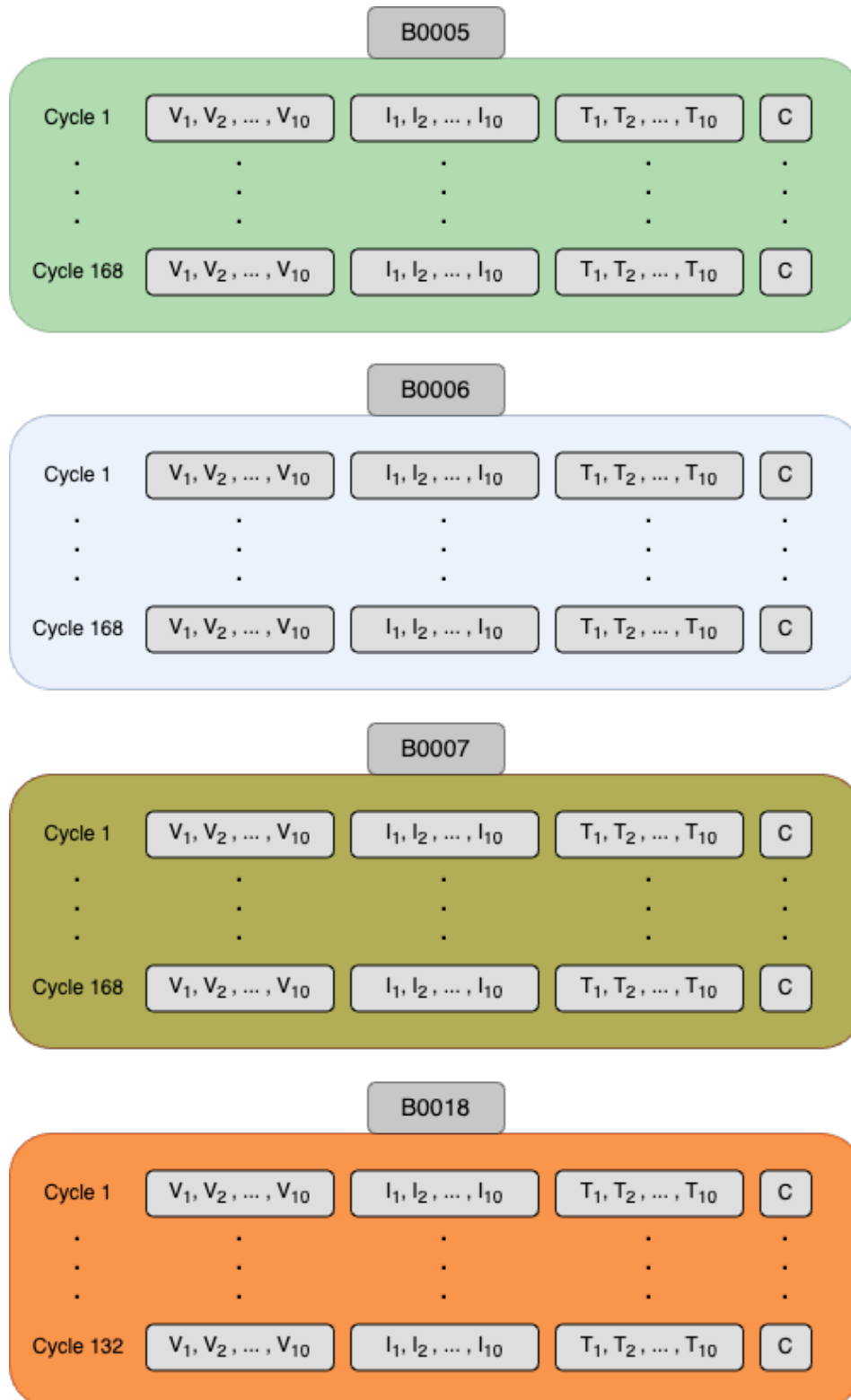
Dunque, assumendo che k sia un numero intero, verranno estratti n campioni dalla popolazione di riferimento ordinata, partendo dal primo elemento, e procedendo a intervalli di k elementi.

Per il training del modello LSTM proposto verranno utilizzati i dati relativi a tensione, intensità di corrente e temperatura relativi a un ciclo di carica, nonché la capacità registrata durante il ciclo di scarica precedente, in modo da stimare la capacità attuale della batteria.

In particolare, per ogni ciclo di carica, verranno estratti 10 valori per la tensione, 10 valori per l'intensità di corrente e 10 valori per la temperatura, estratti tramite la tecnica del campionamento sistematico in modo da formare insieme alla capacità registrata durante il ciclo di scarica precedente un vettore di input formato da 31 feature.

È dunque stato utilizzato lo stesso profilo MCI adottato da Ansari (Ansari et al., 2021a), in quanto sono stati utilizzati diversi parametri per il training, come la tensione, la corrente, la temperatura e la capacità misurata durante il corrispondente ciclo di scarica della batteria stessa.

Figura 4.1: estrazione dei dati di training tramite campionamento sistematico.

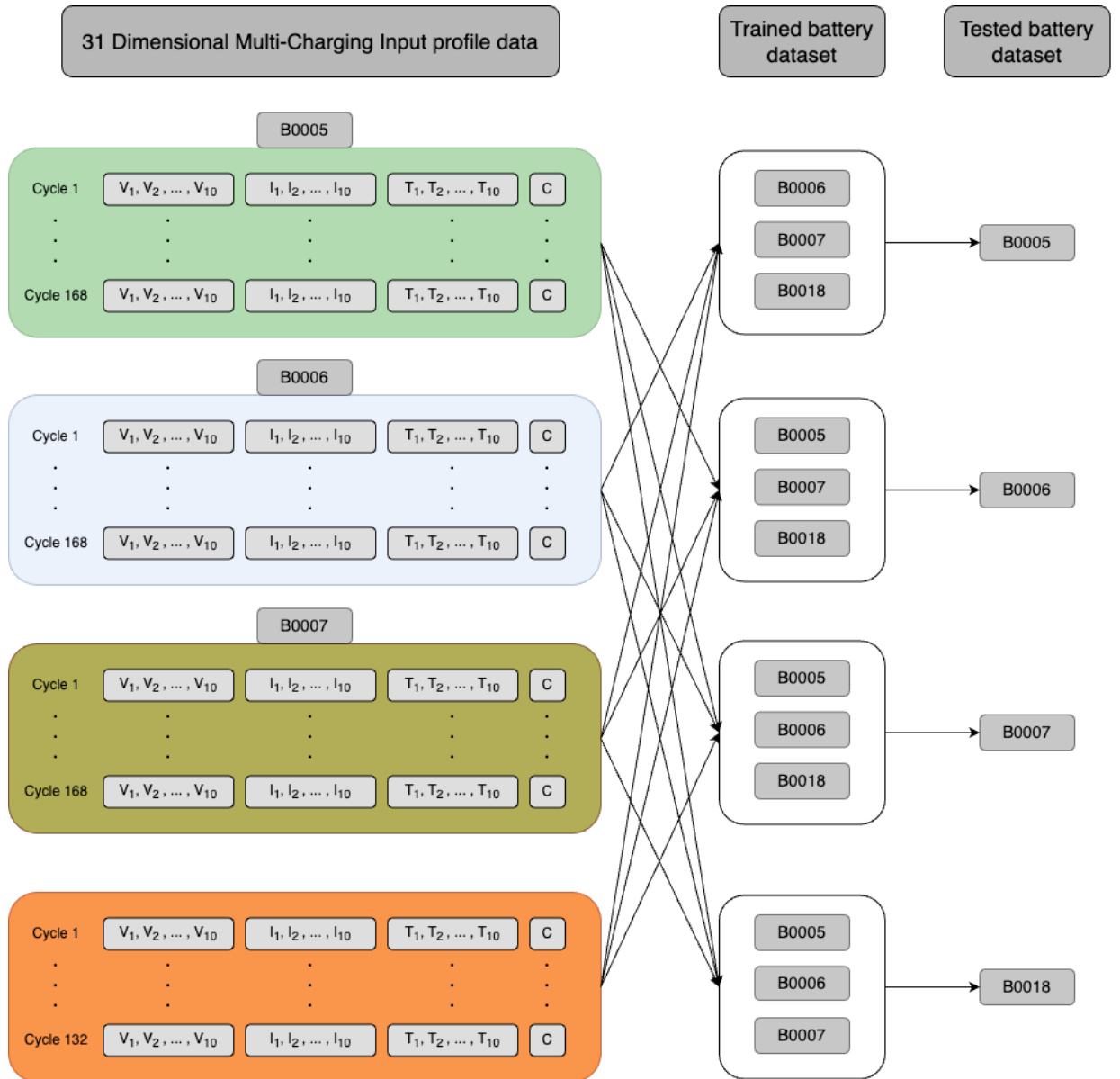


Verranno quindi utilizzati i dataset delle quattro batterie in maniera combinata, in modo da addestrare quattro differenti modelli:

1. Modello B0005: è stato utilizzato il dataset della batteria B0005 come dataset di addestramento e i dataset delle batterie B0006, B0007, B0018 come dataset di training;
2. Modello B0006: è stato utilizzato il dataset della batteria B0006 come dataset di addestramento e i dataset delle batterie B0005, B0007, B0018 come dataset di training;
3. Modello B0007: è stato utilizzato il dataset della batteria B0007 come dataset di addestramento e i dataset delle batterie B0005, B0006, B0018 come dataset di training;
4. Modello B0018: è stato utilizzato il dataset della batteria B0018 come dataset di addestramento e i dataset delle batterie B0005, B0006, B0007 come dataset di training.

Ancora una volta, è stato dunque utilizzato lo stesso profilo MCI adottato da Ansari (Ansari et al., 2021a), in quanto sono stati utilizzati dataset di diverse batterie per addestrare la rete.

Figura 4.2: combinazione dei 4 dataset (B0005, B0006, B0007, B0018) utilizzati per l'implementazione dei 4 modelli LSTM.

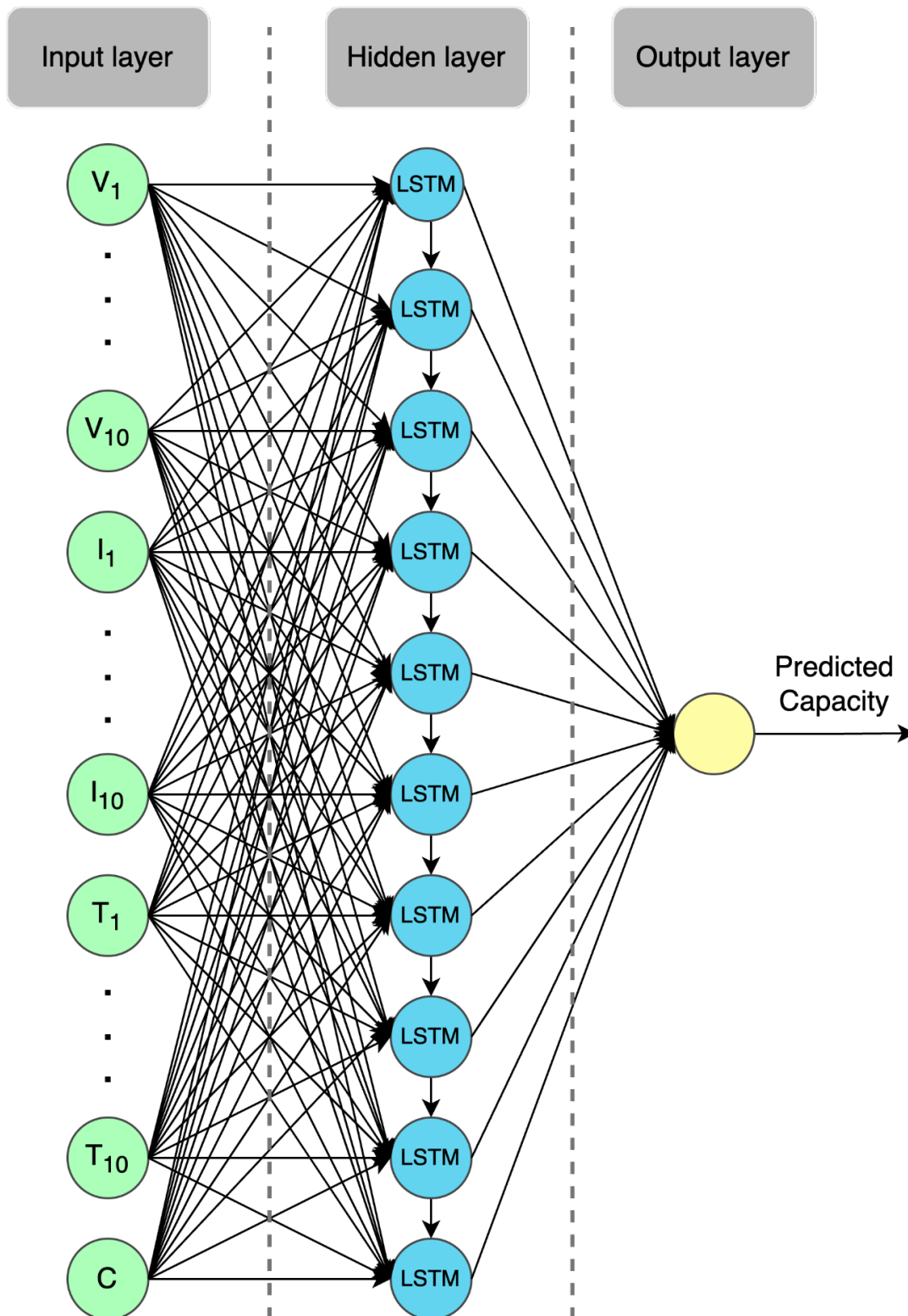


4.3.2 Architettura della rete LSTM

È stata utilizzata l'architettura proposta da Ansari (Ansari et al., 2021a), composta da:

- Un *layer* di input formato da 31 nodi.
- Un *layer* nascosto formato da 10 neuroni LSTM.
- Un *layer* di output composto da un singolo neurone per la stima della capacità della batteria.

Figura 4.3: architettura della rete neurale LSTM.



4.4 Setup sperimentale

I modelli sono stati sviluppati in linguaggio Python versione 3.11.4, tramite la libreria Keras versione 2.13.1, utilizzando la libreria Tensorflow versione 2.13.0 come backend di Keras.

L'addestramento dei modelli è stato effettuato su una macchina MacBook Pro 2020 dotato di chip M1 e 8 GB di RAM.

In particolare⁷:

- La CPU è suddivisa in due cluster da 4 core ciascuno:
 - Il primo cluster (E cluster / Icestorm) è formato da 4 Efficiency Core operativi tra i 0.6 e i 2.064 GHz, e dispongono di una cache di istruzioni da 128 KB e di una cache L2 condivisa da 4 MB,
 - Il secondo cluster (P cluster / Firestorm) è formato da 4 Performance Core operativi tra i 0.6 e i 3.204 GHz, e dispongono di una cache di istruzioni da 192 KB e di una cache L2 condivisa da 12 MB.
- La GPU è composta da 8 Graphics Core, operativi su 8 pipelines a una frequenza di 1.278 GHz, i quali riescono a garantire un throughput fino a 2.6 TFLOPS.

4.5 Training

Per la fase di training, il numero di iterazioni è stato fissato a 20, ed il numero di epoche è stato fissato a 1000, con un learning rate di 0.001. È stato utilizzata la funzione di ottimizzazione Adam, ed è stata utilizzata la funzione Mean Squared Error come funzione di loss.

4.6 Metriche di accuratezza

Di seguito sono riportati per ognuno dei quattro modelli addestrati (B0005, B0006, B0007, B0018) i grafici comparativi della capacità stimata dal modello rispetto alla capacità reale della batteria in analisi.

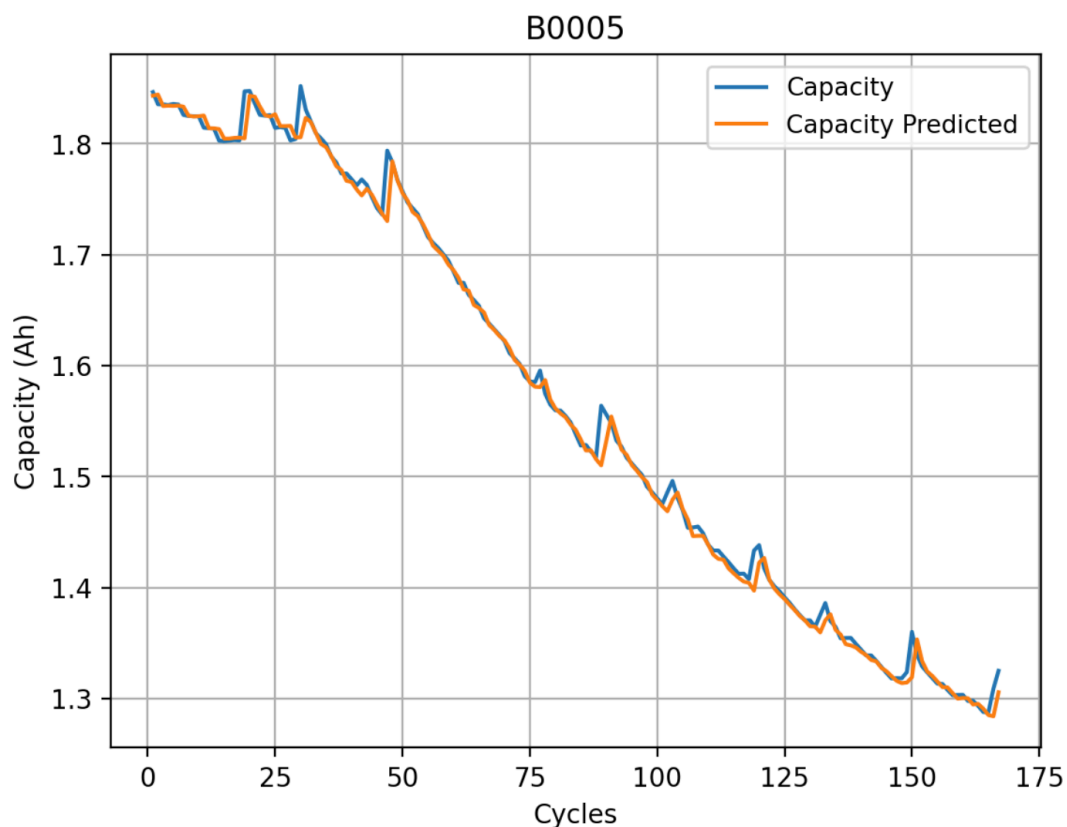
Per ogni modello sono riportate inoltre le metriche di accuratezza calcolate (MSE, RMSE, MAPE, MAE).

⁷ Chip Apple M1: https://it.wikipedia.org/wiki/Apple_M1. Visitato il 21 Luglio 2023.

4.6.1 B0005

In figura 4.4 è riportato il grafico ottenuto a seguito delle stime dei valori di capacità della batteria B0005, effettuate dal modello addestrato con i dataset delle batterie B0006, B0007 e B0018.

Figura 4.4: stima della capacità della batteria B0005.



Nella tabella 4.1 sono riportati i valori delle metriche di accuratezza per le stime dei valori di capacità della batteria B0005, calcolate sul modello addestrato con i dataset delle batterie B0006, B0007 e B0018.

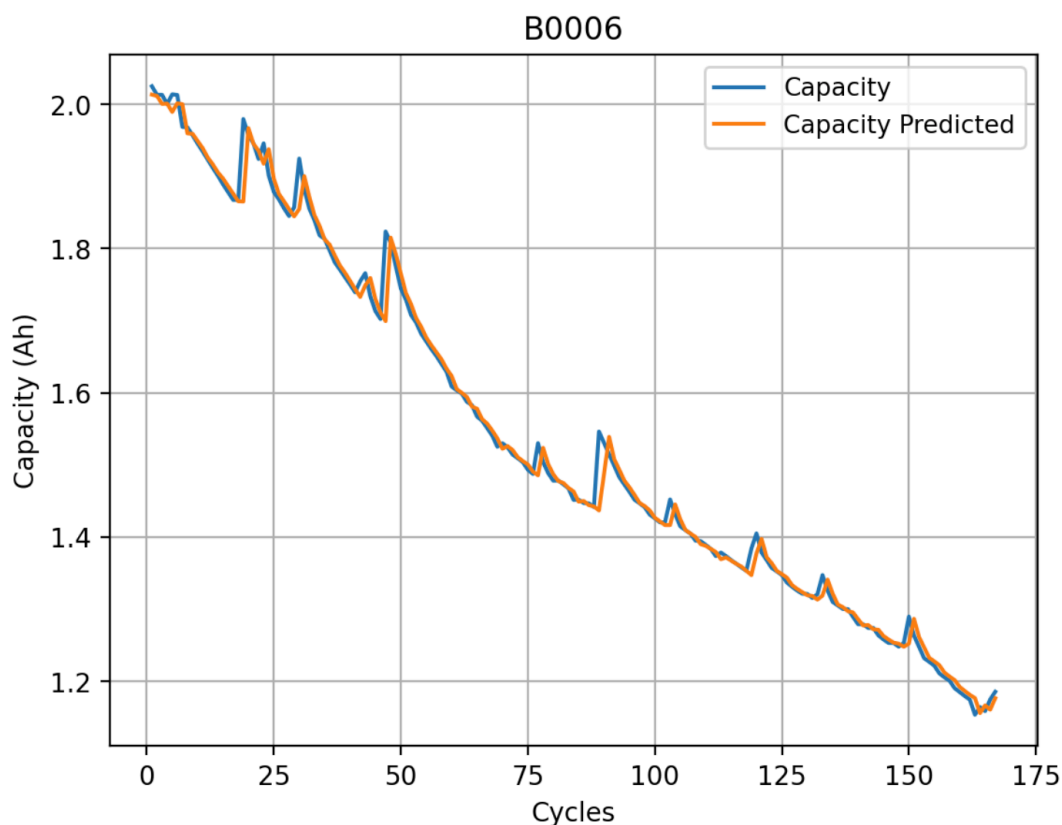
Tabella 4.1: metriche di accuratezza per la stima della capacità della batteria B0005.

MSE	RMSE	MAPE	MAE
0,00011670	0,01080268	0,00361034	0,00561231

4.6.2 B0006

In figura 4.5, è riportato il grafico ottenuto a seguito delle stime dei valori di capacità della batteria B0006, effettuate dal modello addestrato con i dataset delle batterie B0005, B0007 e B0018.

Figura 4.5: stima della capacità della batteria B0006.



Nella tabella 4.2 sono riportati i valori delle metriche di accuratezza per le stime dei valori di capacità della batteria B0006, calcolate sul modello addestrato con i dataset delle batterie B0005, B0007 e B0018.

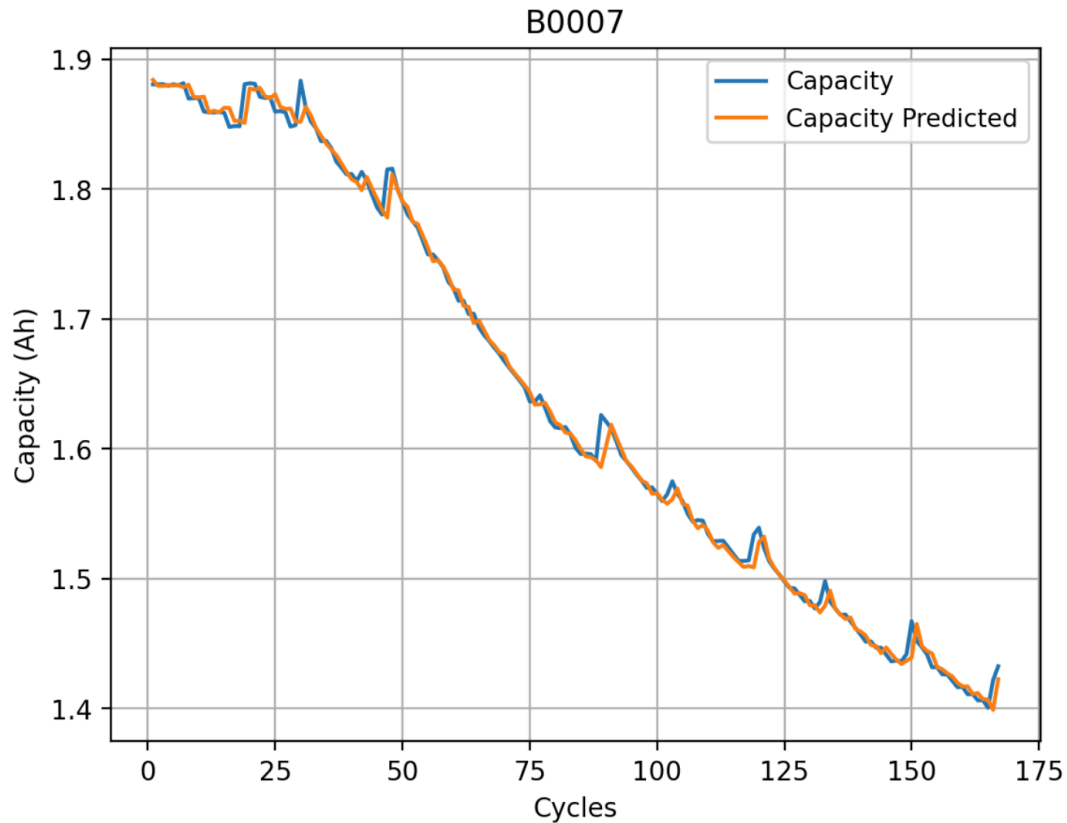
Tabella 4.2: metriche di accuratezza per la stima della capacità della batteria B0006.

MSE	RMSE	MAPE	MAE
0,00041716	0,02042449	0,00697048	0,01102701

4.6.3 B0007

In figura 4.6, viene riportato il grafico ottenuto a seguito delle stime dei valori di capacità della batteria B0007, effettuate dal modello addestrato con i dataset delle batterie B0005, B0006 e B0018.

Figura 4.6: stima della capacità della batteria B0007.



Nella tabella 4.3 sono riportati i valori delle metriche di accuratezza per le stime dei valori di capacità della batteria B0007, calcolate sul modello addestrato con i dataset delle batterie B0005, B0006 e B0018.

