Tecniche di Deep Learning per la classificazione predittiva dei guasti in sistemi off-grid alimentati da pannelli solari

Facoltà di Ingegneria dell'informazione, informatica e statistica

Dipartimento di Ingegneria dell'Informazione, Elettronica e Telecomunicazioni

Corso di laurea in Ingegneria Elettronica

Lorenzo Giraldi

1649909

Relatore Correlatore

Massimo Panella Daniele Nerone

A.A. 2021-2022

[1 Introduzione 6](#_Toc104296734)

[2 Deep Learning 9](#_Toc104296735)

[2.1 Recursive Neural Network - RNN 9](#_Toc104296736)

[2.2 Long Short Term Memory - LSTM 16](#_Toc104296737)

[3 Manutenzione predittiva 19](#_Toc104296738)

[3.1 Uso e definizione 19](#_Toc104296739)

[3.2 Stato dell’arte 24](#_Toc104296740)

[3.3 Caso d’uso - architettura del sistema 26](#_Toc104296741)

[3.3.1 Acquisizione e trasmissione dei dati 26](#_Toc104296742)

[3.3.2 Salvataggio dei campioni nel database 32](#_Toc104296743)

[4 Elaborazione dei dati 33](#_Toc104296744)

[4.1 Presentazione dei dati per costruire il dataset 33](#_Toc104296745)

[4.2 Preelaborazione dei dati (Sincronizzazione) 35](#_Toc104296746)

[4.3 Creazione di sequenze della durata di 3 giorni 41](#_Toc104296747)

[4.3.1 Estrazione delle sequenze 43](#_Toc104296748)

[4.3.2 Identificazione delle sequenze sane e patologiche 46](#_Toc104296749)

[4.3.3 Normalizzazione delle sequenze 56](#_Toc104296750)

[4.3.4 Formazione del dataset finale inglobando le sequenze di tutti i dispositivi 57](#_Toc104296751)

[4.4 Partizione statica e dinamica del dataset 64](#_Toc104296752)

[4.5 Creazione di sequenze della durata di 1 giorno 65](#_Toc104296753)

[5 Reti neurali proposte 66](#_Toc104296754)

[5.1 Composizione della rete 66](#_Toc104296755)

[5.2 Parametri 67](#_Toc104296756)

[5.3 Training Options 68](#_Toc104296757)

[5.4 Training e classificazione 70](#_Toc104296758)

[5.5 K-Folding 72](#_Toc104296759)

[6 Risultati sperimentali 73](#_Toc104296760)

[6.1 Rete neurale di base 74](#_Toc104296761)

[6.1.1 Tensione della cella minima e potenza del pannello 74](#_Toc104296762)

[6.1.1.1 Previsione ad 1 giorno con sequenze lunghe 3 giorni 74](#_Toc104296763)

[6.1.1.2 Previsione ad 1 giorno con sequenze lunghe 1 giorno 75](#_Toc104296764)

[6.1.1.3 Previsione a 3 giorni con sequenze lunghe 3 giorni 76](#_Toc104296765)

[6.1.1.4 Previsione a 3 giorni con sequenze lunghe 1 giorno 77](#_Toc104296766)

[6.1.1.5 Previsione a 7 giorni con sequenze lunghe 3 giorni 78](#_Toc104296767)

[6.1.1.6 Previsione a 7 giorni con sequenze lunghe 1 giorno 79](#_Toc104296768)

[6.1.2 Tensione della cella minima, potenza del pannello, SOC e irraggiamento 80](#_Toc104296769)

[6.1.2.1 Previsione ad 1 giorno con sequenze lunghe 3 giorni 80](#_Toc104296770)

[6.1.2.2 Previsione ad 1 giorno con sequenze lunghe 1 giorno 81](#_Toc104296771)

[6.1.2.3 Previsione a 3 giorni con sequenze lunghe 3 giorni 82](#_Toc104296772)

[6.1.2.4 Previsione a 3 giorni con sequenze lunghe 1 giorno 83](#_Toc104296773)