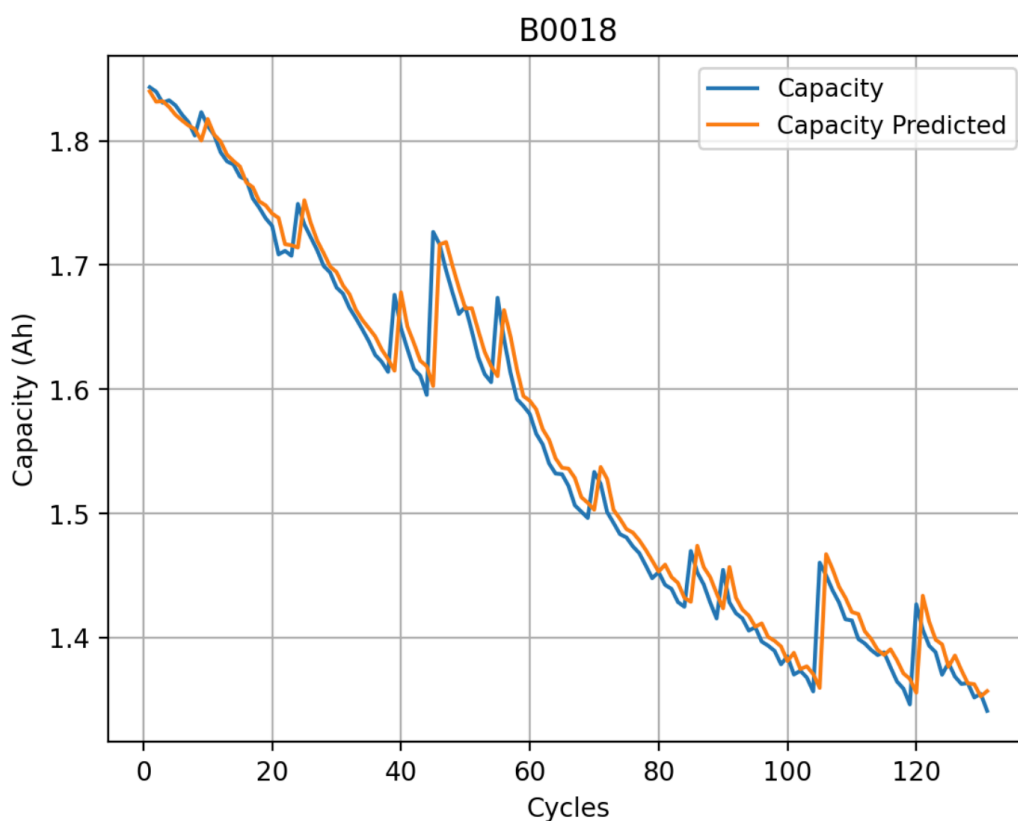
Tabella 4.3: metriche di accuratezza per la stima della capacità della batteria B0007.

MSE	RMSE	MAPE	MAE
0,00006598	0,00812253	0,00297848	0,00486921

4.6.4 B0018

In figura 4.7, è riportato il grafico ottenuto a seguito delle stime dei valori di capacità della batteria B0018, effettuate dal modello addestrato con i dataset delle batterie B0005, B0006 e B0007.

Figura 4.7: stima della capacità della batteria B0018.



Nella tabella 4.4 sono riportati i valori delle metriche di accuratezza per le stime dei valori di capacità della batteria B0018, calcolate sul modello addestrato con i dataset delle batterie B0005, B0006 e B0007.

Tabella 4.4: metriche di accuratezza per la stima della capacità della batteria B0018.

MSE	RMSE	MAPE	MAE
0,00051761	0,02275102	0,01017827	0,01566689

4.6.5 Confronto tra metriche di accuratezza

Di seguito verranno riassunte e messe a confronto le metriche di accuratezza ottenute dal modello LSTM proposto e implementato in questo lavoro di tesi (tabella 4.5), con le metriche di accuratezza ottenute dai modelli proposti e implementati da Park (tabella 4.6) (Park et al., 2020) e Choi (tabella 4.7) (Choi et al., 2019).

Tabella 4.5: metriche di accuratezza del modello LSTM proposto.

Testing dataset	Training dataset	Metrics			
		MSE	RMSE	MAPE	MAE
B0005	B0006 B0007 B0018	0,00011670	0,01080268	0,00361034	0,00561231
B0006	B0005 B0007 B0018	0,00041716	0,02042449	0,00697048	0,01102701
B0007	B0005 B0006 B0018	0,00006598	0,00812253	0,00297848	0,00486921
B0018	B0005 B0006 B0007	0,00051761	0,02275102	0,01017827	0,01566689
Metrics Mean:		0,00027936	0,01552518	0,00593439	0,00929386

Tabella 4.6: metriche di accuratezza del modello proposto da Park (Park et al., 2020).

Testing dataset	Training dataset	Metrics			
		MSE	RMSE	MAPE	MAE
B0005	B0006 B0007 B0018	0,00028224	0,0168	0,0105	0,0146
B0006	B0005 B0007 B0018	0,00023104	0,0152	0,0070	0,0103
B0007	B0005 B0006 B0018	0,00007225	0,0085	0,0047	0,0068
B0018	B0005 B0006 B0007	0,00150544	0,0388	0,0188	0,0261
Metrics Mean:		0,00052274	0,019825	0,01025	0,01445

Confrontando le metriche ottenute dal modello LSTM proposto e implementato in questo lavoro di tesi, con le metriche ottenute dal modello implementato da Park (Park et al., 2020), possiamo notare che il modello LSTM proposto e implementato in questo lavoro di tesi risulta essere più accurato nello stimare la capacità delle batterie B0005, B0007 e B0018.

Il modello implementato da Park (Park et al., 2020) risulta essere più accurato nello stimare la capacità della batteria B0006.

In media, le metriche ottenute dal modello LSTM proposto e implementato in questo lavoro di tesi sono comunque migliori in termini di accuratezza:

- MSE è diminuito del 46,6% (da 0,00052274 a 0,00027936)
- RMSE è diminuito del 21,7% (da 0,019825 a 0,01552518)
- MAPE è diminuito del 42,1% (da 0,01025 a 0,00593439)
- MAE è diminuito del 35,7% (da 0,01445 a 0,00929386)

Tabella 4.7: metriche di accuratezza per il modello proposto da Choi (Choi et al., 2019).

	Metrics			
	MSE	RMSE	MAPE	MAE
Metrics Mean:	0,00060516	0,0246	0,010320	0,0159

Nel lavoro svolto da Choi (Choi et al., 2019) sono riportati solamente i valori riassuntivi delle metriche ottenute con il modello implementato.

Confrontando le metriche ottenute dal modello LSTM proposto e implementato in questo lavoro di tesi, con le metriche ottenute dal modello implementato da Choi (Choi et al., 2019), possiamo notare che il modello LSTM proposto e implementato in questo lavoro di tesi risulta essere più accurato nello stimare la capacità delle batterie B0005, B0006, B0007 e B0018.

Anche in media dunque, le metriche ottenute dal modello LSTM proposto e implementato in questo lavoro di tesi sono migliori in termini di accuratezza:

- MSE è diminuito del 53,8% (da 0,00060516 a 0,00027936)
- RMSE è diminuito del 36,9% (da 0,0246 a 0,01552518)
- MAPE è diminuito del 42,5% (da 0,010320 a 0,00593439)
- MAE è diminuito del 41,5% (da 0,0159 a 0,00929386)

Tabella 4.8: confronto tra le metriche di accuratezza ottenute dal modello LSTM proposto e quelle ottenute da Park (Park et al., 2020) e Choi (Choi et al., 2019).

Model	Parameters	Metrics Mean			
		MSE	RMSE	MAPE	MAE
LSTM proposed	491	0,00027936	0,01552518	0,00593439	0,00929386
LSTM (Park et al., 2020)	2772	0,00052274	0,019825	0,01025	0,01445
LSTM (Choi et al., 2019)	17472	0,00060516	0,0246	0,010320	0,0159

Anche dal punto di vista della complessità della rete, il modello LSTM proposto e implementato in questo lavoro di tesi risulta essere più semplice in termini di numero di parametri su cui effettuare l'addestramento rispetto alle strutture descritte da Park (Park et al., 2020) e Choi (Choi et al., 2019). In particolare il modello LSTM proposto e implementato in questo lavoro di tesi possiede (tabella 4.8):

- 82,3% di parametri in meno rispetto al modello proposto da Park (da 2772 a 491).
- 97,2% di parametri in meno rispetto al modello proposto da Choi (da 17472 a 491).

Abbiamo dunque ottenuto una rete neurale LSTM con la stessa struttura della rete descritta da Ansari (Ansari et al., 2021a), più semplice delle strutture descritte da Park (Park et al., 2020) e Choi (Choi et al., 2019) in termini di numero di parametri su cui effettuare l'addestramento, ma in grado di ottenere gli stessi risultati in termini di accuratezza. Raggiungendo questo obiettivo, abbiamo guadagnato una rete più snella, più veloce nell'elaborazione dell'output, più adatta e versatile per essere distribuita all'interno di BMS con limitate risorse di calcolo e più facile da spiegare tramite le tecniche di Explainability.

Abbiamo quindi risposto in maniera affermativa alla RQ1.

CAPITOLO QUINTO

5. Esperimenti

In questo capitolo verranno descritti gli esperimenti di Explainability che sono stati effettuati sul modello LSTM proposto in questa tesi per la stima della capacità di una batteria al litio.

In particolare verranno testate le seguenti tecniche:

- *SHAP*
- *DeepLIFT*
- *Saliency Map*
- *LRP*

Verrà descritto il setup sperimentale con cui sono state implementate ed eseguite le diverse tecniche di Explainability.

Successivamente, per ogni tecnica di Explainability, verranno descritti i risultati ottenuti dagli esperimenti svolti, con l'obiettivo di rispondere alla RQ2.

5.1 Setup sperimentale

Le tecniche di Explainability sono state sviluppate in linguaggio Python versione 3.11.4.

Le tecniche di Explainability sono poi state eseguite su una macchina MacBook Pro 2020 dotato di chip M1 e 8 GB di RAM.

In particolare⁸:

- La CPU è suddivisa in due cluster da 4 core ciascuno:
 - Il primo cluster (E cluster / Icestorm) è formato da 4 Efficiency Core operativi tra i 0.6 e i 2.064 GHz, e dispongono di una cache di istruzioni da 128 KB e di una cache L2 condivisa da 4 MB,
 - Il secondo cluster (P cluster / Firestorm) è formato da 4 Performance Core operativi tra i 0.6 e i 3.204 GHz, e

⁸ Chip Apple M1: https://it.wikipedia.org/wiki/Apple_M1. Visitato il 21 Luglio 2023.

dispongono di una cache di istruzioni da 192 KB e di una cache L2 condivisa da 12 MB.

- La GPU è composta da 8 Graphics Core, operativi su 8 pipelines a una frequenza di 1.278 GHz, i quali riescono a garantire un throughput fino a 2.6 TFLOPS.

5.2 Explainability

In questa sezione verranno descritti gli esperimenti di Explainability che sono stati effettuati sul modello LSTM proposto in questa tesi per la stima della capacità di una batteria al litio.

In particolare verranno testate le seguenti tecniche:

- *SHAP*
- *DeepLIFT*
- *Saliency Map*
- *LRP*

Per ogni tecnica di Explainability verranno descritti i risultati degli esperimenti svolti.

Da qui in avanti ci riferiremo alle feature utilizzate come input per il modello LSTM proposto ed implementato in questa tesi come segue:

- *volt_i*: tensione misurata dalla batteria durante l'istante *i*-esimo del ciclo di carica;
- *cur_i*: intensità di corrente misurata dalla batteria durante l'istante *i*-esimo del ciclo di carica;
- *temp_i*: temperatura misurata dalla batteria durante l'istante *i*-esimo del ciclo di carica;
- *cap*: capacità della batteria misurata durante il ciclo di scarica precedente;

con *i* che va da 1 a 10, in quanto nella fase di campionamento sistematico l'intervallo di campionamento è stato fissato a 10.

5.2.1 SHAP

Per quanto riguarda la tecnica di Explainability *SHAP*, è stata testata la libreria python *shap/shap*⁹, sviluppata dal dottor Scott Lundberg, uno degli autori del paper di *SHAP* (Lundberg et al., 2017), con l'obbiettivo di riuscire ad applicare tale tecnica sul modello LSTM proposto ed implementato in questa tesi.

In data 21 Luglio 2023 la repository *shap/shap* è nel seguente stato:

- Versione: 0.42.1
- Data ultimo rilascio: 15 Luglio 2023

Per ottenere una panoramica di quali feature si sono dimostrate determinanti nello stimare l'output, è stato redatto un grafico *beeswarm*¹⁰ per ogni batteria, in cui sono stati riportati i valori *SHAP* di ogni feature per ogni campione testato, utilizzando la tecnica *Kernel SHAP* (Lundberg et al., 2017).

I grafici seguenti dunque ordinano le feature in base alla somma dei valori *SHAP* (valori assoluti) su tutti i campioni testati, per poi mostrare la distribuzione dell'impatto che ogni feature ha avuto sull'output del modello. Il colore rappresenta il valore della singola feature (rosso significa valore alto, blu significa valore basso).

Una feature con colore rosso nella metà di destra sta a significare che la crescita del valore della feature stessa contribuisce alla stima di un valore alto dell'output del modello. Analogamente, una feature con colore blu nella metà di sinistra sta a significare che la diminuzione del valore della feature stessa contribuisce alla stima di un valore basso dell'output del modello.

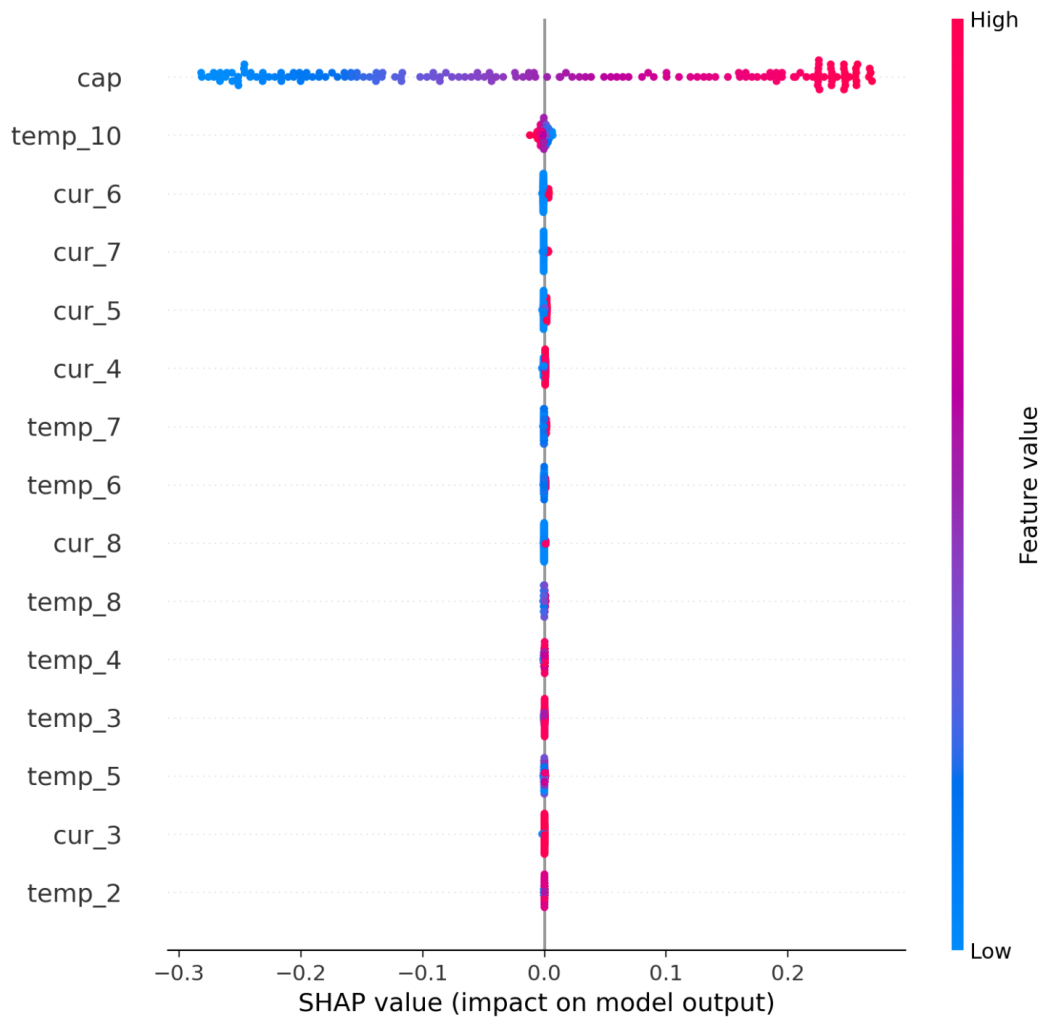
⁹ Libreria *shap/shap*: <https://github.com/shap/shap>. Visitato il 21 Luglio 2023.

¹⁰ Grafico *beeswarm* per i valori *SHAP*:
https://shap.readthedocs.io/en/latest/example_notebooks/api_examples/plots/beeswarm.html.
Visitato il 21 Luglio 2023.

5.2.1.1 B0005

Per la batteria B0005, utilizzando tutti i suoi dati come background, e visualizzando le 15 feature più rilevanti (che hanno contribuito maggiormente al risultato di output), abbiamo ottenuto il grafico *beeswarm* mostrato in figura 5.1.

Figura 5.1: grafico *beeswarm* per i valori *SHAP* della batteria B0005.



Dal grafico *beeswarm* mostrato in figura 5.1, notiamo che la feature *cap* (capacità del ciclo di scarica precedente della batteria) è la feature che maggiormente contribuisce alla stima del valore di capacità di output del modello. In particolare, come possiamo vedere dal grafico sopra, un valore alto della capacità del ciclo di scarica precedente contribuisce alla stima di un valore di capacità alto da parte del modello. Analogamente, un valore

basso della capacità del ciclo di scarica precedente contribuisce alla stima di un valore di capacità basso da parte del modello.

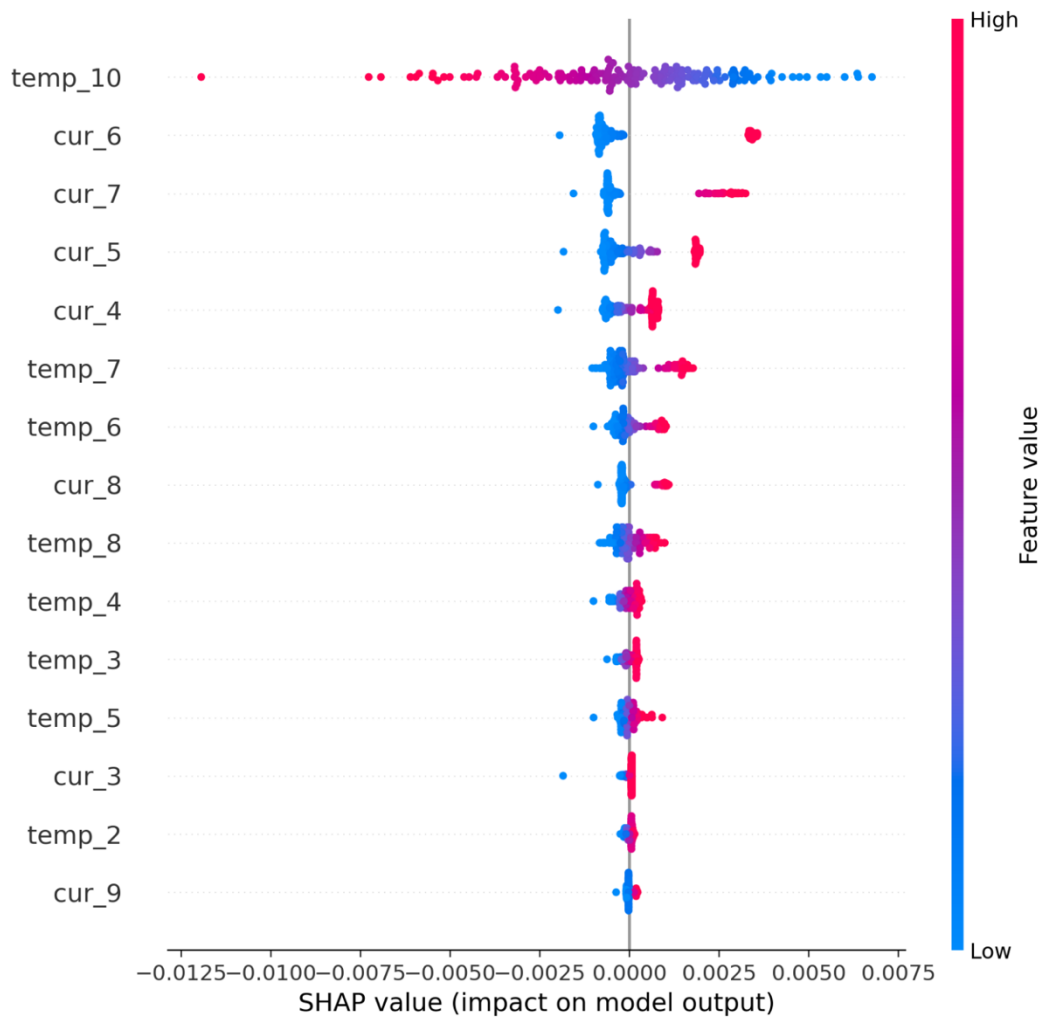
In secondo luogo, la feature *temp_10* (temperatura della batteria misurata al decimo e ultimo istante di tempo del ciclo di carica precedente) è la seconda feature in termini di rilevanza nella stima della capacità della batteria.

Ancora, scendendo verso il basso in termini di rilevanza, troviamo le feature *cur_6*, *cur_7*, *cur_5*, *cur_4* (intensità di corrente misurata agli istanti di tempo 6, 7, 5, e 4 rispettivamente).

Notiamo poi che la rilevanza della feature *cap* nel determinare la capacità di output è così determinante (rosso molto scuro) che è difficile comprendere a colpo d'occhio la rilevanza delle altre feature nel determinare la capacità di output.

Consideriamo dunque la mappa di calore mostrata in figura 5.2, in cui è stata esclusa dalla valutazione la feature *cap*.

Figura 5.2: grafico *beeswarm* per i valori *SHAP* della batteria B0005, escludendo la feature *cap*.



Dal grafico *beeswarm* mostrato in figura 5.2, possiamo osservare con maggior chiarezza l’impatto che le altre feature hanno nel determinare la capacità di output della batteria in analisi.

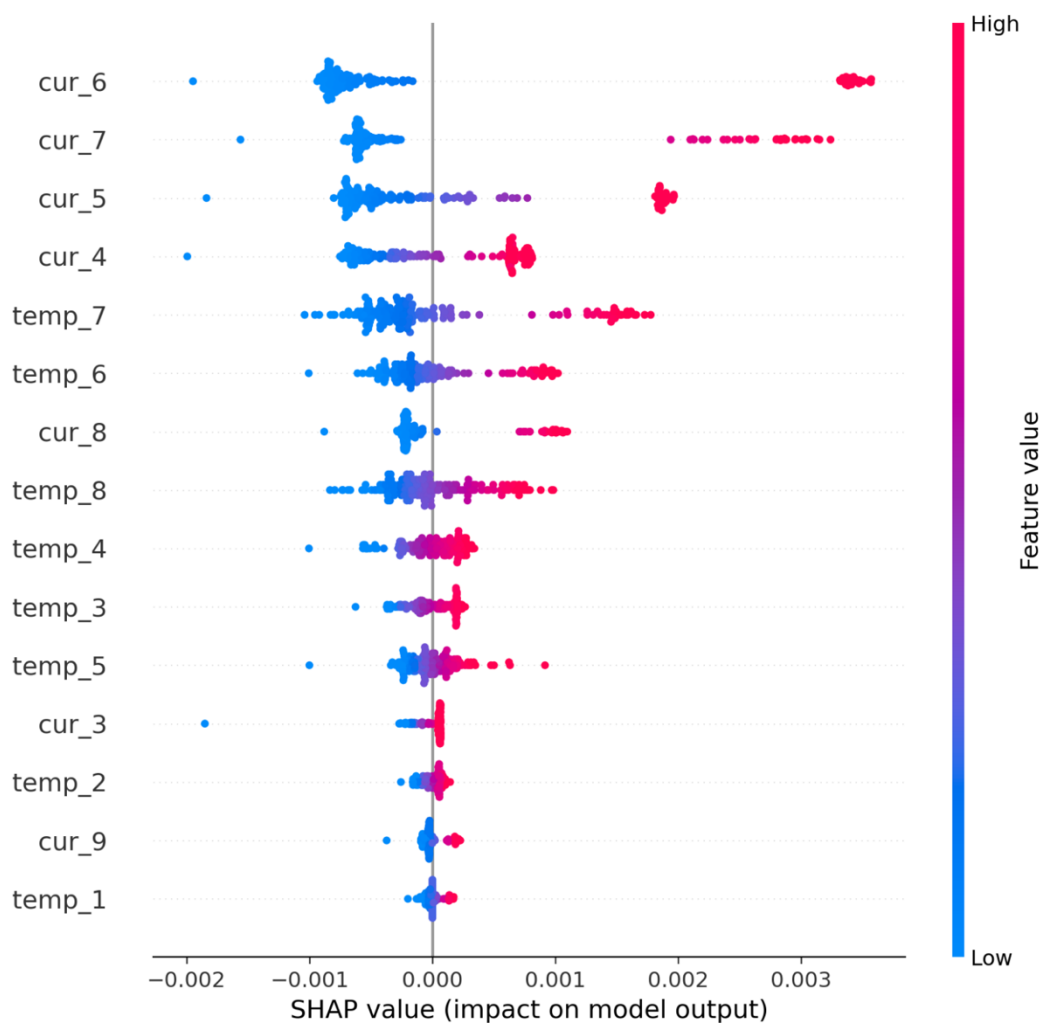
In primo luogo, notiamo che la feature *temp_10* (temperatura della batteria misurata al decimo e ultimo istante di tempo del ciclo di carica precedente) è la seconda feature in termini di rilevanza nella stima della capacità della batteria; più alto è il valore di questa feature, più basso sarà il valore di capacità stimato dal modello.

Ancora, scendendo verso il basso in termini di rilevanza, troviamo le feature *cur_6*, *cur_7*, *cur_5*, *cur_4* (intensità di corrente misurata agli istanti di tempo 6, 7, 5, e 4 rispettivamente); notiamo che un valore di intensità di

corrente alto negli istanti di tempo 6, 7, 5 e 4 contribuisce alla stima di un valore di capacità alto da parte del modello.

Analogamente, possiamo infine escludere dal grafico *beeswarm* anche la feature *temp_10*, per osservare con maggior chiarezza gli impatti che le altre feature hanno nel determinare la capacità di output della batteria in analisi, come mostrato in figura 5.3.

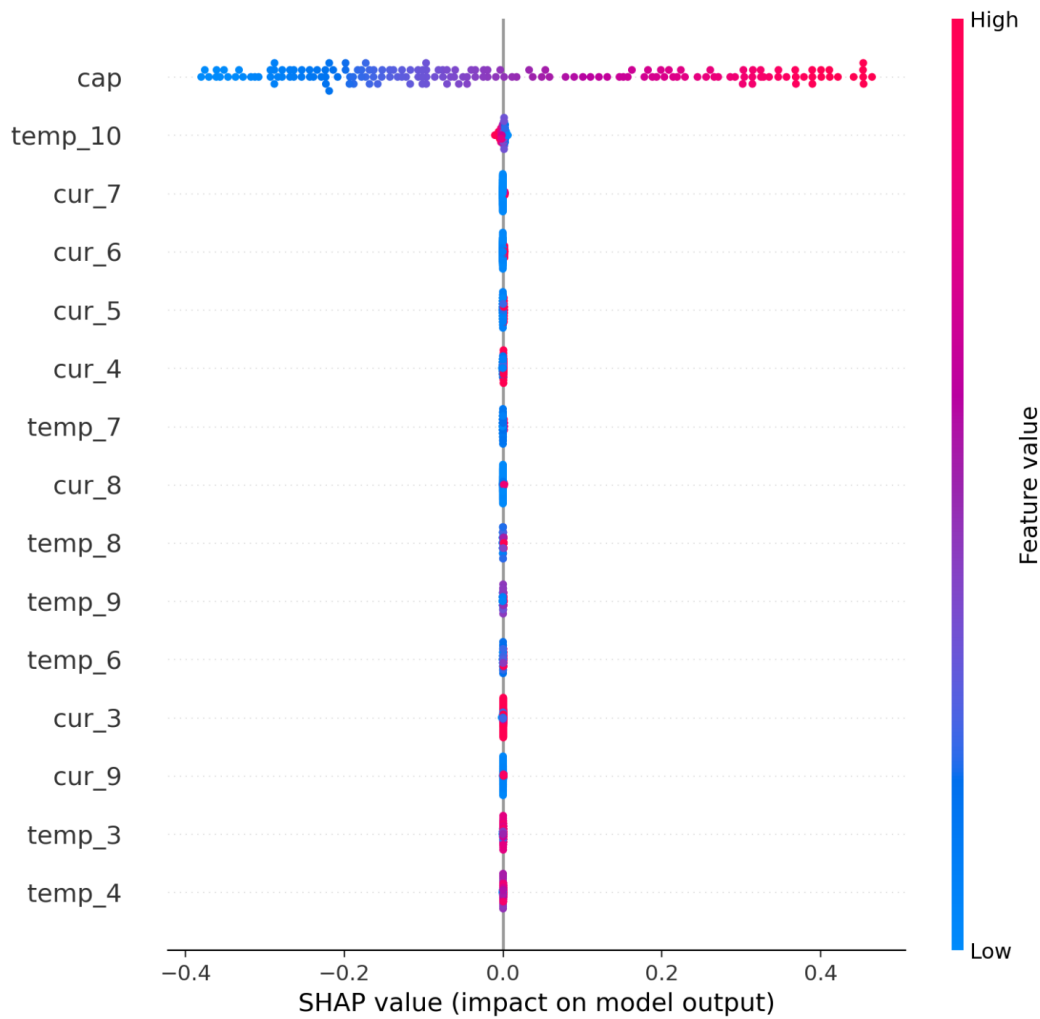
Figura 5.3: grafico *beeswarm* per i valori *SHAP* della batteria B0005, escludendo le feature *cap* e *temp_10*.



5.2.1.2 B0006

Per la batteria B0006, utilizzando tutti i suoi dati come background, e visualizzando le 15 feature più rilevanti (che hanno contribuito maggiormente al risultato di output), abbiamo ottenuto il grafico mostrato in figura 5.4.

Figura 5.4: grafico *beeswarm* per i valori *SHAP* della batteria B0006.



Dal grafico *beeswarm* mostrato in figura 5.4 notiamo che la feature *cap* (capacità del ciclo di scarica precedente della batteria) è la feature che maggiormente contribuisce alla stima del valore di capacità di output del modello. In particolare, un valore alto della capacità del ciclo di scarica precedente contribuisce alla stima di un valore di capacità alto da parte del modello. Analogamente, un valore basso della capacità del ciclo di scarica

precedente contribuisce alla stima di un valore di capacità basso da parte del modello.

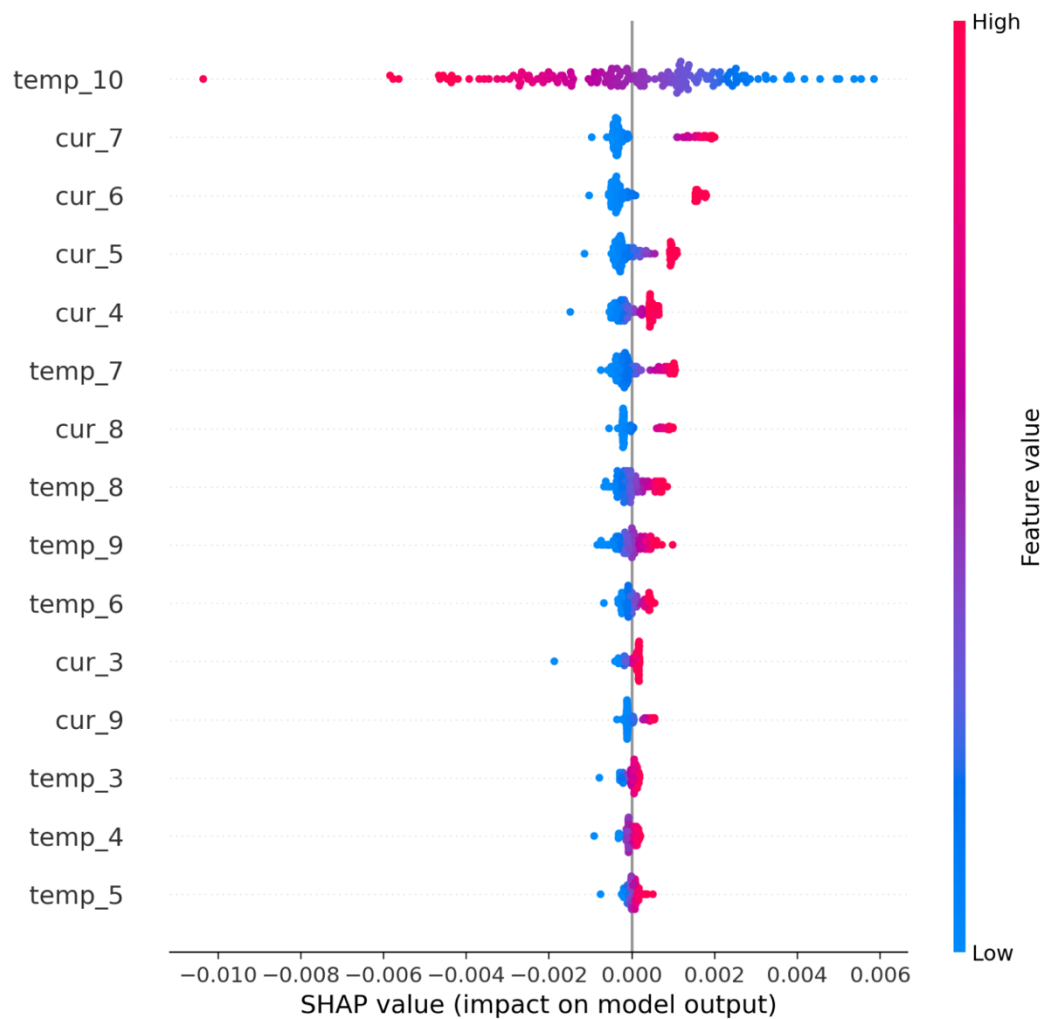
La feature *temp_10* (temperatura della batteria misurata al decimo e ultimo istante di tempo del ciclo di carica precedente) è la seconda feature in termini di rilevanza nella stima della capacità della batteria; più alto è il valore di questa feature, più basso sarà il valore di capacità stimato dal modello.

Ancora, scendendo verso il basso in termini di rilevanza, troviamo le feature *cur_7*, *cur_6*, *cur_5*, *cur_4* (corrente misurata agli istanti di tempo 6, 7, 5, e 4 rispettivamente); notiamo che un valore di corrente alto agli istanti di tempo 6, 7, 5 e 4 contribuisce alla stima di un valore di capacità alto da parte del modello.

Notiamo poi che la rilevanza della feature *cap* nel determinare la capacità di output è così determinante che è difficile comprendere a colpo d'occhio la rilevanza delle altre feature nel determinare la capacità di output.

Consideriamo dunque la mappa di calore mostrata in figura 5.5, in cui è stata esclusa dalla valutazione la feature *cap*.

Figura 5.5: grafico *beeswarm* per i valori *SHAP* della batteria B0005, escludendo la feature *cap*.



Dal grafico *beeswarm* mostrato in figura 5.5, possiamo osservare con maggior chiarezza l’impatto che le altre feature hanno nel determinare la capacità di output della batteria in analisi.

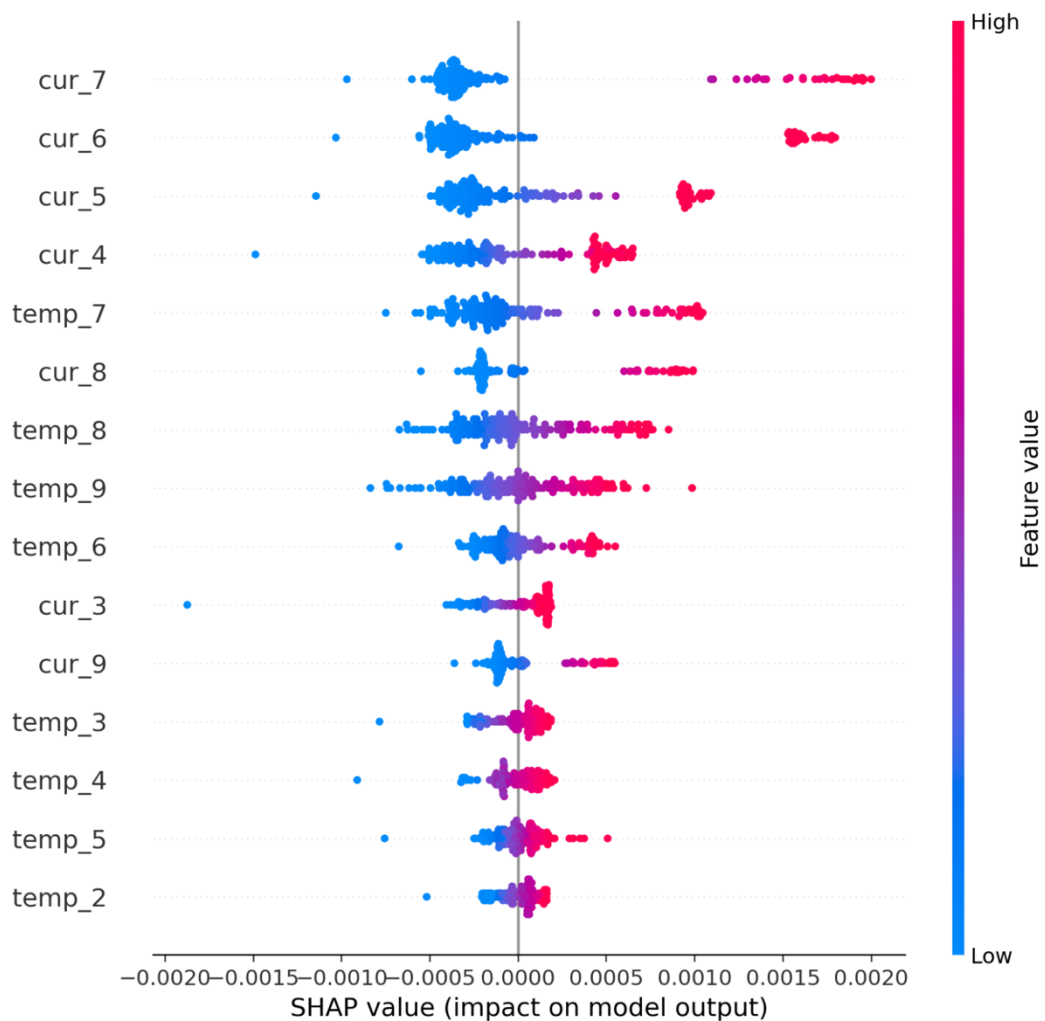
La feature *temp_10* (temperatura della batteria misurata al decimo e ultimo istante di tempo del ciclo di carica precedente) è la seconda feature in termini di rilevanza nella stima della capacità della batteria; più alto è il valore di questa feature, più basso sarà il valore di capacità stimato dal modello.

Ancora, scendendo verso il basso in termini di rilevanza, troviamo le feature *cur_7*, *cur_6*, *cur_5*, *cur_4* (corrente misurata agli istanti di tempo 6, 7, 5, e 4 rispettivamente); notiamo che un valore di corrente alto agli istanti di tempo

6, 7, 5 e 4 contribuisce alla stima di un valore di capacità alto da parte del modello.

Analogamente, possiamo infine escludere dal grafico *beeswarm* anche la feature *temp_10*, per osservare con maggior chiarezza gli impatti che le altre feature hanno nel determinare la capacità di output della batteria in analisi, come mostrato in figura 5.6.

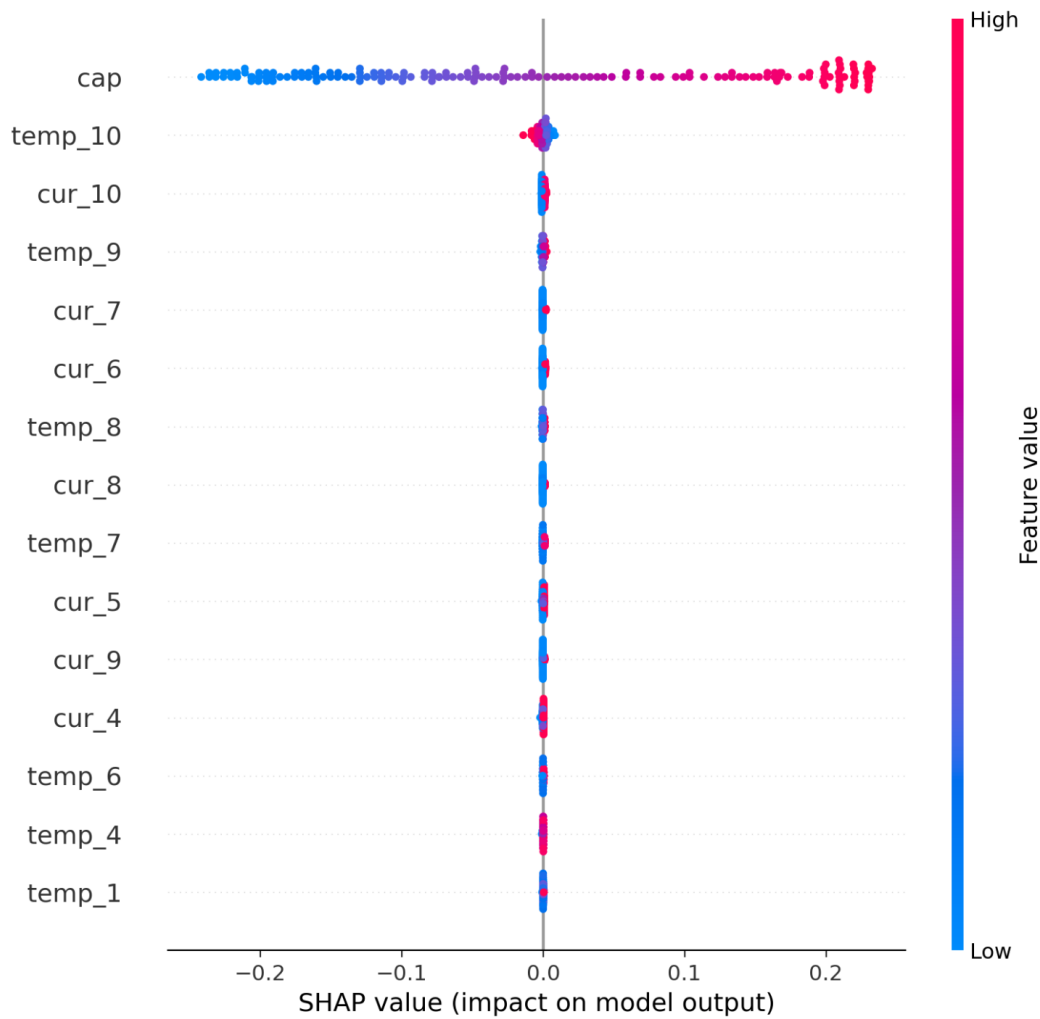
Figura 5.6: grafico *beeswarm* per i valori *SHAP* della batteria B0005, escludendo le feature *cap* e *temp_10*.



5.2.1.3 B0007

Per la batteria B0007, utilizzando tutti i suoi dati come background, e visualizzando le 15 feature più rilevanti (che hanno contribuito maggiormente al risultato di output), abbiamo ottenuto il grafico mostrato in figura 5.7.

Figura 5.7: grafico *beeswarm* per i valori *SHAP* della batteria B0007.



Dal grafico *beeswarm* mostrato in figura 5.7 notiamo che la feature *cap* (capacità del ciclo di scarica precedente della batteria) è la feature che maggiormente contribuisce alla stima del valore di capacità di output del modello. In particolare, un valore alto della capacità del ciclo di scarica precedente contribuisce alla stima di un valore di capacità alto da parte del modello. Analogamente, un valore basso della capacità del ciclo di scarica

precedente contribuisce alla stima di un valore di capacità basso da parte del modello.

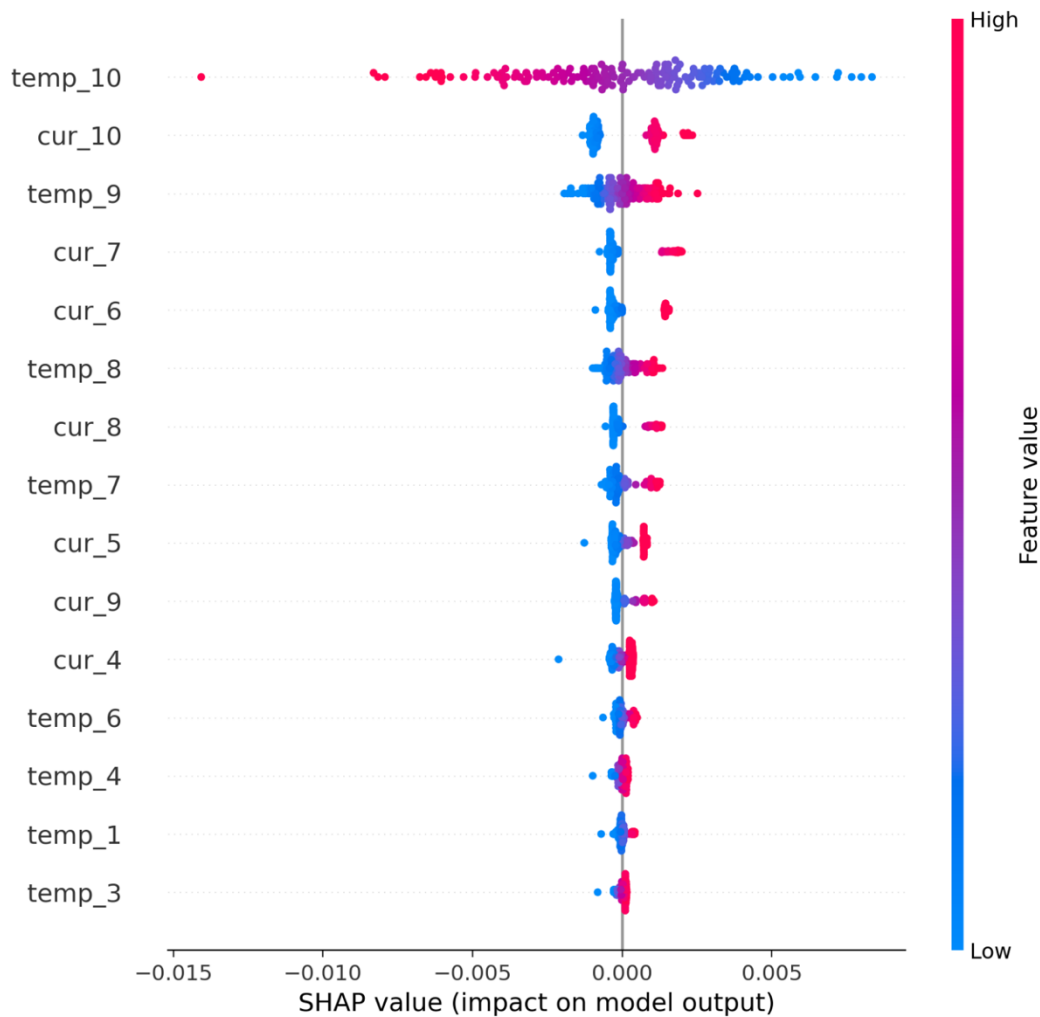
La feature *temp_10* (temperatura della batteria misurata al decimo e ultimo istante di tempo del ciclo di carica precedente) è la seconda feature in termini di rilevanza nella stima della capacità della batteria.

Scendendo verso il basso in termini di rilevanza, troviamo poi le feature *cur_10*, *temp_9*, *cur_7*, *cur_6* (corrente misurata all'istante di tempo 10, temperatura misurata all'istante di tempo 9, e correnti misurate agli istanti di tempo 7 e 6 rispettivamente).

Notiamo poi che la rilevanza della feature *cap* nel determinare la capacità di output è così determinante che è difficile comprendere a colpo d'occhio la rilevanza delle altre feature nel determinare la capacità di output.

Consideriamo dunque la mappa di calore mostrata in figura 5.8, in cui è stata esclusa dalla valutazione la feature *cap*.

Figura 5.8: grafico *beeswarm* per i valori *SHAP* della batteria B0005, escludendo la feature *cap*.



Dal grafico *beeswarm* mostrato in figura 5.8, possiamo osservare con maggior chiarezza l’impatto che le altre feature hanno nel determinare la capacità di output della batteria in analisi.

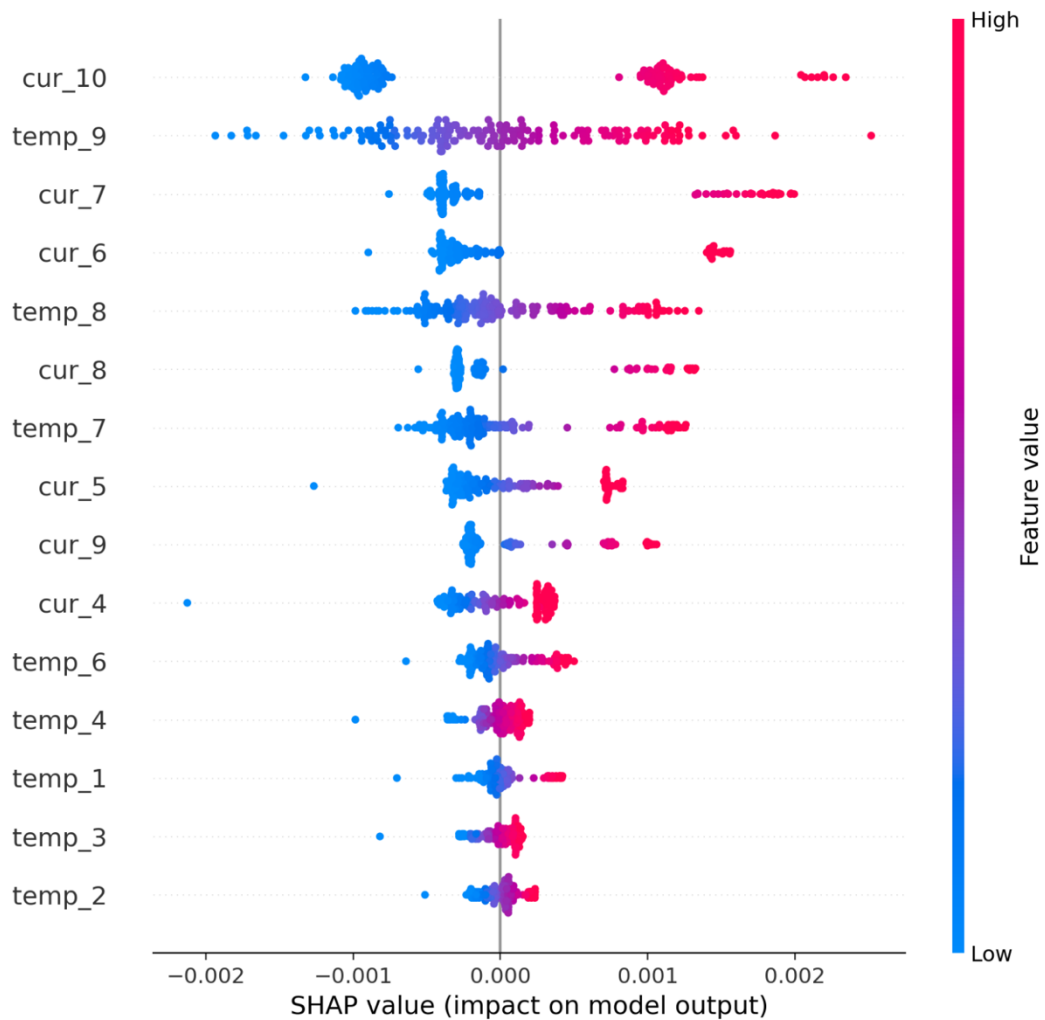
In primo luogo, notiamo che la feature *temp_10* (temperatura della batteria misurata al decimo e ultimo istante di tempo del ciclo di carica precedente) è la seconda feature in termini di rilevanza nella stima della capacità della batteria; più alto è il valore di questa feature, più basso sarà il valore di capacità stimato dal modello.