[6.1.2.5 Previsione a 7 giorni con sequenze lunghe 3 giorni 84](#_Toc104296774)

[6.1.2.6 Previsione a 7 giorni con sequenze lunghe 1 giorno 85](#_Toc104296775)

[6.1.3 Tensione della cella minima, potenza del pannello, SOC, irraggiamento e bilancio delle correnti della batteria 86](#_Toc104296776)

[6.1.3.1 Previsione ad 1 giorno con sequenze lunghe 3 giorni 86](#_Toc104296777)

[6.1.3.2 Previsione ad 1 giorno con sequenze lunghe 1 giorno 87](#_Toc104296778)

[6.1.3.3 Previsione a 3 giorni con sequenze lunghe 3 giorni 88](#_Toc104296779)

[6.1.3.4 Previsione a 3 giorni con sequenze lunghe 1 giorno 89](#_Toc104296780)

[6.1.3.5 Previsione a 7 giorni con sequenze lunghe 3 giorni 90](#_Toc104296781)

[6.1.3.6 Previsione a 7 giorni con sequenze lunghe 1 giorno 91](#_Toc104296782)

[6.1.4 Confronto delle accuratezze in base alla durata delle sequenze 92](#_Toc104296783)

[6.1.5 Confronto delle accuratezze in base alla combinazione di grandezze 96](#_Toc104296784)

[6.2 Prima rete alternativa 100](#_Toc104296785)

[6.2.1 Tensione della cella minima e potenza del pannello 101](#_Toc104296786)

[6.2.1.1 Previsione ad 1 giorno con sequenze lunghe 3 giorni 101](#_Toc104296787)

[6.2.1.2 Previsione ad 1 giorno con sequenze lunghe 1 giorno 102](#_Toc104296788)

[6.2.1.3 Previsione a 3 giorni con sequenze lunghe 3 giorni 103](#_Toc104296789)

[6.2.1.4 Previsione a 3 giorni con sequenze lunghe 1 giorno 104](#_Toc104296790)

[6.2.1.5 Previsione a 7 giorni con sequenze lunghe 3 giorni 105](#_Toc104296791)

[6.2.1.6 Previsione a 7 giorni con sequenze lunghe 1 giorno 106](#_Toc104296792)

[6.2.2 Tensione della cella minima, potenza del pannello, SOC e irraggiamento 107](#_Toc104296793)

[6.2.2.1 Previsione ad 1 giorno con sequenze lunghe 3 giorni 107](#_Toc104296794)

[6.2.2.2 Previsione ad 1 giorno con sequenze lunghe 1 giorno 108](#_Toc104296795)

[6.2.2.3 Previsione a 3 giorni con sequenze lunghe 3 giorni 109](#_Toc104296796)

[6.2.2.4 Previsione a 3 giorni con sequenze lunghe 1 giorno 110](#_Toc104296797)

[6.2.2.5 Previsione a 7 giorni con sequenze lunghe 3 giorni 111](#_Toc104296798)

[6.2.2.6 Previsione a 7 giorni con sequenze lunghe 1 giorno 112](#_Toc104296799)

[6.2.3 Tensione della cella minima, potenza del pannello, SOC, irraggiamento e bilancio delle correnti della batteria 113](#_Toc104296800)

[6.2.3.1 Previsione ad 1 giorno con sequenze lunghe 3 giorni 113](#_Toc104296801)

[6.2.3.2 Previsione ad 1 giorno con sequenze lunghe 1 giorno 114](#_Toc104296802)

[6.2.3.3 Previsione a 3 giorni con sequenze lunghe 3 giorni 115](#_Toc104296803)

[6.2.3.4 Previsione a 3 giorni con sequenze lunghe 1 giorno 116](#_Toc104296804)

[6.2.3.5 Previsione a 7 giorni con sequenze lunghe 3 giorni 117](#_Toc104296805)

[6.2.3.6 Previsione a 7 giorni con sequenze lunghe 1 giorno 118](#_Toc104296806)

[6.2.4 Confronto delle accuratezze in base alla durata delle sequenze 119](#_Toc104296807)