Scendendo verso il basso in termini di rilevanza, troviamo poi le feature *cur_10*, *temp_9*, *cur_7*, *cur_6* (corrente misurata all’istante di tempo 10, temperatura misurata all’istante di tempo 9, e correnti misurate agli istanti di

tempo 7 e 6 rispettivamente); notiamo che un valore alto di ognuna di queste feature contribuisce alla stima di un valore di capacità alto da parte del modello.

Analogamente, possiamo infine escludere dal grafico *beeswarm* anche la feature *temp_10*, per osservare con maggior chiarezza gli impatti che le altre feature hanno nel determinare la capacità di output della batteria in analisi, come mostrato in figura 5.9.

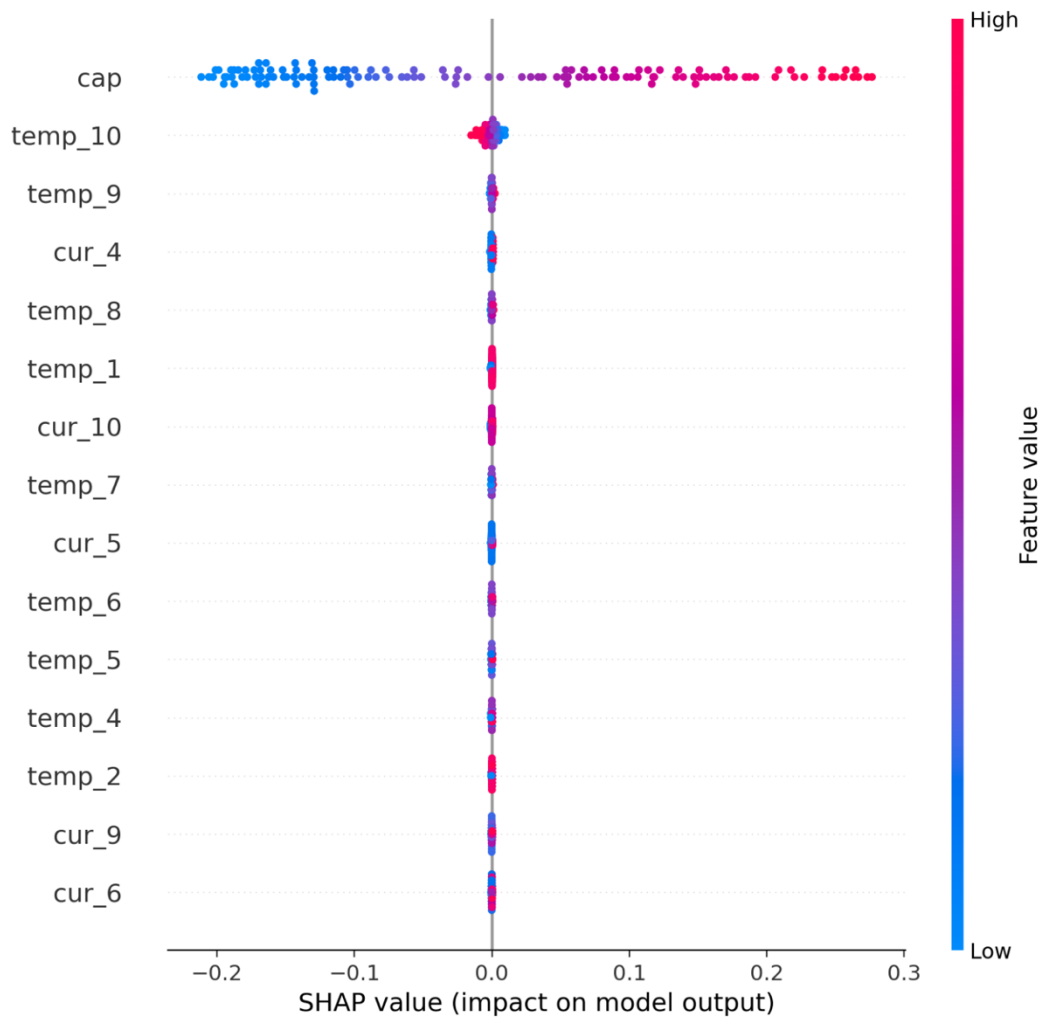
Figura 5.9: grafico *beeswarm* per i valori *SHAP* della batteria B0005, escludendo le feature *cap* e *temp_10*.



5.2.1.4 B0018

Infine, per la batteria B0018, utilizzando tutti i suoi dati come background, e visualizzando le 15 feature più rilevanti (che hanno contribuito maggiormente al risultato di output), abbiamo ottenuto il grafico mostrato in figura 5.10.

Figura 5.10: grafico *beeswarm* per i valori *SHAP* della batteria B0018.



Dal grafico *beeswarm* mostrato in figura 5.10 notiamo che la feature *cap* (capacità del ciclo di scarica precedente della batteria) è la feature che maggiormente contribuisce alla stima del valore di capacità di output del modello. In particolare, un valore alto della capacità del ciclo di scarica precedente contribuisce alla stima di un valore di capacità alto da parte del modello. Analogamente, un valore basso della capacità del ciclo di scarica

precedente contribuisce alla stima di un valore di capacità basso da parte del modello.

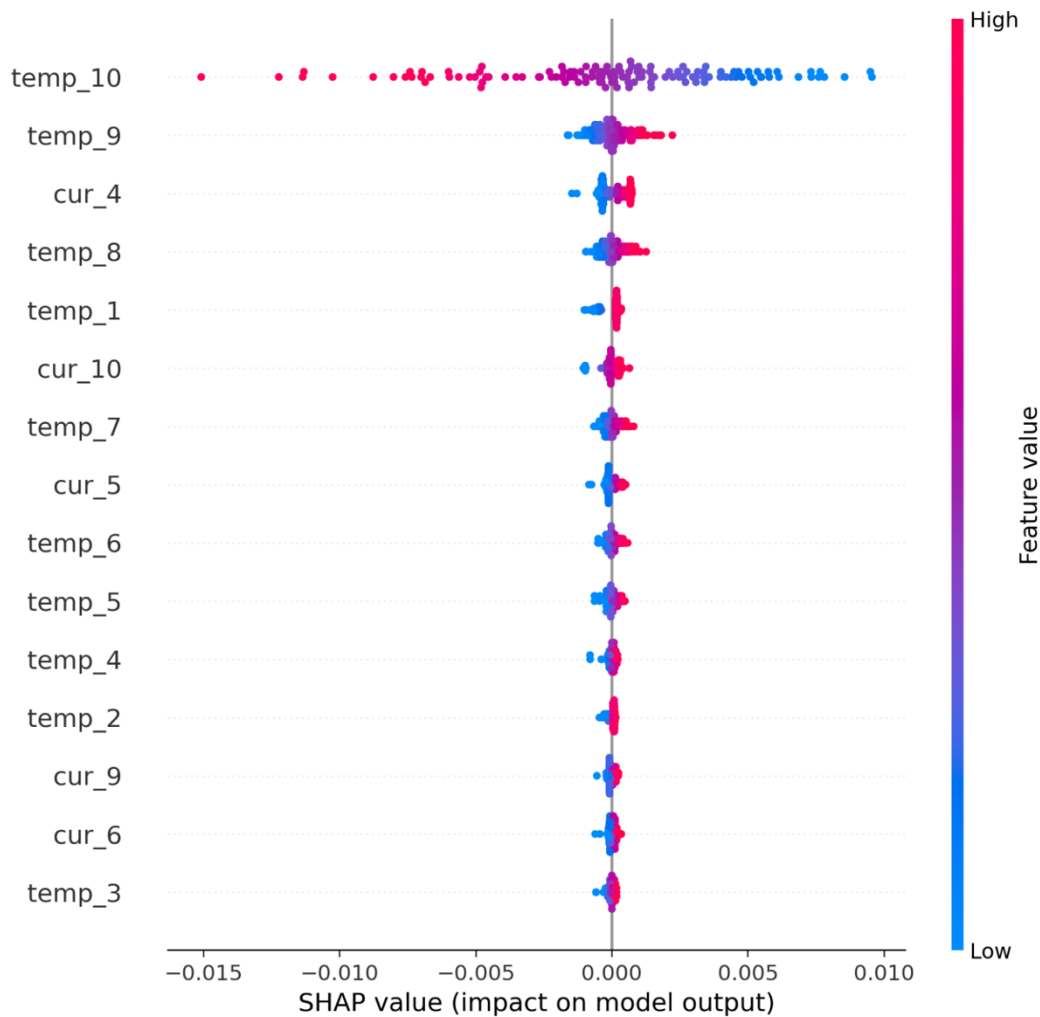
Di nuovo, la feature *temp_10* (temperatura della batteria misurata al decimo e ultimo istante di tempo del ciclo di carica precedente) è la seconda feature in termini di rilevanza nella stima della capacità della batteria.

Scendendo verso il basso in termini di rilevanza, troviamo questa volta le feature *temp_9*, *cur_4*, *temp_8*, *temp_1* (temperatura misurata all'istante di tempo 9, corrente misurata all'istante di tempo 4, e temperature misurate agli istanti di tempo 8 e 1 rispettivamente).

Notiamo poi che la rilevanza della feature *cap* nel determinare la capacità di output è così determinante che è difficile comprendere a colpo d'occhio la rilevanza delle altre feature nel determinare la capacità di output.

Consideriamo dunque la mappa di calore mostrata in figura 5.11, in cui è stata esclusa dalla valutazione la feature *cap*.

Figura 5.11: grafico *beeswarm* per i valori *SHAP* della batteria B0005, escludendo la feature *cap*.



Dal grafico *beeswarm* mostrato in figura 5.11, possiamo osservare con maggior chiarezza l’impatto che le altre feature hanno nel determinare la capacità di output della batteria in analisi.

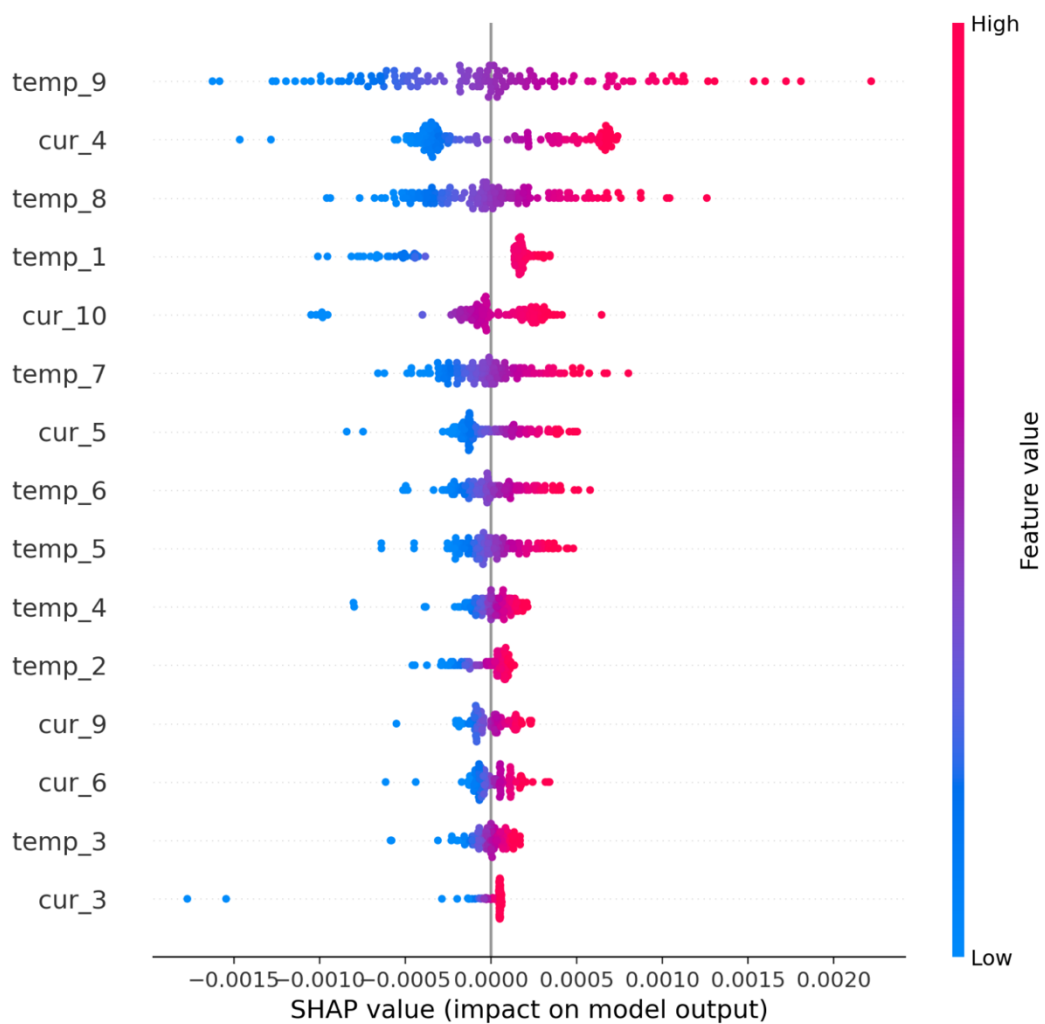
In primo luogo, notiamo che la feature *temp_10* (temperatura della batteria misurata al decimo e ultimo istante di tempo del ciclo di carica precedente) è la seconda feature in termini di rilevanza nella stima della capacità della batteria; più alto è il valore di questa feature, più basso sarà il valore di capacità stimato dal modello.

Scendendo verso il basso in termini di rilevanza, troviamo questa volta le feature *temp_9*, *cur_4*, *temp_8*, *temp_1* (temperatura misurata all’istante di tempo 9, corrente misurata all’istante di tempo 4, e temperature misurate agli

istanti di tempo 8 e 1 rispettivamente); notiamo che un valore alto di ognuna di queste feature contribuisce alla stima di un valore di capacità alto da parte del modello.

Analogamente, possiamo infine escludere dal grafico *beeswarm* anche la feature *temp_10*, per osservare con maggior chiarezza gli impatti che le altre feature hanno nel determinare la capacità di output della batteria in analisi, come mostrato in figura 5.12.

Figura 5.12: grafico *beeswarm* per i valori *SHAP* della batteria B0005, escludendo le feature *cap* e *temp_10*.



5.2.2 DeepLIFT

Per quanto riguarda la tecnica di Explainability *DeepLIFT*, è stata testata la libreria python *kundajelab/deeplift*¹¹, sviluppata dal dottor Anshul Kundaje, uno degli autori del paper di *DeepLIFT* (Shrikumar et al., 2017), con l'obiettivo di riuscire ad applicare tale tecnica sul modello LSTM proposto ed implementato in questa tesi per la stima della capacità di una batteria al litio.

In data 21 Luglio 2023, la repository *kundajelab/deeplift* è nel seguente stato:

- Versione: 0.6.13.0
- Data ultimo rilascio: 11 Novembre 2020.

Nella home page della repository *kundajelab/deeplift*, l'autore riporta che questa libreria è stata sviluppata e testata usando Keras versione 2.2.4 e Tensorflow versione 1.14.0. Le versioni di entrambe le librerie sono considerate obsolete in ambito tecnico e di sviluppo.

Sempre l'autore riporta poi che questa libreria funziona solamente per modelli sequenziali che possiedono al loro interno layer di tipo dense, conv1d, conv2d, e con funzioni di attivazioni di tipo lineare, relu, sigmoide, softmax, prelu.

Provando ad applicare la tecnica di Explainability *DeepLIFT* tramite la libreria *kundajelab/deeplift* sul modello LSTM proposto ed implementato in questa tesi, sono sorte effettivamente sia le problematiche relative alle versioni obsolete di Keras e Tensorflow, sia le problematiche relative al mancato supporto della libreria per i layer di tipo LSTM.

Dato che nel nostro setup sperimentale utilizziamo una versione di Keras moderna (2.13.1) e una versione di Tensorflow moderna (2.13.0), e dato il nostro obiettivo di applicare *DeepLIFT* su una rete neurale con layer LSTM, non è stato possibile applicare la tecnica di Explainability *DeepLIFT* tramite la libreria *kundajelab/deeplift* sul modello LSTM proposto ed implementato in questa tesi.

¹¹ Libreria *kundajelab/deeplift*: <https://github.com/kundajelab/deeplift>. Visitato il 21 Luglio 2023.

Sempre nella home page della repository *kundajelab/deeplift*, Kundaje cita altre 2 librerie che implementano la tecnica *DeepLIFT*: *marcoancona/DeepExplain* e *shap/shap*. Sono quindi state testate anche queste 2 librerie, con l'obbiettivo di riuscire ad applicare la tecnica di Explainability *DeepLIFT* sul modello LSTM proposto ed implementato in questa tesi.

Come indicato da Kundaje, è stata quindi testata la libreria python *marcoancona/DeepExplain*¹², che implementa diverse tecniche di Explainability, fra cui *DeepLIFT*, fornendo un framework unificato per i metodi di Explainability basati sul gradiente e sulla perturbazione.

In data 21 Luglio 2023, la repository *marcoancona/DeepExplain* è nel seguente stato:

- Non sono ancora presenti versioni ufficiali della libreria.
- L'ultima modifica risale al 26 Agosto 2020.

Nella home page della libreria, l'autore riporta che attualmente è supportata solamente Tensorflow versione 1.x. Questa versione di Tensorflow è considerata obsoleta in ambito tecnico e di sviluppo.

L'autore riporta inoltre un riferimento a una versione sperimentale della libreria, in cui è stata introdotta la compatibilità con Tensorflow versione 2.x. Tuttavia, l'autore stesso deve ancora consolidare questa versione della libreria da lui sviluppata, spiegando che ci sono ancora bug legati a problemi di compatibilità con il moderno Tensorflow 2.x.

Provando ad applicare la tecnica di Explainability *DeepLIFT* tramite la libreria *marcoancona/DeepExplain* sul modello LSTM proposto ed implementato in questa tesi, sono sorte effettivamente le problematiche relative alla versione obsoleta di Tensorflow. Anche testando la libreria sperimentale compatibile con Tensorflow 2.x, sono sorte le problematiche preannunciate dall'autore stesso.

¹² Libreria *marcoancona/DeepExplain*: <https://github.com/marcoancona/DeepExplain>. Visitato il 21 Luglio 2023.

Dato che nel nostro setup sperimentale utilizziamo una versione di Tensorflow moderna (2.13.0), non è stato possibile applicare la tecnica di Explainability *DeepLIFT* tramite la libreria *marcoancona/DeepExplain* sul modello LSTM proposto ed implementato in questa tesi.

Come indicato da Kundaje, è stata poi testata la libreria python *shap/shap*¹³, già utilizzata nei paragrafi precedenti per la tecnica di Explainability *SHAP*, la quale implementa anche la tecnica di Explainability *DeepLIFT*.

In data 21 Luglio 2023 la repository *shap/shap* è nel seguente stato:

- Versione: 0.42.1
- Data ultimo rilascio: 15 Luglio 2023

Nella pagina web della libreria *shap/shap*, sono riportati diversi problemi di compatibilità con le versioni moderne di Tensorflow per quanto riguarda la tecnica di Explainability *DeepLIFT* implementata; in particolare è riportato che la tecnica *DeepLIFT* implementata nella libreria attualmente supporta solamente Tensorflow versione 1.x. Questa versione di Tensorflow è considerata obsoleta in ambito tecnico e di sviluppo.

Provando ad applicare la tecnica di Explainability *DeepLIFT* tramite la libreria *shap/shap* sul modello LSTM proposto ed implementato in questa tesi, sono sorte effettivamente le problematiche relative alla versione obsoleta di Tensorflow.

Dato che nel nostro setup sperimentale utilizziamo una versione di Tensorflow moderna (2.13.0), non è stato possibile applicare la tecnica di Explainability *DeepLIFT* tramite la libreria *shap/shap* sul modello LSTM proposto ed implementato in questa tesi.

In conclusione, a causa del mancato supporto e aggiornamento delle librerie disponibili sul web che implementano tecniche di Explainability *DeepLIFT*, non è stato possibile effettuare esperimenti di Explainability con la tecnica *DeepLIFT* sul modello LSTM proposto ed implementato in questo lavoro di

¹³ Libreria *shap/shap*: <https://github.com/shap/shap>. Visitato il 21 Luglio 2023.

tesi utilizzando come setup sperimentale un ambiente recente, con Python versione 3.11.4, Keras versione 2.13.1 e Tensorflow versione 2.13.0.

5.2.3 Saliency Map

Per quanto riguarda la tecnica di Explainability *Saliency Map*, è stata testata la libreria python *keisen/tf-keras-vis*¹⁴, con l'obiettivo di applicare tale tecnica di Explainability sul modello LSTM proposto ed implementato in questa tesi.

La libreria python *keisen/tf-keras-vis* fornisce gli strumenti per effettuare debug su una rete neurale Keras, permettendo di visualizzare in maniera intuitiva alcuni aspetti delle reti neurali implementando tecniche di Explainability note. In particolare la libreria fornisce gli strumenti per generare *Saliency Map*. La libreria è stata creata principalmente per fornire strumenti di Explainability per task di classificazione che prendono come input immagini. Le reti neurali di esempio proposte dall'autore della libreria sono tutte reti neurali convoluzionali (*Convolutional Neural Networks*).

Questa libreria è stata quindi testata con l'obiettivo di ottenere tali risultati anche su task di regressione, in particolare sul modello LSTM proposto per la stima della capacità di una batteria.

In data 21 Luglio 2023 la repository *keisen/tf-keras-vis* è nel seguente stato:

- Versione: 0.8.5
- Data ultimo rilascio: 16 Maggio 2023

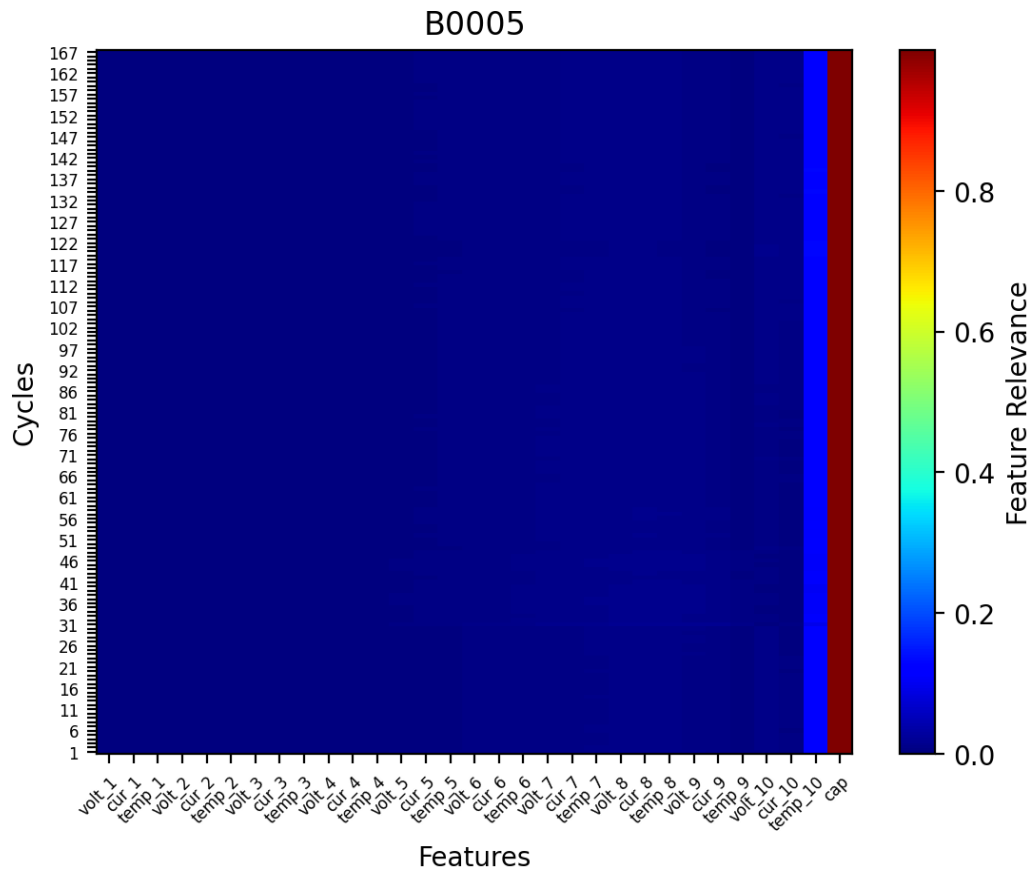
L'obiettivo del test è quello di riuscire a visualizzare tramite una mappa di calore redatta tramite la tecnica di Explainability *Saliency Map* quali siano le feature che maggiormente hanno contribuito nell'andare a stimare la capacità della batteria.

¹⁴ Libreria *keisen/tf-keras-vis*: <https://github.com/keisen/tf-keras-vis>. Visitato il 21 Luglio 2023.

5.2.3.1 B0005

Per la batteria B0005, visualizzando per ogni ciclo la rilevanza di ogni feature nel determinare l'output, abbiamo ottenuto tramite la tecnica *Saliency Map* la mappa di calore mostrata in figura 5.13.

Figura 5.13: mappa di calore generata tramite la tecnica *Saliency Map* per la batteria B0005.



Osservando la mappa di calore mostrata in figura 5.13, notiamo innanzitutto che la feature *cap* (capacità del ciclo di scarica precedente della batteria) è la feature che maggiormente contribuisce alla stima del valore di capacità di output del modello.

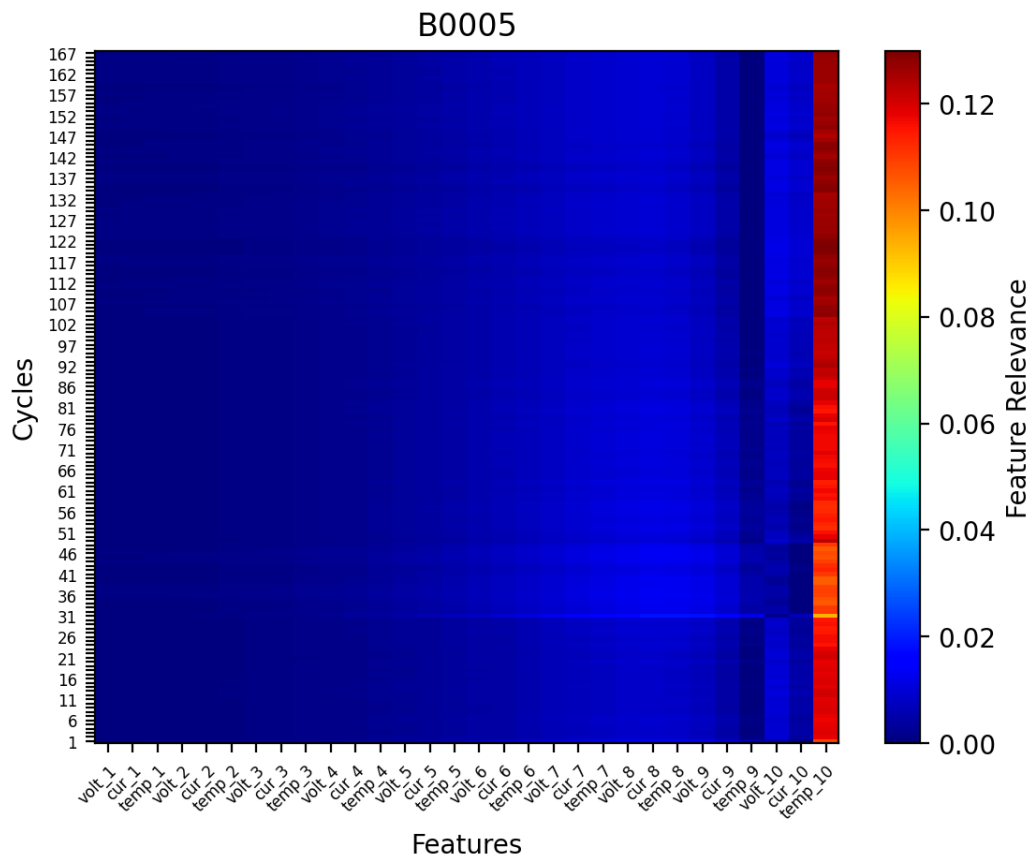
Proseguendo, la feature *temp_10* (temperatura della batteria misurata al decimo e ultimo istante di tempo del ciclo di carica precedente) è la seconda feature in termini di rilevanza nella stima della capacità della batteria.

Notiamo poi che la rilevanza della feature *cap* nel determinare la capacità di output è così determinante (rosso molto scuro) che è difficile comprendere a

colpo d'occhio la rilevanza delle altre feature nel determinare la capacità di output.

Consideriamo dunque la mappa di calore mostrata in figura 5.14, in cui è stata esclusa dalla valutazione la feature *cap*.

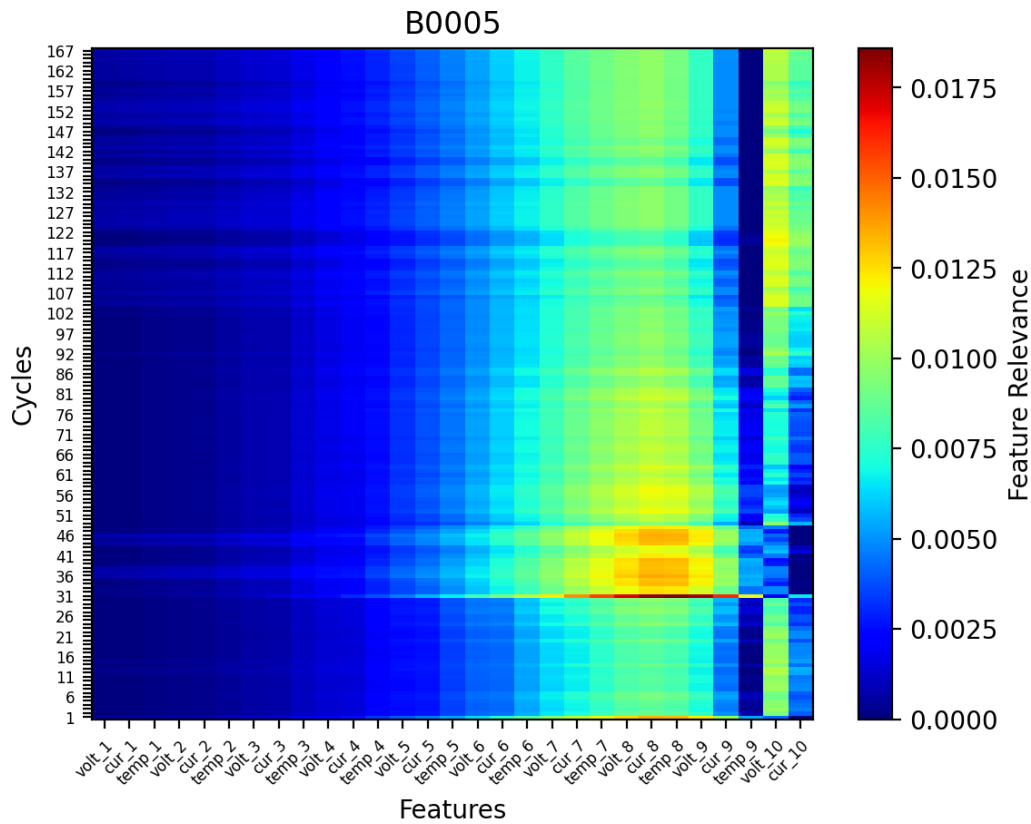
Figura 5.14: mappa di calore generata tramite la tecnica *Saliency Map* per la batteria B0005, escludendo la feature *cap*.



Notiamo dalla mappa di calore mostrata in figura 5.14 che la rilevanza della feature *temp_10* nel determinare la capacità di output è così determinante (rosso scuro) che è difficile comprendere a colpo d'occhio la rilevanza delle altre feature nel determinare la capacità di output.

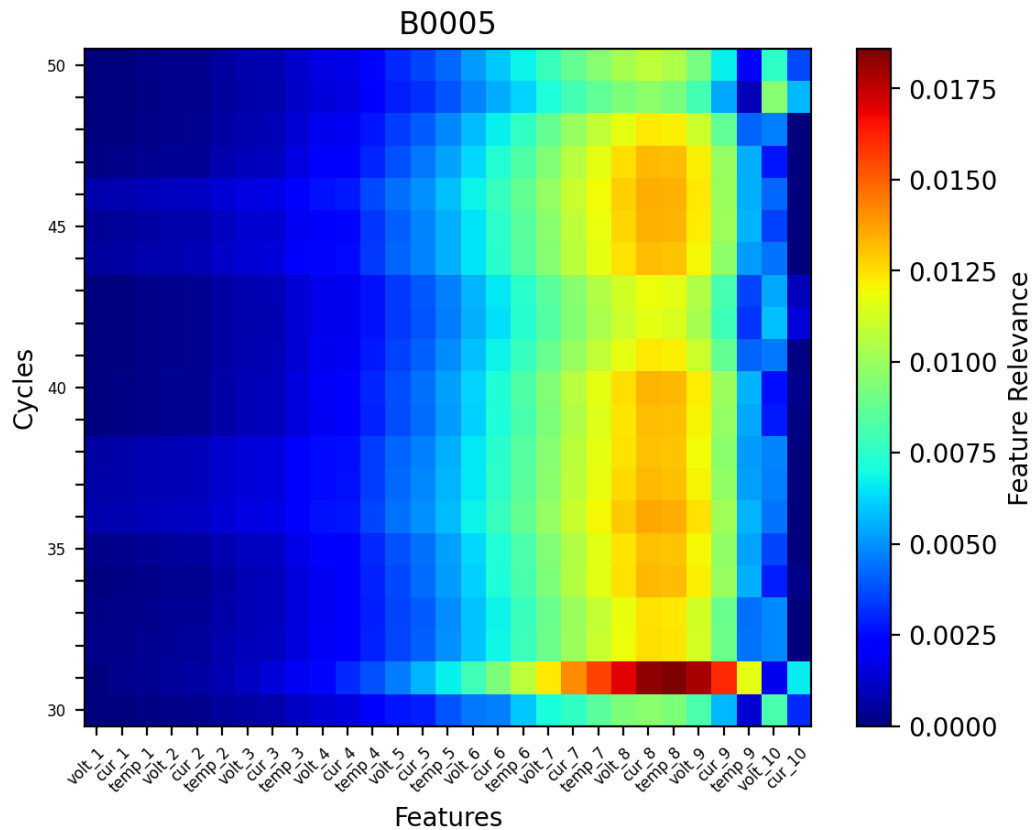
Consideriamo quindi la mappa di calore mostrata in figura 5.15, in cui è stata esclusa dalla valutazione anche la feature *temp_10*.

Figura 5.15: mappa di calore generata tramite la tecnica *Saliency Map* per la batteria B0005, escludendo le feature *cap* e *temp_10*.



Dalla mappa di calore mostrata in figura 5.15, possiamo notare che le feature *volt_7*, *cur_7*, *temp_7*, *volt_8*, *cur_8*, *temp_8* hanno anch'esse una certa rilevanza nel determinare la capacità di output. Notiamo poi che nei cicli da 30 a 50 circa, la rilevanza delle feature *volt_8*, *cur_8*, *temp_8* cresce. Andiamo quindi ad osservare meglio questo fenomeno, concentrandoci sui cicli da 30 a 50 della batteria in analisi (figura 5.16).

Figura 5.16: mappa di calore generata tramite la tecnica *Saliency Map* per la batteria B0005, escludendo le feature *cap* e *temp_10*, per i cicli da 30 a 50.

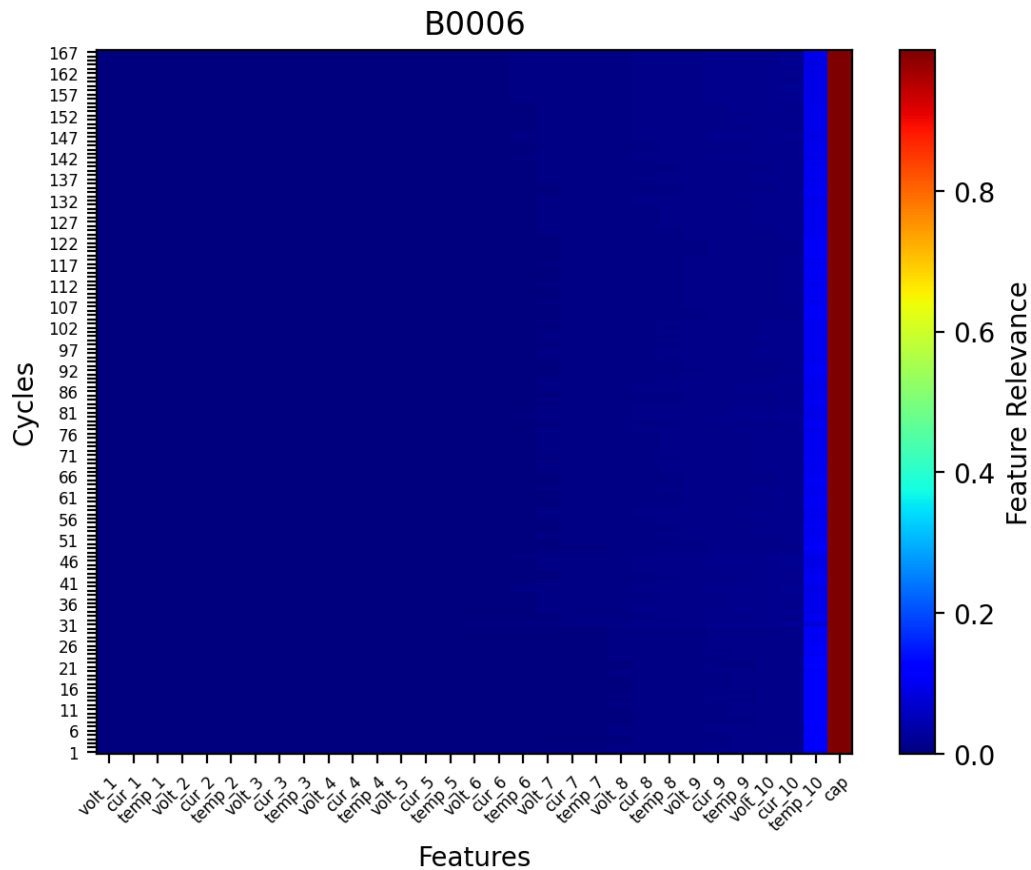


La mappa di calore mostrata in figura 5.16 mette in evidenza alcuni aspetti interessanti. Dal ciclo 31 al ciclo 41, e dal ciclo 44 al ciclo 48, la rilevanza delle feature *volt_8*, *cur_8*, *temp_8* nel determinare la capacità di output cresce. In particolare, nel ciclo 31, la rilevanza delle feature *volt_7*, *cur_7*, *temp_7*, *volt_8*, *cur_8*, *temp_8* è ancora più marcata nel determinare l'output. Anche nel primo ciclo, la rilevanza delle feature sopra citate è decisamente più marcata.

5.2.3.2 B0006

Per la batteria B0006, visualizzando per ogni ciclo la rilevanza di ogni feature nel determinare l'output, abbiamo ottenuto tramite la tecnica *Saliency Map* la mappa di calore mostrata in figura 5.17.

Figura 5.17: mappa di calore generata tramite la tecnica *Saliency Map* per la batteria B0006.



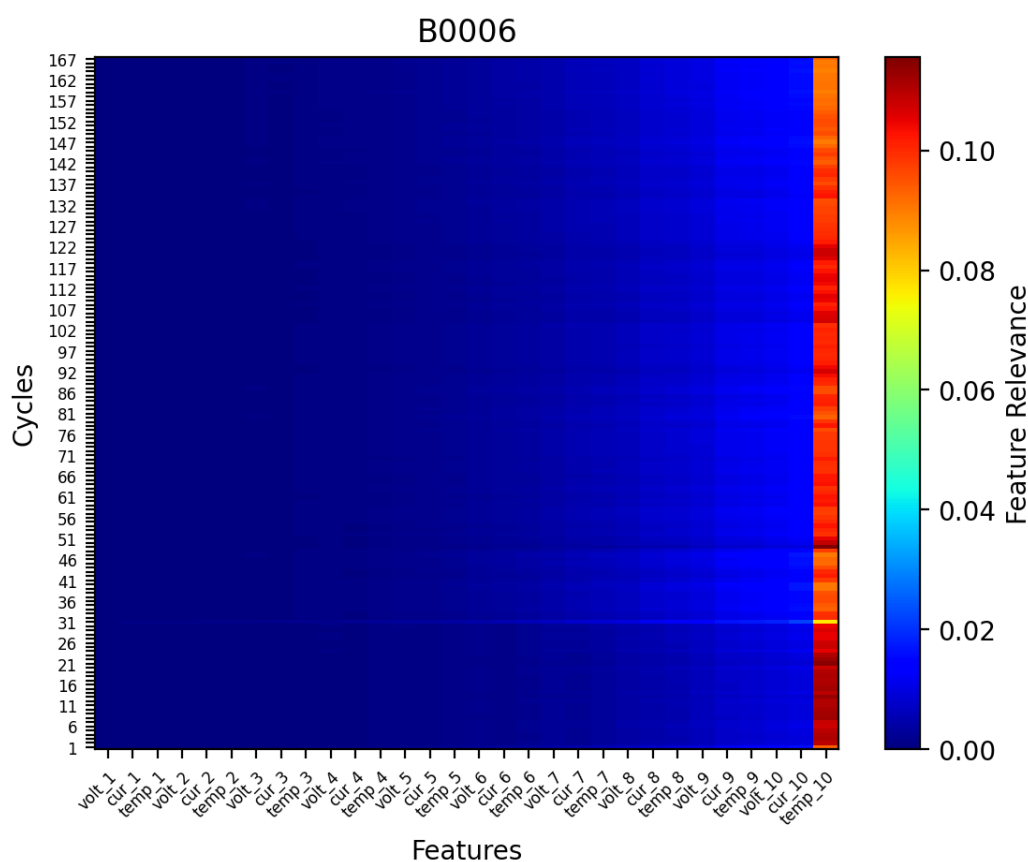
Osservando la mappa di calore mostrata in figura 5.17, notiamo innanzitutto che la feature *cap* (capacità del ciclo di scarica precedente della batteria) è la feature che maggiormente contribuisce alla stima del valore di capacità di output del modello.

In secondo luogo, la feature *temp_10* (temperatura della batteria misurata al decimo e ultimo istante di tempo del ciclo di carica precedente) è di nuovo la seconda feature in termini di rilevanza nella stima della capacità della batteria.

Notiamo poi che ancora una volta la rilevanza della feature *cap* nel determinare la capacità di output è così determinante (rosso molto scuro) che è difficile comprendere a colpo d'occhio la rilevanza delle altre feature nel determinare la capacità di output.

Consideriamo dunque la mappa di calore mostrata in figura 5.18, in cui è stata esclusa dalla valutazione la feature *cap*.

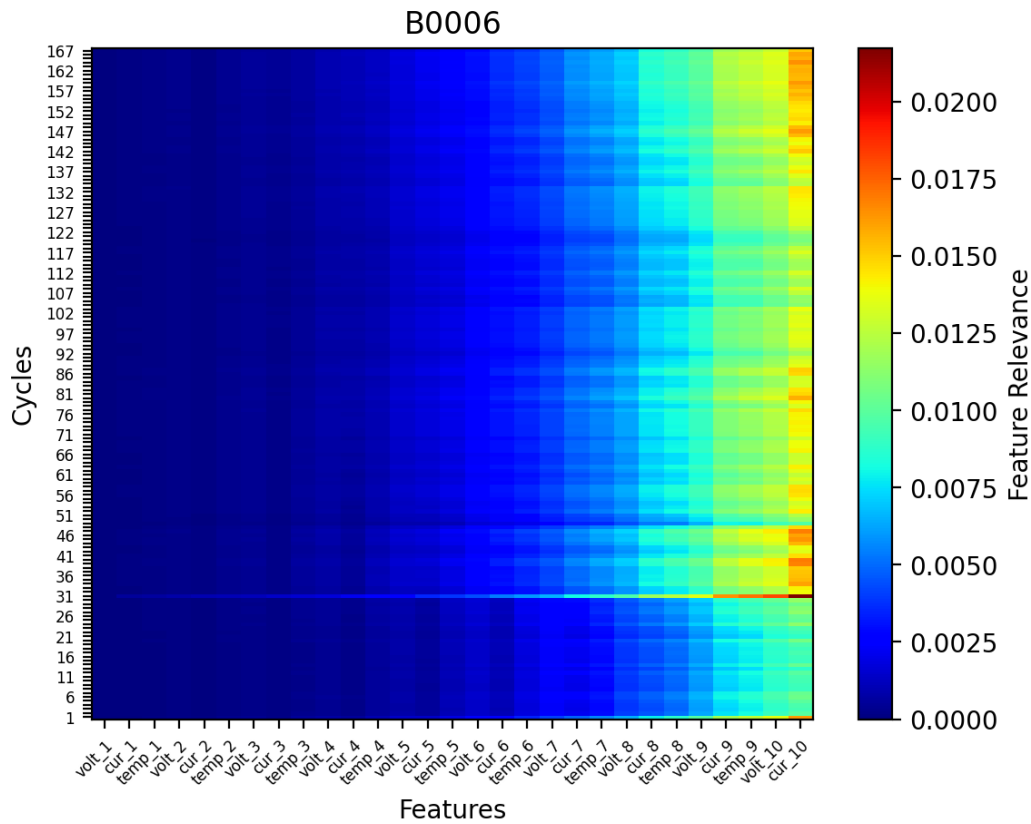
Figura 5.18: mappa di calore generata tramite la tecnica *Saliency Map* per la batteria B0006, escludendo la feature *cap*.



Dalla mappa di calore mostrata in figura 5.18 notiamo che la rilevanza della feature *temp_10* nel determinare la capacità di output è così determinante (rosso scuro) che è difficile comprendere a colpo d'occhio la rilevanza delle altre feature nel determinare la capacità di output.

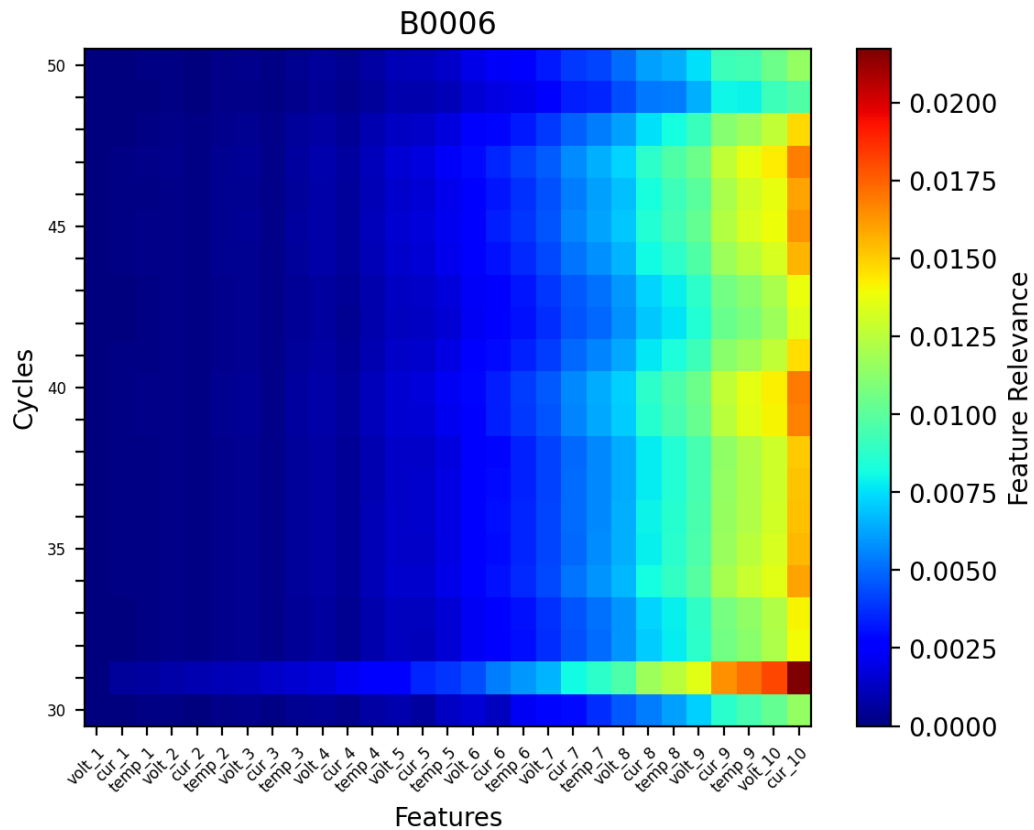
Ancora una volta, consideriamo la mappa di calore mostrata in figura 5.19, in cui è stata esclusa dalla valutazione la feature *temp_10*.

Figura 5.19: mappa di calore generata tramite la tecnica *Saliency Map* per la batteria B0006, escludendo le feature *cap* e *temp_10*.



Dalla mappa di calore mostrata in figura 5.19, possiamo notare che le feature *cur_9*, *temp_9*, *volt_10*, *cur_10* hanno anch'esse una certa rilevanza nel determinare la capacità di output. Notiamo poi che nei cicli da 30 a 50 circa, la rilevanza delle feature *temp_9*, *volt_10*, *cur_10* cresce. Lo stesso succede nei cicli da 70 a 90 circa, e nei cicli da 130 a 167 circa. Andiamo quindi ad osservare meglio questo fenomeno, concentrandoci sui cicli da 30 a 50 della batteria in analisi (figura 5.20).

Figura 5.20: mappa di calore generata tramite la tecnica *Saliency Map* per la batteria B0006, escludendo le feature *cap* e *temp_10*, per i cicli da 30 a 50.