[6.2.5 Confronto delle accuratezze in base alla combinazione di grandezze 123](#_Toc104296808)

[6.3 Seconda rete alternativa 126](#_Toc104296809)

[6.3.1 Tensione della cella minima e potenza del pannello 127](#_Toc104296810)

[6.3.1.1 Previsione ad 1 giorno con sequenze lunghe 3 giorni 127](#_Toc104296811)

[6.3.1.2 Previsione ad 1 giorno con sequenze lunghe 1 giorno 128](#_Toc104296812)

[6.3.1.3 Previsione a 3 giorni con sequenze lunghe 3 giorni 129](#_Toc104296813)

[6.3.1.4 Previsione a 3 giorni con sequenze lunghe 1 giorno 130](#_Toc104296814)

[6.3.1.5 Previsione a 7 giorni con sequenze lunghe 3 giorni 131](#_Toc104296815)

[6.3.1.6 Previsione a 7 giorni con sequenze lunghe 1 giorno 132](#_Toc104296816)

[6.3.2 Tensione della cella minima, potenza del pannello, SOC e irraggiamento 133](#_Toc104296817)

[6.3.2.1 Previsione ad 1 giorno con sequenze lunghe 3 giorni 133](#_Toc104296818)

[6.3.2.2 Previsione ad 1 giorno con sequenze lunghe 1 giorno 134](#_Toc104296819)

[6.3.2.3 Previsione a 3 giorni con sequenze lunghe 3 giorni 135](#_Toc104296820)

[6.3.2.4 Previsione a 3 giorni con sequenze lunghe 1 giorno 136](#_Toc104296821)

[6.3.2.5 Previsione a 7 giorni con sequenze lunghe 3 giorni 137](#_Toc104296822)

[6.3.2.6 Previsione a 7 giorni con sequenze lunghe 1 giorno 138](#_Toc104296823)

[6.3.3 Tensione della cella minima, potenza del pannello, SOC, irraggiamento e bilancio delle correnti della batteria 139](#_Toc104296824)

[6.3.3.1 Previsione ad 1 giorno con sequenze lunghe 3 giorni 139](#_Toc104296825)

[6.3.3.2 Previsione ad 1 giorno con sequenze lunghe 1 giorno 140](#_Toc104296826)

[6.3.3.3 Previsione a 3 giorni con sequenze lunghe 3 giorni 141](#_Toc104296827)

[6.3.3.4 Previsione a 3 giorni con sequenze lunghe 1 giorno 142](#_Toc104296828)

[6.3.3.5 Previsione a 7 giorni con sequenze lunghe 3 giorni 143](#_Toc104296829)

[6.3.3.6 Previsione a 7 giorni con sequenze lunghe 1 giorno 144](#_Toc104296830)

[6.3.4 Confronto delle accuratezze in base alla durata delle sequenze 145](#_Toc104296831)

[6.3.5 Confronto delle accuratezze in base alla combinazione di grandezze 149](#_Toc104296832)

[6.4 Confronto delle prestazioni delle reti neurali 152](#_Toc104296833)

[7 Conclusioni 156](#_Toc104296834)

[8 Appendice 159](#_Toc104296835)

[9 Indice delle figure 168](#_Toc104296836)

[10 Indice delle tabelle 173](#_Toc104296837)

[11 Bibliografia 179](#_Toc104296838)

# Introduzione

Negli ultimi anni, nel quadro dell’efficientamento dei processi industriali, l'interesse per una stima automatica delle condizioni di manutenzione di apparati critici per i processi aziendali, è fortemente aumentato (1). Questo, nell’ottica di mitigare l’occorrenza di guasti potenzialmente forieri di perdite economiche, quali arresti della catena di produzione o impatti sulle catene logistiche a supporto delle attività.

Dato che, ogni impianto è costituito da sottosistemi che operano sinergicamente per eseguire una o più funzioni, è facile capire come l’affidabilità dipenda in sostanza dalle condizioni di lavoro dei componenti. Nel caso in cui un componente si rompa, potrebbe essere necessario arrestare tutto il sistema per effettuare la manutenzione/riparazione e, nel peggiore dei casi, l'arresto improvviso di un sistema potrebbe causare danni gravi a cose e/o persone (7). Per questo motivo, la manutenzione dei componenti è un aspetto essenziale per il corretto funzionamento di ogni sistema.