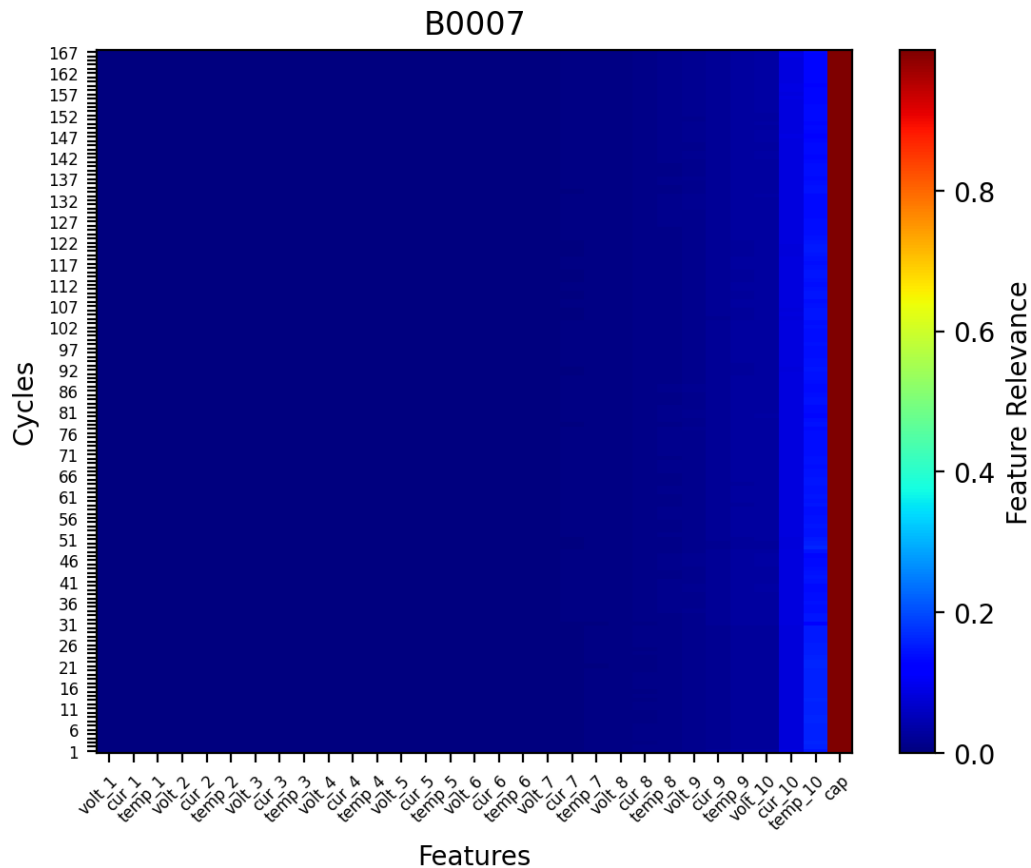


La mappa di calore mostrata in figura 5.20 mette in evidenza alcuni aspetti interessanti. Dal ciclo 34 al ciclo 42, e dal ciclo 44 al ciclo 48, la rilevanza delle feature *temp_9*, *volt_10*, *cur_10* nel determinare la capacità di output cresce. In particolare, nel ciclo 31, la rilevanza delle feature *volt_9*, *cur_9*, *temp_9*, *volt_10*, *cur_10* è ancora più marcata nel determinare l'output. Anche nel primo ciclo, la rilevanza delle feature sopra citate è decisamente più marcata. In generale, la feature *cur_10* delle 29 mostrate è quella che più influisce nella stima della capacità di output.

5.2.3.3 B0007

Per la batteria B0007, visualizzando per ogni ciclo la rilevanza di ogni feature nel determinare l'output, abbiamo ottenuto tramite la tecnica *Saliency Map* la mappa di calore mostrata in figura 5.21.

Figura 5.21: mappa di calore generata tramite la tecnica *Saliency Map* per la batteria B0007.



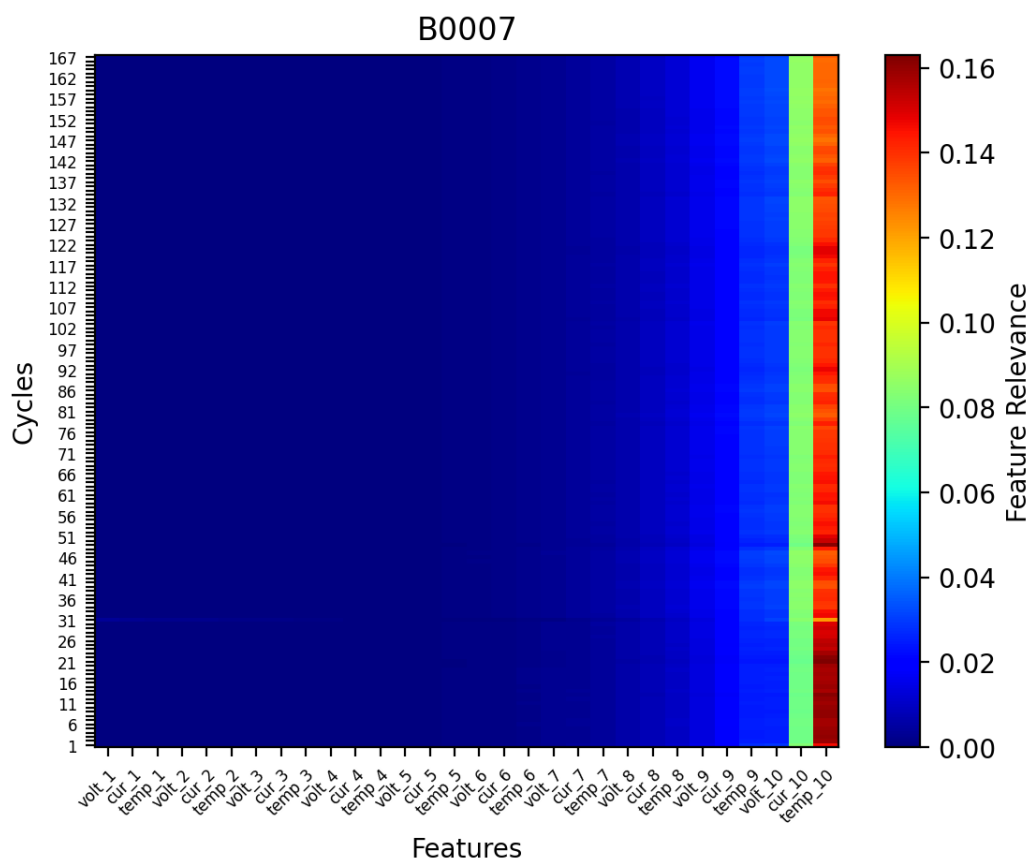
Dalla mappa di calore in figura 5.21, notiamo ancora una volta che la feature *cap* (capacità del ciclo di scarica precedente della batteria) è la feature che maggiormente contribuisce alla stima del valore di capacità di output del modello.

In secondo luogo, la feature *temp_10* (temperatura della batteria misurata al decimo e ultimo istante di tempo del ciclo di carica precedente) è di nuovo la seconda feature in termini di rilevanza nella stima della capacità della batteria. È possibile osservare che anche la feature *cur_10* risalta tra le feature che maggiormente contribuiscono a determinare la capacità di output.

Notiamo poi che ancora una volta la rilevanza della feature *cap* nel determinare la capacità di output è così determinante (rosso molto scuro) che è difficile comprendere a colpo d'occhio la rilevanza delle altre feature nel determinare la capacità di output.

Consideriamo dunque la mappa di calore mostrata in figura 5.22, in cui è stata esclusa dalla valutazione la feature *cap*.

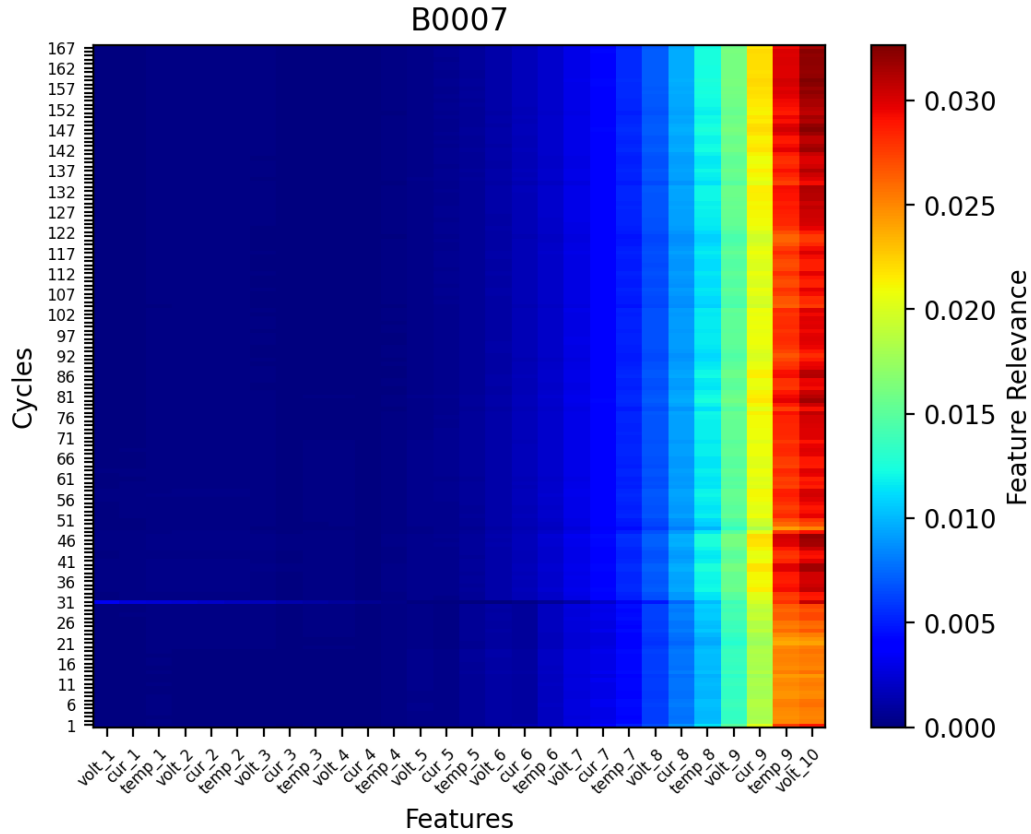
Figura 5.22: mappa di calore generata tramite la tecnica *Saliency Map* per la batteria B0007, escludendo la feature *cap*.



Dalla mappa di calore mostrata in figura 5.22 notiamo che la rilevanza delle feature *temp_10* e *cur_10* nel determinare la capacità di output è così determinante (rosso scuro e verde chiaro rispettivamente) che è difficile comprendere a colpo d'occhio la rilevanza delle altre feature nel determinare la capacità di output.

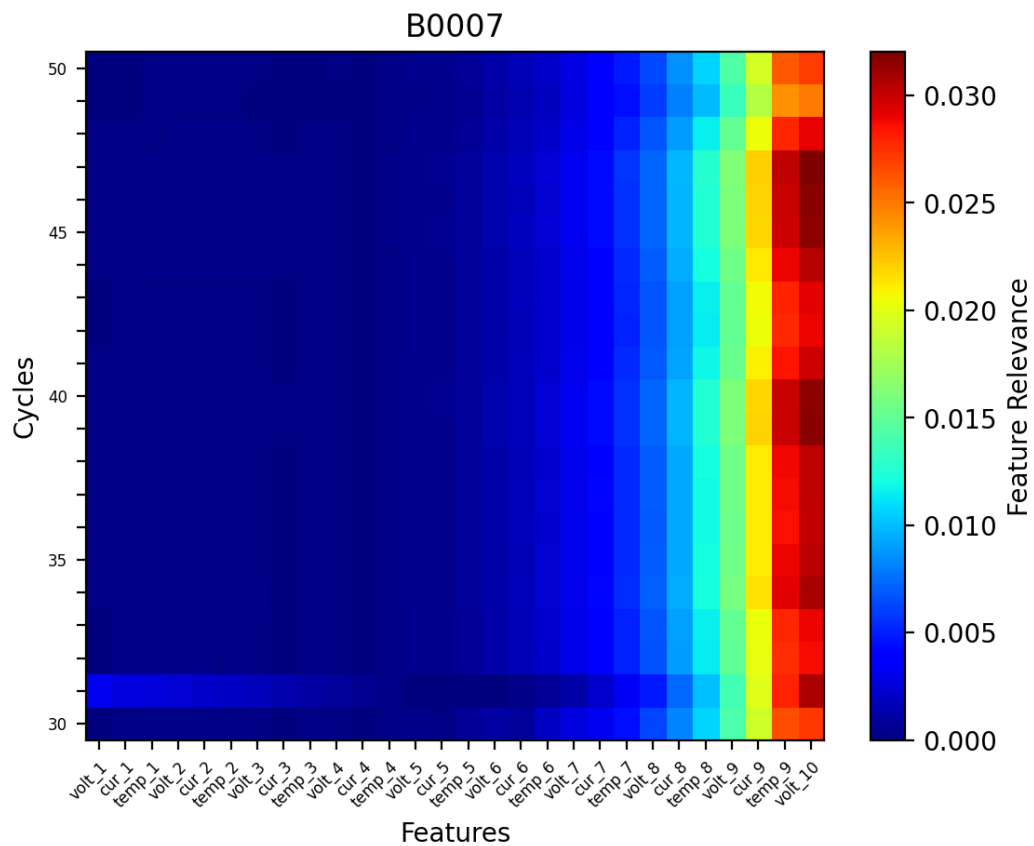
Ancora una volta, consideriamo la mappa di calore mostrata in figura 5.23, in cui sono state escluse dalla valutazione le feature *temp_10* e *cur_10*.

Figura 5.23: mappa di calore generata tramite la tecnica *Saliency Map* per la batteria B0007, escludendo le feature *cap*, *temp_10* e *cur_10*.



Dalla mappa di calore mostrata in figura 5.23, possiamo notare che le feature *temp_9*, *volt_10* hanno anch'esse una certa rilevanza nel determinare la capacità di output. Notiamo poi che nei cicli da 30 a 50 circa, la rilevanza delle feature *cur_9*, *temp_9*, *volt_10* cresce. Lo stesso succede nei cicli da 70 a 90 circa, e nei cicli da 130 a 167 circa. Andiamo quindi ad osservare meglio questo fenomeno, concentrandoci sui cicli da 30 a 50 della batteria in analisi (figura 5.24).

Figura 5.24: mappa di calore generata tramite la tecnica *Saliency Map* per la batteria B0007, escludendo le feature *cap*, *temp_10* e *cur_10* per i cicli da 30 a 50.

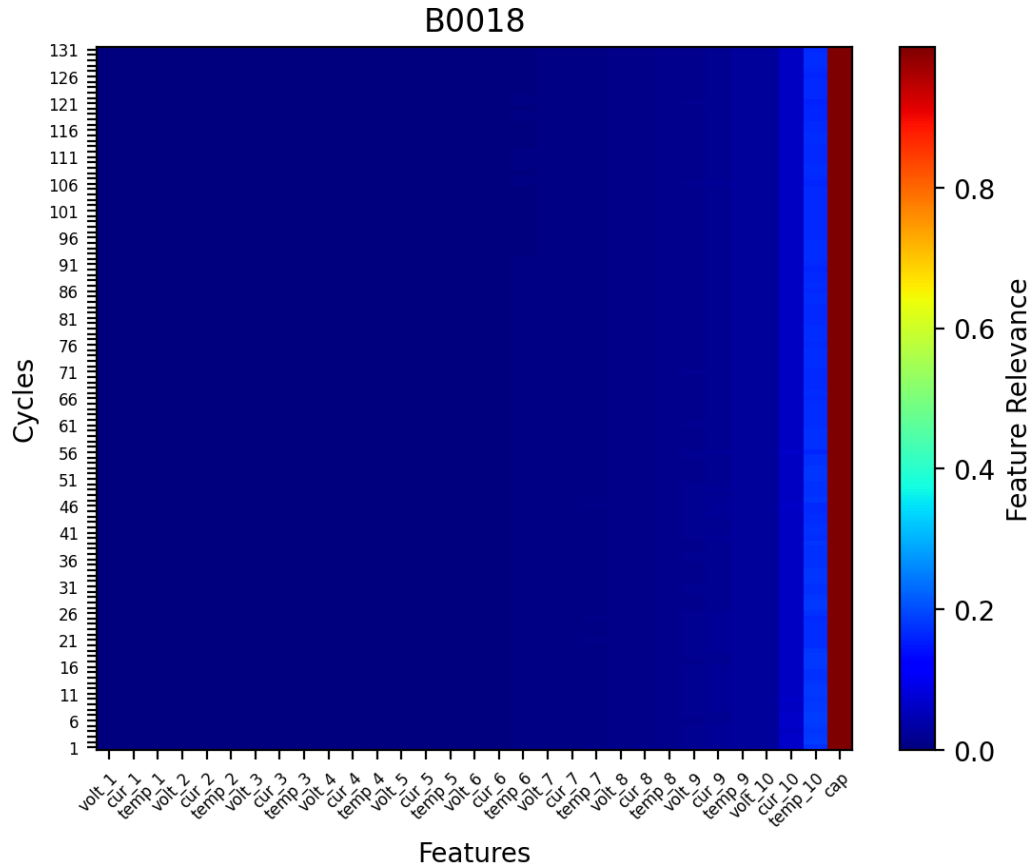


La mappa di calore mostrata in figura 5.24 mette in evidenza alcuni aspetti interessanti. Dal ciclo 34 al ciclo 41, e dal ciclo 44 al ciclo 48, la rilevanza delle feature *cur_9*, *temp_9*, *volt_10* nel determinare la capacità di output cresce. Notiamo poi che nel ciclo 31, la rilevanza delle feature da *volt_1* a *temp_4* è più marcata rispetto alla rilevanza delle feature da *volt_5* a *temp_8* nel determinare l'output.

5.2.3.4 B0018

Per la batteria B0018, visualizzando per ogni ciclo la rilevanza di ogni feature nel determinare l'output, abbiamo ottenuto tramite la tecnica *Saliency Map* la mappa di calore mostrata in figura 5.25.

Figura 5.25: mappa di calore generata tramite la tecnica *Saliency Map* per la batteria B0018.



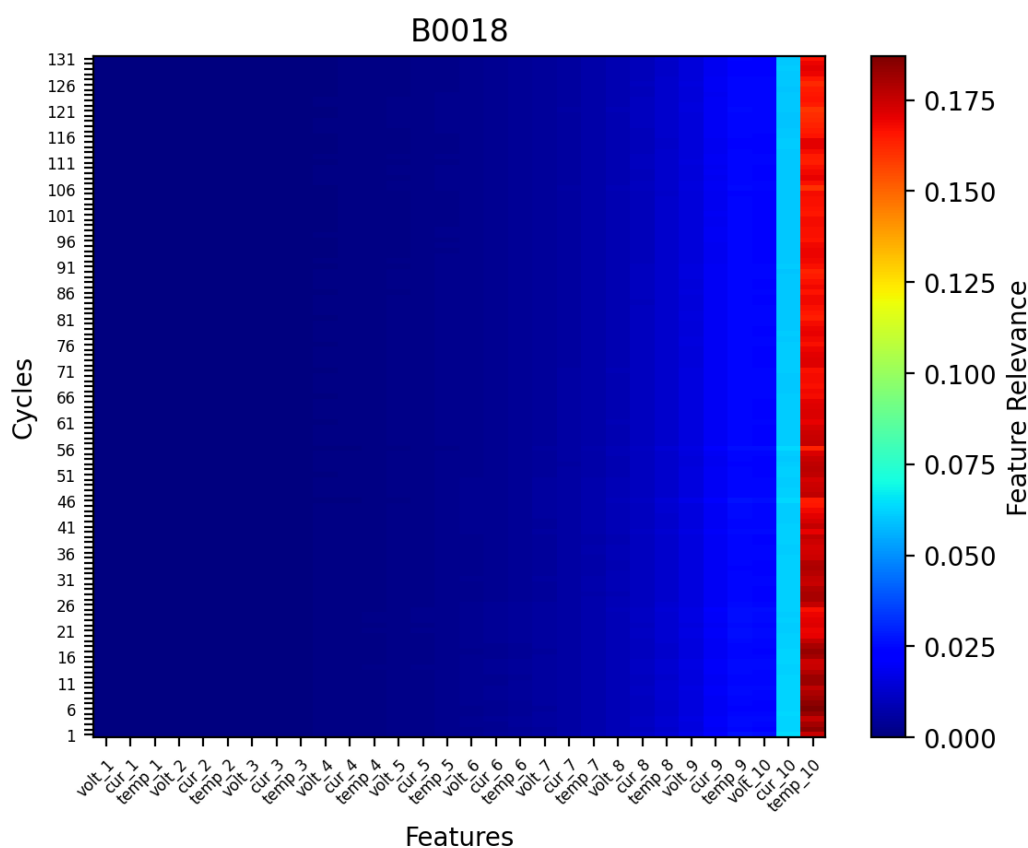
Osservando la mappa di calore mostrata in figura 5.25, notiamo innanzitutto che la feature *cap* (capacità del ciclo di scarica precedente della batteria) è la feature che maggiormente contribuisce alla stima del valore di capacità di output del modello.

In secondo luogo, la feature *temp_10* (temperatura della batteria misurata al decimo e ultimo istante di tempo del ciclo di carica precedente) è di nuovo la seconda feature in termini di rilevanza nella stima della capacità della batteria. È possibile osservare che anche la feature *cur_10* risalta tra le feature che maggiormente contribuiscono a determinare la capacità di output.

Notiamo poi che ancora una volta la rilevanza della feature *cap* nel determinare la capacità di output è così determinante (rosso molto scuro) che è difficile comprendere a colpo d'occhio la rilevanza delle altre feature nel determinare la capacità di output.

Consideriamo dunque la mappa di calore mostrata in figura 5.26, in cui è stata esclusa dalla valutazione la feature *cap*.

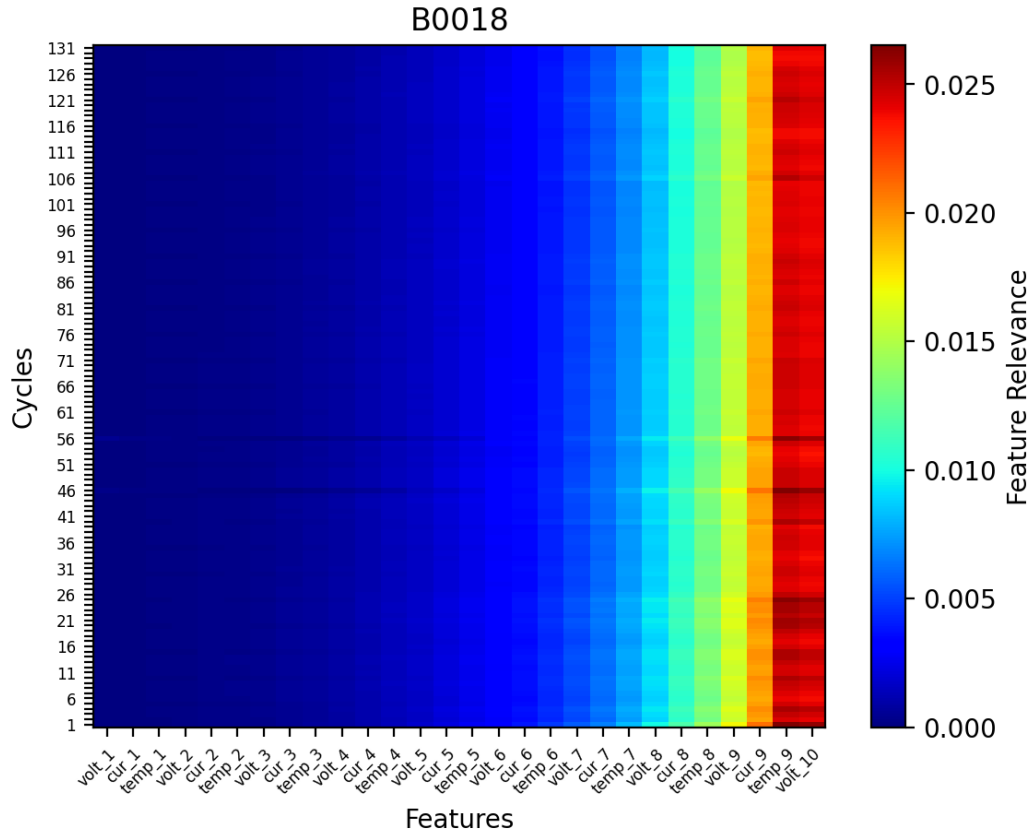
Figura 5.26: mappa di calore generata tramite la tecnica *Saliency Map* per la batteria B0018, escludendo la feature *cap*.



Dalla mappa di calore mostrata in figura 5.26, notiamo che la rilevanza delle feature *temp_10* e *cur_10* nel determinare la capacità di output è così determinante (rosso scuro e azzurro chiaro rispettivamente) che è difficile comprendere a colpo d'occhio la rilevanza delle altre feature nel determinare la capacità di output.

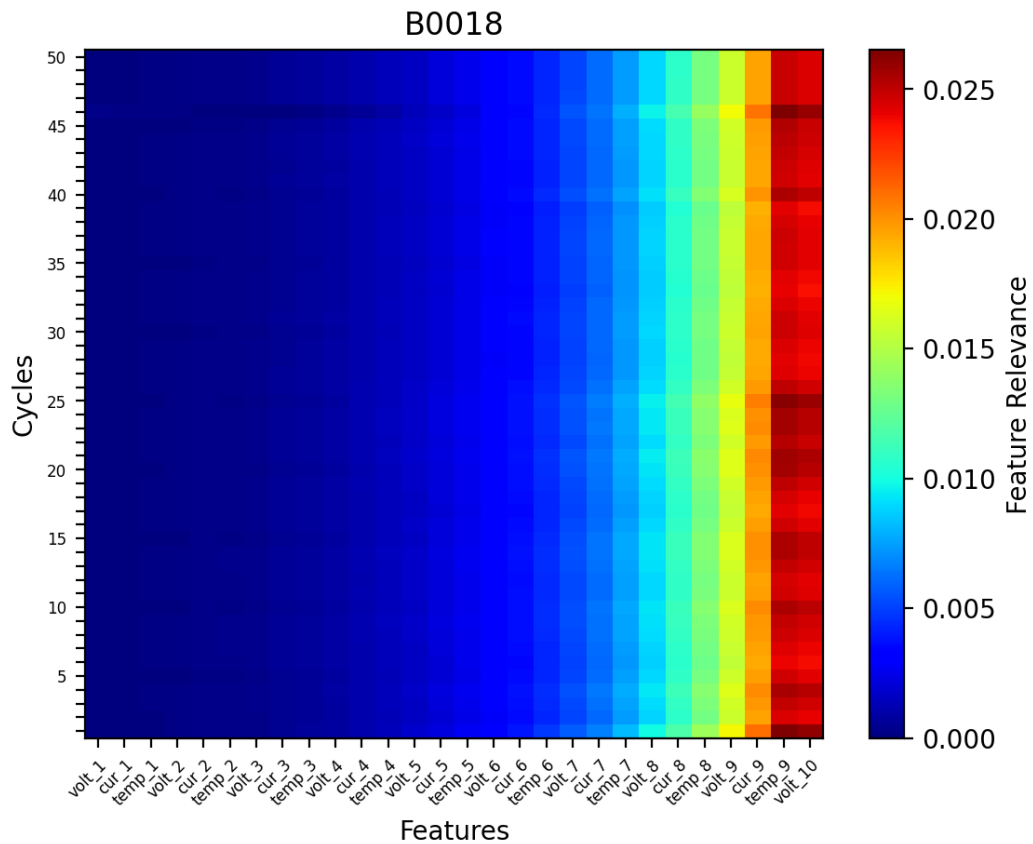
Ancora una volta, consideriamo la mappa di calore mostrata in figura 5.27, in cui sono state escluse dalla valutazione le feature *temp_10* e *cur_10*.

Figura 5.27: mappa di calore generata tramite la tecnica *Saliency Map* per la batteria B0018, escludendo le feature *cap*, *temp_10* e *cur_10*.



Dalla mappa di calore mostrata in figura 5.27, possiamo notare che le feature *temp_9*, *volt_10* hanno anch'esse una certa rilevanza nel determinare la capacità di output. Notiamo poi che nei cicli da 1 a 26 circa, e nei cicli 46 e 56 in particolar modo, la rilevanza delle feature *cur_9*, *temp_9*, *volt_10* cresce. Andiamo quindi ad osservare meglio questo fenomeno, concentrandoci sui primi 50 cicli della batteria in analisi (figura 5.28).

Figura 5.28: mappa di calore generata tramite la tecnica *Saliency Map* per la batteria B0018, escludendo le feature *cap*, *temp_10* e *cur_10* per i cicli da 30 a 50.



La mappa di calore mostrata in figura 5.28 mette in evidenza alcuni aspetti interessanti. Dal ciclo 1 al ciclo 5, dal ciclo 7 al ciclo 10, dal ciclo 13 al ciclo 16, dal ciclo 19 al ciclo 26, e nei cicli 40 e 46, la rilevanza delle feature *cur_9*, *temp_9*, *volt_10* nel determinare la capacità di output cresce.

5.2.4 LRP

Per quanto riguarda la tecnica di Explainability *LRP*, sono state testate diverse repository che la implementano, con l'obiettivo di applicare tale tecnica sul modello LSTM proposto ed implementato in questa tesi per la stima della capacità di una batteria al litio.

È stata testata in primo luogo la libreria *albermax/innvestigate*¹⁵ (Alber et al., 2019), che implementa diversi metodi di Explainability per reti neurali, fra

¹⁵ Libreria *albermax/innvestigate*: <https://github.com/albermax/innvestigate>. Visitato il 21 Luglio 2023.

cui *LRP*, con l'obiettivo di fornire un tool completo ed efficace che permetta di analizzare le previsioni delle reti neurali in maniera semplice ed intuitiva. In data 21 Luglio 2023 la repository *albermax/innvestigate* è nel seguente stato:

- Versione: 2.1.0
- Data ultimo rilascio: 5 Maggio 2023

Nella pagina web della libreria *albermax/innvestigate*, è stato segnalato un errore dovuto al mancato supporto di alcune architetture di reti neurali per l'applicazione delle tecniche di Explainability implementate nella libreria; tra le architetture non supportate, rientra anche l'architettura LSTM, presente nel modello proposto ed implementato in questo lavoro di tesi. Ad oggi, 21 Luglio 2023, questa segnalazione risulta aperta e non ancora risolta.

Provando ad applicare la tecnica di Explainability *LRP* tramite la libreria *albermax/innvestigate* sul modello LSTM proposto ed implementato in questa tesi, sono sorte effettivamente le problematiche relative al mancato supporto di architetture LSTM per l'applicazione della tecnica *LRP*.

Dato che nel modello proposto ed implementato in questa tesi utilizziamo un'architettura LSTM, non è stato possibile applicare la tecnica di Explainability *LRP* tramite la libreria *albermax/innvestigate* sul modello proposto ed implementato in questa tesi.

È stata poi testata la libreria python *marcoancona/DeepExplain*¹⁶, che implementa diverse tecniche di Explainability, fra cui *LRP*, fornendo un framework unificato per i metodi di Explainability basati sul gradiente e sulla perturbazione.

La libreria è già stata testata in questo lavoro di tesi per l'applicazione della tecnica di Explainability *DeepLIFT*.

In data 21 Luglio 2023, la repository *marcoancona/DeepExplain* è nel seguente stato:

- Non sono ancora presenti versioni ufficiali della libreria.

¹⁶ Libreria *marcoancona/DeepExplain*: <https://github.com/marcoancona/DeepExplain>. Visitato il 21 Luglio 2023.

- L'ultima modifica risale al 26 Agosto 2020.

Nella home page della libreria, l'autore riporta che attualmente è supportata solamente Tensorflow versione 1.x. Questa versione di Tensorflow è considerata obsoleta in ambito tecnico e di sviluppo.

L'autore riporta inoltre un riferimento a una versione sperimentale della libreria, in cui è stata introdotta la compatibilità con Tensorflow versione 2.x. Tuttavia, l'autore stesso deve ancora consolidare questa versione della libreria da lui sviluppata, spiegando che ci sono ancora bug legati a problemi di compatibilità con il moderno Tensorflow 2.x.

Provando ad applicare la tecnica di Explainability *LRP* tramite la libreria *marcoancona/DeepExplain* sul modello LSTM proposto ed implementato in questa tesi, sono sorte effettivamente le problematiche relative alla versione obsoleta di Tensorflow. Anche testando la libreria sperimentale compatibile con Tensorflow 2.x, sono sorte le problematiche preannunciate dall'autore stesso.

Dato che nel nostro setup sperimentale utilizziamo una versione di Tensorflow moderna (2.13.0), non è stato possibile applicare la tecnica di Explainability *LRP* tramite la libreria *marcoancona/DeepExplain* sul modello LSTM proposto ed implementato in questa tesi.

È stata infine testata la libreria *primozgodec/keras-explain*¹⁷, un repository che implementa le tecniche di Explainability più diffuse ed utilizzate nel campo delle reti neurali che prendono come input immagini.

In data 21 Luglio 2023 la repository *primozgodec/keras-explain* è nel seguente stato:

- Release: 0.0.1
- Data ultimo rilascio: 7 Gennaio 2020

¹⁷ Libreria *primozgodec/keras-explain*: <https://github.com/primozgodec/keras-explain>. Visitato il 21 Luglio 2023.

Nella home page della libreria, l'autore riporta che attualmente è supportato solamente Keras con versione inferiore alla 2.2.0. Queste versioni di Keras sono considerate obsolete in ambito tecnico e di sviluppo.

Provando ad applicare la tecnica di Explainability *LRP* tramite la libreria *primozgodec/keras-explain* sul modello LSTM proposto ed implementato in questa tesi, sono sorte effettivamente le problematiche relative alla versione obsoleta di Keras.

Dato che nel nostro setup sperimentale utilizziamo una versione di Keras moderna (2.13.1), non è stato possibile applicare la tecnica di Explainability *LRP* tramite la libreria *primozgodec/keras-explain* sul modello LSTM proposto ed implementato in questa tesi.

In conclusione, a causa del mancato supporto e aggiornamento delle librerie disponibili sul web che implementano tecniche di Explainability *LRP*, non è stato possibile effettuare esperimenti di Explainability con la tecnica *LRP* sul modello LSTM proposto ed implementato in questo lavoro di tesi utilizzando come setup sperimentale un ambiente recente, con Python versione 3.11.4, Keras versione 2.13.1 e Tensorflow versione 2.13.0.

CAPITOLO SESTO

6. Sviluppo applicazione web

Per permettere a chiunque di esplorare il dataset *Li-ion Battery Aging Dataset* utilizzato per l'addestramento della rete neurale LSTM proposta, di interagire con il modello LSTM proposto verificandone l'accuratezza e di sperimentare le tecniche di Explainability testate sulla rete neurale proposta verificandone in autonomia l'affidabilità, è stata sviluppata e pubblicata un'applicazione web Streamlit¹⁸ contenente tutto il lavoro di ricerca e sviluppo svolto in questa tesi.

L'applicazione web è stata sviluppata quindi in ottica di fornire uno strumento, utilizzabile come base di partenza per lavori successivi di esplorazione e approfondimento su tecniche di Explainability applicate al dominio della stima della capacità delle batterie agli ioni di litio su reti LSTM.

In questo capitolo verrà presentata e descritta l'applicazione web sviluppata.

6.1 Streamlit

Streamlit¹⁹ è una libreria Python open-source che permette di creare applicazioni web personalizzate, consentendo poi di distribuire e pubblicare tali applicazioni direttamente all'interno dell'infrastruttura *Streamlit Cloud Community*²⁰. Questa libreria è utilizzata prevalentemente per lo sviluppo e la distribuzione di applicazioni basate sul machine learning e in ambito scientifico in generale.

È stata adottata la libreria Streamlit per lo sviluppo dell'applicazione web, utilizzando poi l'infrastruttura *Streamlit Cloud Community* per distribuire e pubblicare l'applicazione stessa.

¹⁸ Applicazione web Streamlit: <https://battery-lstm-explainability.streamlit.app/>.

¹⁹ Streamlit: <https://streamlit.io/>. Visitato il 21 Luglio 2023.

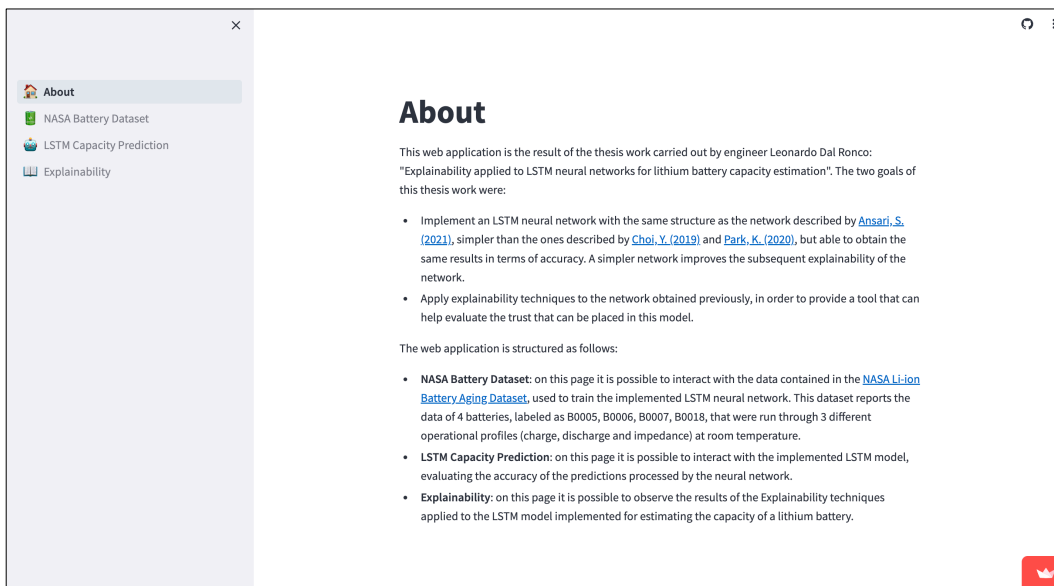
²⁰ Streamlit Cloud Community: <https://streamlit.io/cloud>. Visitato il 21 Luglio 2023.

6.2 Applicazione web

L'applicazione web sviluppata²¹ è stata strutturata inserendo 4 pagine, accessibili da un apposito menu laterale:

- *About*: pagina introduttiva, in cui è descritto il lavoro svolto in questa tesi.
- *Nasa Battery Dataset*: in questa pagina, è possibile esplorare il dataset *Li-ion Battery Aging Dataset* messo a disposizione dal *NASA Prognostics Center of Excellence Data Set Repository* ed utilizzato per l'addestramento della rete LSTM proposta.
- *LSTM Capacity Prediction*: in questa pagina è possibile interagire con il modello LSTM proposto verificandone l'accuratezza.
- *Explainability*: in questa pagina è possibile sperimentare le tecniche di Explainability testate sulla rete neurale LSTM proposta verificandone in autonomia l'affidabilità.

Figura 6.1: pagina *About* dell'applicazione web sviluppata.



Nei paragrafi successivi verranno descritte nel dettaglio le pagine *NASA Battery Dataset*, *LSTM Capacity Prediction* ed *Explainability*.

²¹ Applicazione web: <https://battery-lstm-explainability.streamlit.app/>.

6.2.1 Nasa Battery Dataset

Nella pagina *Nasa Battery Dataset* è contenuto il lavoro di ricerca e analisi svolto nel Paragrafo 3.1 *Nasa Battery Dataset* di questa tesi.

Tramite questa pagina è possibile esplorare il dataset *Li-ion Battery Aging Dataset* messo a disposizione dal *NASA Prognostics Center of Excellence Data Set Repository*²² ed utilizzato per l'addestramento della rete LSTM proposta.

In particolare è possibile:

- Scegliere una batteria di riferimento.
- Scegliere se visualizzare l'andamento dei valori di tensione, intensità di corrente e temperatura misurati durante i cicli di carica, oppure i valori di capacità misurati durante i cicli di scarica della batteria scelta.
- Se si sceglie di visualizzare i valori di tensione, di intensità di corrente oppure di temperatura misurati durante i cicli di carica, sarà possibile filtrare il grafico risultante in base al tempo (istante iniziale e istante finale), e sarà possibile scegliere quali cicli di carica visualizzare.
- Se si sceglie invece di visualizzare i valori di capacità misurati durante i cicli di scarica, sarà possibile filtrare il grafico risultante in base ai cicli (ciclo iniziale e ciclo finale).

Una volta impostati tali parametri, verrà generato e mostrato nella pagina il grafico risultante configurato in base ai valori sopra selezionati.

Nelle figure 6.2 e 6.3 sono mostrati alcuni possibili esempi di utilizzo della pagina *NASA Battery Dataset*.

²² NASA Li-ion Battery Aging Dataset: <https://www.nasa.gov/content/prognostics-center-of-excellence-data-set-repository>. Visitato il 21 Luglio 2023.

Figura 6.2: pagina *NASA Battery Dataset* in è mostrato, per la batteria B0005, l'andamento della tensione nel tempo per i cicli di carica 2, 21, 41, 61, 81, 102, 122 e 142, dal minuto 0 fino al minuto 100.

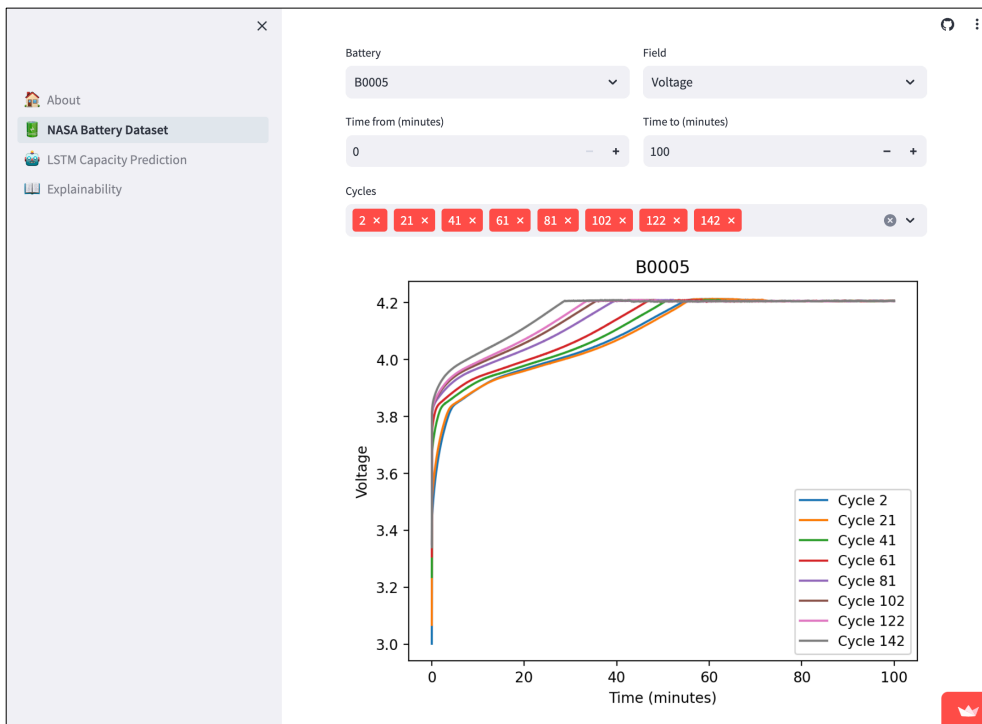
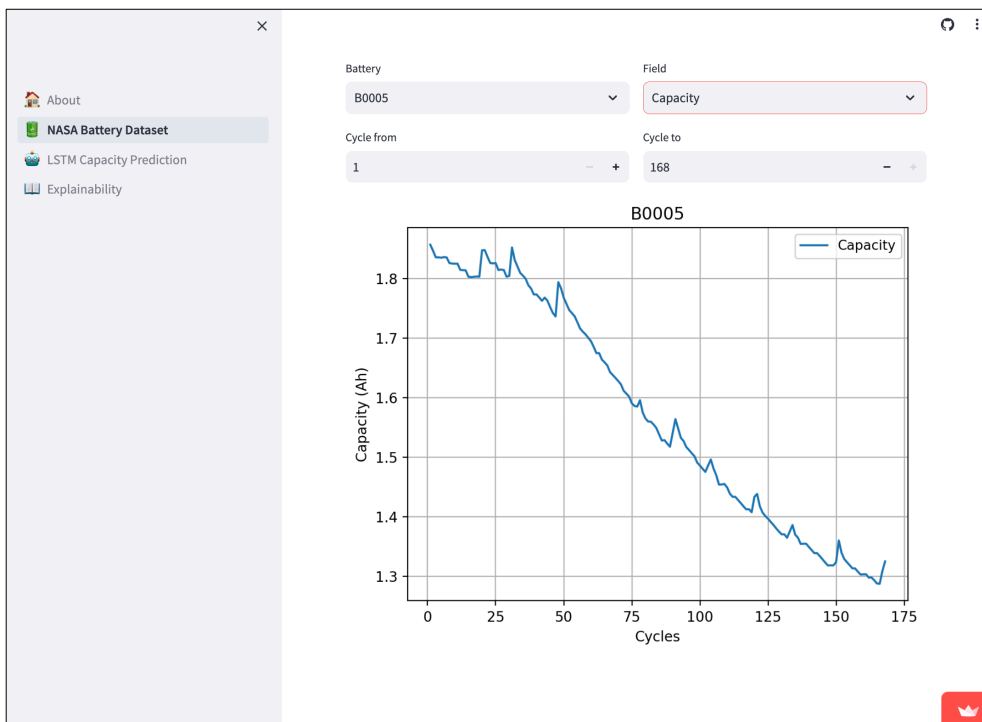


Figura 6.3: pagina *NASA Battery Dataset* in è mostrato, per la batteria B0005, l'andamento della capacità nel tempo, dal ciclo 1 fino al ciclo 168.



6.2.2 LSTM Capacity Prediction

Nella pagina *LSTM Capacity Prediction* sono contenuti i risultati del lavoro di ricerca e sviluppo svolto nel Capitolo 4. *Progettazione e implementazione del sistema* di questa tesi.

Tramite questa pagina è possibile interagire con il modello LSTM proposto ed implementato in questa tesi, verificandone l'accuratezza delle previsioni effettuate.

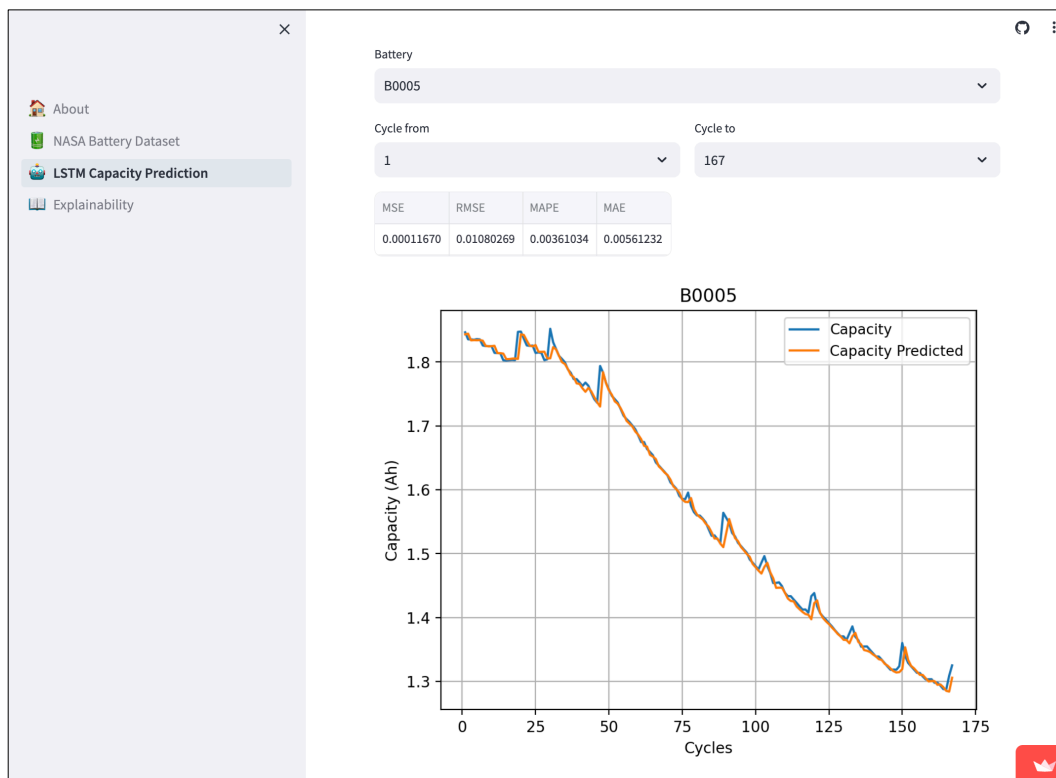
In particolare è possibile:

- Scegliere una batteria di riferimento.
- Scegliere i cicli per i quali si vuole effettuare una previsione (ciclo iniziale e ciclo finale).

Una volta impostati tali parametri, verrà visualizzata una tabella in cui saranno riportate le metriche di accuratezza (MSE, RMSE, MAPE, MAE) del modello LSTM che utilizza la batteria sopra selezionata come batteria di test. Verrà poi generato e visualizzato nella pagina un grafico in cui vengono messi a confronto i valori di capacità reali con i valori di capacità stimati dal modello LSTM, sui cicli per i quali si vuole effettuare una previsione.

In figura 6.4 è mostrato un possibile esempio di utilizzo della pagina *LSTM Capacity Prediction*.

Figura 6.4: pagina *LSTM Capacity Prediction* in sono mostrate le metriche di accuratezza per il modello che utilizza la batteria B0005 come batteria di test. Viene inoltre mostrato il grafico in cui vengono messi a confronto i valori di capacità reali con i valori di capacità stimati per la batteria B0005, dal ciclo 1 al ciclo 167.



6.2.3 Explainability

Nella pagina *Explainability* sono contenuti i risultati delle tecniche di Explainability testate nel Capitolo 5. *Esperimenti* di questa tesi.

Tramite questa pagina è possibile sperimentare le tecniche di Explainability testate sulla rete neurale LSTM proposta e implementata in questo lavoro di tesi verificandone in autonomia l'affidabilità.

In particolare è possibile:

- Scegliere una batteria di riferimento.
- Scegliere la tecnica di Explainability che si desidera testare, scegliendo tra *SHAP* e *Saliency Map*.
- Scegliere quali feature si vogliono escludere dal grafico di spiegazione che verrà generato.
- Scegliere i cicli per i quali si vuole visualizzare il grafico di spiegazione (ciclo iniziale e ciclo finale).
- Per la tecnica *SHAP* è poi possibile:
 - Scegliere il numero di features da mostrare nel grafico di spiegazione
 - Ordinare le feature mostrate in base al tempo, dalle feature più recenti (*volt_10*, *cur_10*, *temp_10*) alle feature meno recenti (*volt_1*, *cur_1*, *temp_1*); altrimenti, le feature saranno mostrate in ordine di rilevanza, dalla più rilevante alla meno rilevante.

Una volta impostati tali parametri, verrà generato e visualizzato nella pagina un grafico di spiegazione configurato in base ai valori sopra selezionati.

Nelle figure 6.5 e 6.6 sono mostrati alcuni possibili esempi di utilizzo della pagina *Explainability*.

Figura 6.5: pagina *Explainability* in cui è mostrato il grafico di spiegazione generato tramite la tecnica *SHAP* per il modello che utilizza la batteria B0005 come batteria di test, escludendo le feature *cap* e *temp_10*, per i cicli da 1 a 167; vengono mostrate le prime 12 feature in ordine di rilevanza nella stima della capacità da parte del modello.

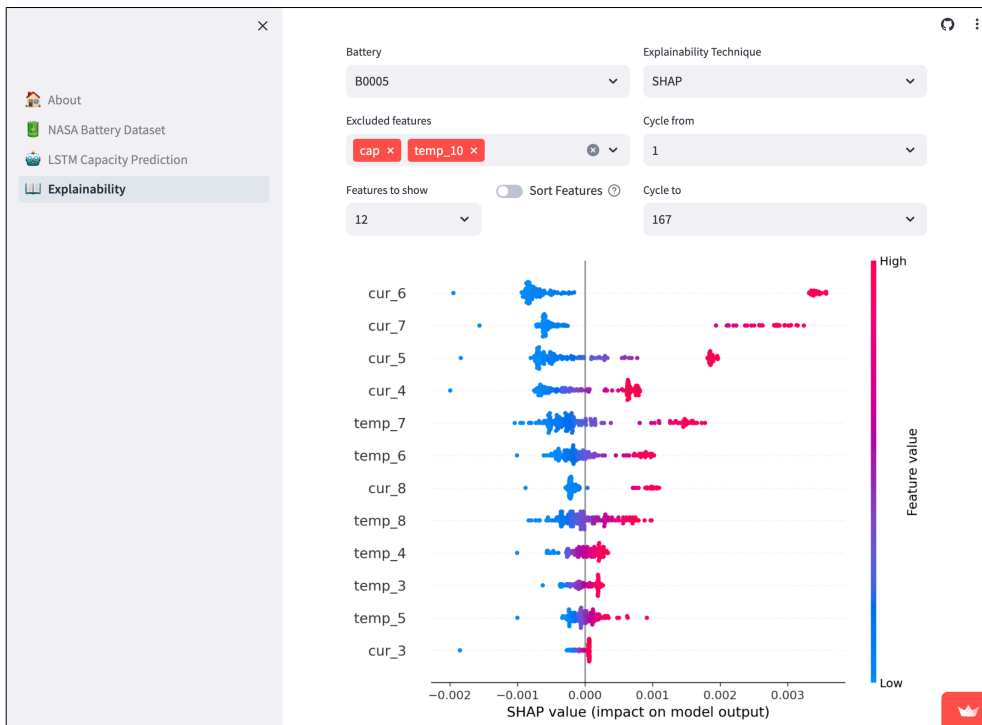
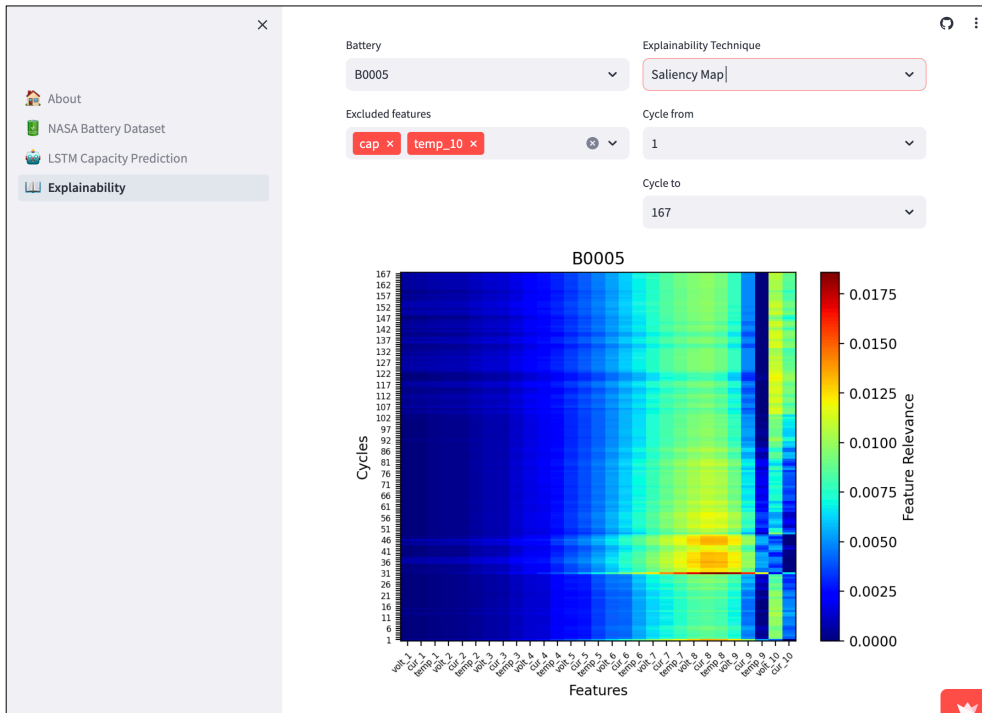


Figura 6.6: pagina *Explainability* in cui è mostrato il grafico di spiegazione generato tramite la tecnica *Saliency Map* per il modello che utilizza la batteria B0005 come batteria di test, escludendo le feature *cap* e *temp_10*, per i cicli da 1 a 167;



CAPITOLO SETTIMO

7. Valutazione dei risultati, limitazioni e sviluppi futuri

In questo capitolo verranno valutati i risultati ottenuti in questo lavoro di tesi; in particolare verranno valutate le risposte trovate alle domande di ricerca iniziali.

Verranno poi analizzate e discusse le limitazioni del lavoro effettuato e gli sviluppi futuri.

7.1 Valutazione dei risultati

In risposta alla domanda di ricerca RQ1, abbiamo ottenuto una rete neurale LSTM con la stessa struttura della rete descritta da Ansari (Ansari et al., 2021a), più semplice delle strutture descritte da Park (Park et al., 2020) e Choi (Choi et al., 2019) in termini di numero di parametri su cui effettuare l'addestramento, ma in grado di ottenere gli stessi risultati in termini di accuratezza. Raggiungendo questo obiettivo, abbiamo guadagnato una rete più snella, più veloce nell'elaborazione dell'output, più adatta e versatile per essere distribuita all'interno di BMS con limitate risorse di calcolo e più facile da spiegare tramite le tecniche di Explainability.

In particolare il modello LSTM proposto e implementato in questo lavoro di tesi in termini di parametri su cui effettuare l'addestramento possiede:

- 82,3% di parametri in meno rispetto al modello proposto da Park (da 2772 a 491).
- 97,2% di parametri in meno rispetto al modello proposto da Choi (da 17472 a 491).

Come possiamo vedere, il numero di parametri su cui effettuare l'addestramento è diminuito drasticamente nel modello LSTM proposto ed implementato in questa tesi.

Confrontando le metriche ottenute dalla rete LSTM proposta ed implementata in questo lavoro di tesi con quelle ottenute da Park (Park et al., 2020), risulta che:

- MSE è diminuito del 46,6% (da 0,00052274 a 0,00027936)
- RMSE è diminuito del 21,7% (da 0,019825 a 0,01552518)

- MAPE è diminuito del 42,1% (da 0,01025 a 0,00593439)
- MAE è diminuito del 35,7% (da 0,01445 a 0,00929386)

Analogamente, confrontando le metriche ottenute dalla rete LSTM proposta ed implementata in questo lavoro di tesi con quelle ottenute da Choi (Choi et al., 2019), risulta che:

- MSE è diminuito del 53,8% (da 0,00060516 a 0,00027936)
- RMSE è diminuito del 36,9% (da 0,0246 a 0,01552518)
- MAPE è diminuito del 42,5% (da 0,010320 a 0,00593439)
- MAE è diminuito del 41,5% (da 0,0159 a 0,00929386)

Abbiamo dunque ottenuto un modello LSTM non solo comparabile in termini di accuratezza con quelli proposti da Park (Park et al., 2020) e Choi (Choi et al., 2019), ma addirittura migliore.

In risposta poi alla domanda RQ2, sono state applicate le tecniche di Explainability *SHAP* e *Saliency Map* sulla rete LSTM proposta ed implementata in questo lavoro di tesi in un ambiente di sviluppo moderno.

A causa del mancato supporto e aggiornamento delle librerie disponibili sul web che implementano tecniche di Explainability *DeepLIFT* e *LRP*, non è stato possibile effettuare esperimenti di Explainability con tali tecniche sul modello LSTM proposto ed implementato in questo lavoro di tesi utilizzando come setup sperimentale un ambiente recente, con Python versione 3.11.4, Keras versione 2.13.1 e Tensorflow versione 2.13.0.

Analizzando i risultati ottenuti dagli esperimenti svolti tramite le tecniche *SHAP* e *Saliency Map*, possiamo notare che entrambe le tecniche hanno messo in luce una rilevanza predominante della feature *cap* (capacità del ciclo precedente della batteria) nell'andare a stimare la capacità attuale.

Ciò conferma l'importanza dello storico dei valori di capacità di una batteria nell'andare a stimare la capacità attuale della batteria (Li et al., 2019).

In particolare, tramite la tecnica *SHAP*, notiamo che più grande è la capacità del ciclo precedente, maggiore sarà la capacità stimata dal modello. Questa spiegazione fornita dalla tecnica *SHAP* aumenta di molto l'affidabilità del modello stesso, in quanto se una batteria dimostrerà una capacità elevata nel ciclo precedente, il modello premierà in maniera positiva questa misurazione,

andando a stimare una capacità attuale similmente alta. Sempre tramite la tecnica *SHAP*, notiamo che il modello si comporterà analogamente nel caso in cui la capacità del ciclo precedente risulti essere inferiore, andando a stimare una capacità attuale più piccola. Allo stesso modo, anche questa spiegazione fornita dalla tecnica *SHAP* aumenta di molto l'affidabilità del modello, mostrandoci che la rete neurale si comporta in maniera coerente, dando la giusta importanza ai valori di capacità dei cicli precedenti.

Tuttavia, modelli basati solamente sullo storico dei valori di capacità di una batteria non sono in grado di catturare il fenomeno della rigenerazione della capacità, che consiste in un aumento irregolare della capacità durante i cicli di carica-scarica (Li et al., 2019).

Tale fenomeno è stato dunque gestito dal modello LSTM proposto ed implementato in questa tesi grazie all'introduzione delle misurazioni di tensione, corrente e temperatura durante il ciclo di carica della batteria. Queste feature aggiuntive (*volt_i*, *cur_i*, *temp_i*), unite alla feature *cap*, hanno permesso infatti di catturare il fenomeno della rigenerazione della capacità.

In particolare, tramite la tecnica *Saliency Map*, possiamo notare che tali feature contribuiscono alla messa a punto del valore di capacità elaborato dalla rete neurale. Nel dettaglio, i contributi di queste feature secondarie variano da ciclo a ciclo e le mappe di calore generate evidenziano porzioni di cicli adiacenti che sono influenzati dalle stesse feature.

7.2 Limitazioni del lavoro svolto

Siamo consapevoli del fatto che attualmente è stata presa a riferimento l'architettura proposta da Ansari (Ansari et al., 2021a), senza esplorare eventuali varianti dell'architettura stessa dal punto di vista degli iperparametri. Non è stata condotta quindi alcuna attività di *hyperparameter tuning* della rete LSTM proposta, dato che questa non rientrava nelle domande di ricerca poste all'inizio della tesi. Sarebbe comunque interessante svolgere tale attività, in quanto siamo consapevoli che una messa a punto degli iperparametri potrebbe migliorare ulteriormente l'accuratezza delle previsioni effettuate dal modello stesso.

Siamo poi consapevoli del fatto che il Dataset *Li-ion Battery Aging Dataset* presenta del rumore al suo interno, e che dunque andrebbe condotta un'analisi approfondita di tale dataset per rimuovere il rumore in eccesso dai dati al suo interno. La rimozione del rumore potrebbe anch'essa migliorare ulteriormente l'accuratezza delle previsioni effettuate dal modello LSTM proposto.

7.3 Sviluppi futuri

Esistono diverse strade tramite le quali proseguire il lavoro svolto in questa tesi, partendo dai risultati raggiunti:

1. In ottica di potenziare ulteriormente l'accuratezza delle previsioni prodotte dal modello LSTM proposto in questa tesi, sarebbe interessante aggiungere altri campi di input al modello, che potrebbero essere particolarmente determinanti nel catturare il fenomeno della rigenerazione della capacità, e quindi potrebbero contribuire a fornire una stima più accurata della capacità attuale di una batteria agli ioni di litio. Alcuni campi rilevanti per la cattura del fenomeno di rigenerazione di capacità potrebbero essere ad esempio *“tempo in cui è stata raggiunta la tensione massima durante il ciclo di carica”*, *“rapidità di diminuzione dell'intensità di corrente durante un ciclo di carica”*, *“tempo in cui è stata raggiunta la tensione massima durante il ciclo di carica”* e *“valore della massima temperatura raggiunta”*. Questo perché, come già visto nel Paragrafo 3.1 *NASA Battery Dataset*, durante un ciclo di carica di una batteria più usurata (sulla quale sono stati effettuati più cicli di carica – scarica) il valore della tensione di soglia 4.2 V viene raggiunto molto prima, il valore dell'intensità di corrente diminuisce più rapidamente, e il valore di temperatura massima viene raggiunto più rapidamente rispetto a una batteria più giovane.
2. In ottica di sperimentare altre tecniche di Explainability sulla rete LSTM proposta, sarebbe utile condurre un lavoro di aggiornamento e adattamento delle librerie esistenti disponibili sul web che implementano tecniche di Explainability *DeepLIFT* e *LRP*, così da

riuscire a renderle operative in un ambiente di sviluppo moderno su un modello LSTM. Una volta compiuto questo primo passo, sarà possibile testare tali metodologie sul dominio della stima della capacità di batterie agli ioni di litio, analizzare le spiegazioni ottenute come output da tali tecniche, e confrontarle con le tecniche *SHAP* e *Saliency Map* già testate in questo lavoro di tesi.

3. Come ultimo passo, in ottica di distribuire il modello LSTM proposto in un ambiente di produzione, sarebbe interessante condurre un lavoro di sperimentazione per installare tale rete neurale all'interno di un BMS, in modo da analizzare il comportamento di tale rete quando avrà a che fare con dati di input inediti. A questa attività farà poi seguito un lavoro di monitoraggio dell'ambiente di produzione stesso, analizzando l'efficacia del sistema di stima dello stato di salute delle batterie, e verificando se effettivamente vengono riscontrate diminuzioni nei costi di manutenzione e di sostituzione delle batterie effettivamente logore.

CAPITOLO OTTAVO

8. Conclusioni

In questo lavoro di tesi abbiamo ottenuto una rete neurale LSTM in grado di stimare la capacità di una batteria agli ioni di litio con la stessa struttura della rete neurale descritta da Ansari (Ansari et al., 2021a), più semplice delle strutture descritte da Park (Park et al., 2020) e Choi (Choi et al., 2019) in termini di numero di parametri su cui effettuare l'addestramento, ma in grado di ottenere gli stessi risultati in termini di accuratezza. Raggiungendo questo obiettivo, abbiamo guadagnato una rete più snella, più veloce nell'elaborazione dell'output, più adatta e versatile per essere distribuita all'interno di BMS con limitate risorse di calcolo e più facile da spiegare tramite le tecniche di Explainability.

In particolare il modello LSTM proposto, in termini di parametri su cui effettuare l'addestramento, possiede l'82,3% di parametri in meno rispetto al modello proposto da Park (Park et al., 2020), e il 97,2% di parametri in meno rispetto al modello proposto da Choi (Choi et al., 2019).

Dal punto di vista delle metriche di accuratezza, nella rete LSTM proposta l'MSE è diminuito del 46,6%, l'RMSE è diminuito del 21,7%, il MAPE è diminuito del 42,1% e il MAE è diminuito del 35,7% rispetto ai valori ottenuti da Park (Park et al., 2020); in maniera analoga, l'MSE è diminuito del 53,8%, l'RMSE è diminuito del 36,9%, il MAPE è diminuito del 42,5% e il MAE è diminuito del 41,5% rispetto ai valori ottenuti da Choi (Choi et al., 2019).

Abbiamo quindi risposto in maniera affermativa alla RQ1, ottenendo un modello LSTM non solo comparabile in termini di accuratezza con quelli proposti da Park (Park et al., 2020) e Choi (Choi et al., 2019), ma addirittura migliore.

In risposta poi alla domanda RQ2, tramite gli esperimenti svolti, sono state applicate le tecniche di Explainability *SHAP* e *Saliency Map* sulla rete LSTM proposta ed implementata in questo lavoro di tesi in un ambiente di sviluppo moderno. Le tecniche di Explainability testate hanno confermato il fatto che il solo storico dei valori di capacità di una batteria non è in grado di catturare

il fenomeno della rigenerazione della capacità (aumento irregolare della capacità durante i cicli di carica-scarica), mettendo in luce la rilevanza delle varie feature nell'elaborazione dell'output generato dal modello LSTM proposto. A causa del mancato supporto e aggiornamento delle librerie disponibili sul web che implementano tecniche di Explainability *DeepLIFT* e *LRP*, non è stato possibile effettuare esperimenti di Explainability con tali tecniche sul modello LSTM proposto ed implementato in questo lavoro di tesi utilizzando come setup sperimentale un ambiente recente, con Python versione 3.11.4, Keras versione 2.13.1 e Tensorflow versione 2.13.0.

Infine, per permettere a chiunque di esplorare il dataset *Li-ion Battery Aging Dataset* utilizzato per l'addestramento della rete neurale LSTM proposta, di interagire con il modello LSTM proposto verificandone l'accuratezza e di sperimentare le tecniche di Explainability testate sulla rete neurale proposta verificandone in autonomia l'affidabilità, è stata sviluppata e pubblicata un'applicazione web Streamlit²³ contenente tutto il lavoro di ricerca e sviluppo svolto in questa tesi. L'applicazione web è stata sviluppata quindi in ottica di fornire uno strumento, utilizzabile come base di partenza per lavori successivi di esplorazione e approfondimento su tecniche di Explainability applicate al dominio della stima della capacità delle batterie agli ioni di litio su reti LSTM.

Il lavoro fino a qui svolto potrà proseguire in diverse direzioni, ad esempio conducendo un'ulteriore analisi sull'aggiunta di altri campi di input al modello che potrebbero essere particolarmente determinanti nel catturare il fenomeno della rigenerazione della capacità, e che quindi potrebbero contribuire a fornire una stima più accurata della capacità attuale di una batteria agli ioni di litio. Sarebbe utile condurre poi un lavoro di aggiornamento e adattamento delle librerie esistenti disponibili sul web che implementano tecniche di Explainability *DeepLIFT* e *LRP*, così da riuscire a renderle operative in un ambiente di sviluppo moderno su un modello LSTM. Infine, in ottica di distribuire il modello LSTM proposto in un ambiente di produzione, sarebbe interessante condurre un lavoro di sperimentazione per

²³ Applicazione web Streamlit: <https://battery-lstm-explainability.streamlit.app/>.

installare tale rete neurale all'interno di un BMS, in modo da analizzare il comportamento di tale rete quando avrà a che fare con dati di input inediti.

Nomenclatura

ANN = Artificial Neural Network

ARIMA = Auto-Regressive

Integrated Moving Average

BESS = Battery Energy Storage
System

BMS = Battery Management System

BPNN = Back Propagation Neural
Network

CAES = Compressed Air Energy
Storage

CC = Constant Current

CES = Cryogenic Energy Storage

CFNN = Cascade Forward Neural
Network

CV = Constant Voltage

DeepLIFT = Deep Learning

Important FeaTures

EEMD = Ensemble Empirical Mode
Decomposition

ESS = Efficient Energy Storage
Systems

EV = Electric Vehicle

FES = Flywheel Energy Storage

FFNN = Feed-Forward Neural
Network

GRU = Gated Recurrent Unit

IMMPF = Interacting Multiple

Model Particle Filter

LIME = Local Interpretable Model-
Agnostic Explanations

LRP = Layer-wise Relevance

Propagation

LSTM = Long Short-Term Memory

MAE = Mean Absolute Error

MAPE = Mean Absolute Percentage
Error

MCI = Multi Channel Input profile

MSE = Mean Squared Error

PDF = Probability Density Function

PF = Particle Filter

PHM = Prognostic and Health
Management

PL = Particle Learning

RES = Renewable Energy Sources

RMSE = Root Mean Squared Error

RNN = Recurrent Neural Network

RQ1 = Research Question 1

RQ2 = Research Question 2

RUL = Remaining Useful Life

RVM = Relevance Vector Machine

SCI = Single Channel Input profile

SHAP = SHapley Additive

exPlanations

SMES = Superconducting Magnetic

Energy Storage

SOH = State Of Health

SRU = Simple Recurrent Unit

SVM = Support Vector Machine

XAI = Explainable Artificial

Intelligence

Bibliografia

- Alber, M., Lapuschkin, S., Seegerer, P., Hägele, M., Schütt, K. T., Montavon, G., & Kindermans, P. J. (2019). *iNNvestigate neural networks!* Journal of Machine Learning Research (JMLR).
- Ansari, S., Ayob, A., Hossain Lipu, M. S., Hussain, A., & Saad, M. H. (2021a). *Data-driven remaining useful life prediction for lithium-ion batteries using multi-charging profile framework: A recurrent neural network approach*. Sustainability.
- Ansari, S., Ayob, A., Hossain Lipu, M. S., Hussain, A., & Saad, M. H. (2021b). *Multi-channel profile based artificial neural network approach for remaining useful life prediction of electric vehicle lithium-ion batteries*. Energies.
- Ansari, S., Ayob, A., Hossain Lipu, M. S., Saad, M. H., & Hussain, A. (2021c). *A comparative analysis of lithium ion battery input profiles for remaining useful life prediction by cascade forward neural network*. 2021 IEEE World AI IoT Congress (AIIoT).
- Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K. R., & Samek, W. (2015). *On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation*. PloS one.
- Bengio, Y., Simard, P., & Frasconi, P. (1994). *Learning long-term dependencies with gradient descent is difficult*. IEEE transactions on neural networks.
- Biagetti, T., & Sciubba, E. (2004). *Automatic diagnostics and prognostics of energy conversion processes via knowledge-based systems*. Energy.
- Choi, Y., Ryu, S., Park, K., & Kim, H. (2019). *Machine learning-based lithium-ion battery capacity estimation exploiting multi-channel charging profiles*. Ieee Access.
- Hannan, M. A., Lipu, M. H., Hussain, A., Ker, P. J., Mahlia, T. I., Mansor, M., & Dong, Z. Y. (2020). *Toward enhanced state of charge estimation of lithium-ion batteries using optimized machine learning techniques*. Scientific reports.
- Hannan, M. A., Lipu, M. H., Ker, P. J., Begum, R. A., Agelidis, V. G., & Blaabjerg, F. (2019). *Power electronics contribution to renewable*

- energy conversion addressing emission reduction: Applications, issues, and recommendations.* Applied Energy.
- Hochreiter, S. (1998). *The vanishing gradient problem during learning recurrent neural nets and problem solutions.* International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems.
- Hochreiter, S., & Schmidhuber, J. (1997). *Long short-term memory.* Neural computation.
- Li, J., Lyu, C., Wang, L., Zhang, L., & Li, C. (2014). *Remaining capacity estimation of Li-ion batteries based on temperature sample entropy and particle filter.* Journal of Power Sources.
- Li, X., Zhang, L., Wang, Z., & Dong, P. (2019). *Remaining useful life prediction for lithium-ion batteries based on a hybrid model combining the long short-term memory and Elman neural networks.* Journal of Energy Storage.
- Liao, L., & Köttig, F. (2014). *Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction.* IEEE Transactions on Reliability.
- Lipu, M. S., Hannan, M. A., Hussain, A., Saad, M. H., Ayob, A., & Blaabjerg, F. (2018). *State of charge estimation for lithium-ion battery using recurrent NARX neural network model based lightning search algorithm.* IEEE access.
- Liu, D., Zhou, J., Liao, H., Peng, Y., & Peng, X. (2015). *A health indicator extraction and optimization framework for lithium-ion battery degradation modeling and prognostics.* IEEE Transactions on Systems, Man, and Cybernetics.
- Liu, J., Saxena, A., Goebel, K., Saha, B., & Wang, W. (2010). *An adaptive recurrent neural network for remaining useful life prediction of lithium-ion batteries.* Annual Conference of the PHM Society.
- Liu, Z., Jia, Z., Vong, C. M., Han, J., Yan, C., & Pecht, M. (2018). *A patent analysis of prognostics and health management (PHM) innovations for electrical systems.* IEEE Access.

- Liu, Z., Sun, G., Bu, S., Han, J., Tang, X., & Pecht, M. (2016). *Particle learning framework for estimating the remaining useful life of lithium-ion batteries*. IEEE Transactions on Instrumentation and Measurement.
- Lundberg, S. M., & Lee, S. I. (2017). *A unified approach to interpreting model predictions*. Advances in neural information processing systems.
- Lyu, C., Lai, Q., Ge, T., Yu, H., Wang, L., & Ma, N. (2017). *A lead-acid battery's remaining useful life prediction by using electrochemical model in the Particle Filtering framework*. Energy.
- Małek, A., & Marciniak, A. (2021). *The use of deep recurrent neural networks to predict performance of photovoltaic system for charging electric vehicles*. Open Engineering.
- Medsker, L. R., & Jain, L. C. (1999). *Recurrent neural networks: design and applications*. CRC press.
- Nuhic, A., Terzimehic, T., Soczka-Guth, T., Buchholz, M., & Dietmayer, K. (2013). *Health diagnosis and remaining useful life prognostics of lithium-ion batteries using data-driven methods*. Journal of power sources.
- Park, K., Choi, Y., Choi, W. J., Ryu, H. Y., & Kim, H. (2020). *LSTM-based battery remaining useful life prediction with multi-channel charging profiles*. Ieee Access.
- Patil, M. A., Tagade, P., Hariharan, K. S., Kolake, S. M., Song, T., Yeo, T., & Doo, S. (2015). *A novel multistage Support Vector Machine based approach for Li ion battery remaining useful life estimation*. Applied energy.
- Plett, G. L. (2004). *Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 3. State and parameter estimation*. Journal of Power sources.
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). *"Why Should I Trust You?": Explaining the Predictions of Any Classifier*. Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining.

- Saha, B., Goebel, K., Poll, S., & Christophersen, J. (2008). *Prognostics methods for battery health monitoring using a Bayesian framework*. IEEE Transactions on instrumentation and measurement.
- Schlegel, U., Arnout, H., El-Assady, M., Oelke, D., & Keim, D. A. (2019). *Towards a rigorous evaluation of XAI methods on time series*. 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW).
- Shapley, L. S. (1953). *A value for n-person games*.
- Shrikumar, A., Greenside, P., & Kundaje, A. (2017). *Learning important features through propagating activation differences*. International conference on machine learning.
- Simonyan, K., Vedaldi, A., & Zisserman, A. (2019). *Deep inside convolutional networks: Visualising image classification models and saliency maps*. arXiv.
- Su, X., Wang, S., Pecht, M., Zhao, L., & Ye, Z. (2017). *Interacting multiple model particle filter for prognostics of lithium-ion batteries*. Microelectronics Reliability.
- Wu, J., Zhang, C., & Chen, Z. (2016). *An online method for lithium-ion battery remaining useful life estimation using importance sampling and neural networks*. Applied energy.
- Yang, Y., Wang, Z., Guo, P., Chen, S., Bian, H., Tong, X., & Ni, L. (2021). *Carbon oxides emissions from lithium-ion batteries under thermal runaway from measurements and predictive model*. Journal of Energy Storage.
- Zhang, Y., Xiong, R., He, H., & Pecht, M. G. (2018a). *Lithium-ion battery remaining useful life prediction with Box–Cox transformation and Monte Carlo simulation*. IEEE Transactions on Industrial Electronics.
- Zhang, Y., Xiong, R., He, H., & Pecht, M. G. (2018b). *Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries*. IEEE Transactions on Vehicular Technology.
- Zhou, Y., & Huang, M. (2016). *Lithium-ion batteries remaining useful life prediction based on a mixture of empirical mode decomposition and ARIMA model*. Microelectronics Reliability.