A questo proposito, l’avvento dell’IoT (*Internet of Things*) ha favorito la raccolta di grandi moli di dati facilitando il monitoraggio di sistemi complessi e permettendo di visualizzarli da remoto (1). Tipicamente, l’andamento dei dati viene rappresentato tramite interfacce grafiche come gli SCADA (*Supervisory Control And Data Acquisition*), che permettono ad un utente specializzato di avere una panoramica sull’andamento delle variabili critiche di un sistema (2). In questo modo, qualora il sistema subisca un guasto, è possibile con maggior facilità risalire alla causa di quest’ultimo ed effettuare la riparazione.

Tuttavia, tale approccio orientato alla sola *Manutenzione Reattiva*, è oggetto di forti critiche, non essendo in grado di prevedere e segnalare una potenziale situazione di guasto in anticipo rispetto all’occorrenza concreta.

Nelle Smart Industries, la M*anutenzione Predittiva* è la tecnica desiderata e ricercata, dal momento che valuta in continuità le condizioni di funzionamento degli apparati, con l'obbiettivo di prevederne possibili malfunzionamenti. Utilizzando questo tipo di tecnica, è possibile fare delle manutenzioni non ordinarie mirate, per prevenire dei guasti e quindi ottimizzare ulteriormente il processo in questione (3).

Per predire lo stato di salute di un sistema, esistono tecniche che fanno uso di sensoristica IoT (o tradizionale), installata direttamente sui componenti, per acquisire misurazioni delle grandezze critiche e compararle con i valori di lavoro standard dei componenti, riportati nelle schede tecniche.

Grazie alla crescita dell’affidabilità delle tecniche di I*ntelligenza Artificiale*, vengono creati dei modelli che consentono di prevedere automaticamente, a partire dalla elaborazione dei dati di cui sopra, se la situazione osservata è prodromica di un guasto, oppure tipica di una situazione di lavoro normale. Ciò, è reso possibile grazie all’estrazione automatica di correlazioni tra il valore delle grandezze e gli eventi di guasto (3).

Nella fattispecie, in questa tesi, sono state utilizzate tecniche specifiche di *Deep Learning*, per realizzare un sistema automatico, in grado di prevenire dei guasti. Nel nostro caso, una condizione di spegnimento dell’apparato, a fronte di uno scaricamento del pacco batterie, è assimilabile, dal punto di vista funzionale, ad un guasto.

Gli apparati monitorati, consistono di sistemi di *Supervisione e Controllo* a base IoT, installati sui tralicci elettrici di Terna in zone remote, alimentati da pannelli solari e dotati di autonomia grazie ad un pacco batterie al litio (LiFePO4). Le dimensioni dei panelli fotovoltaici ed il peso delle batterie, in questo tipo di realizzazioni, devono rispettare limiti stringenti, per minimizzare i carichi meccanici statici e dinamici (carico al vento) imposti dalle normative.

Tali circostanze, sono in conflitto con le esigenze di esercizio dell’operatore elettrico, per il quale, riveste particolare importanza, poter contare su un monitoraggio continuo dello stato dei tralicci, ovvero conoscere in anticipo la probabilità di potenziali interruzioni del servizio causa perdita dell’alimentazione. L’occorrenza di tale evento, se prevista in anticipo, può essere infatti contrastata, diminuendo il consumo dell’apparato operando opportunamente sul *duty cycle* delle misure, e fornendo il tempo necessario di intervenire per evitare che si spengano.

Data la complessità e la varietà dei dati provenienti dall’apparato di supervisione da proteggere (riguardanti batterie, carica batterie, irraggiamento, etc.), è stato necessario sviluppare un modello di *Deep Learning,* opportunamente allenato a distinguere le situazioni critiche da quelle normali, usando sequenze temporali di dati storici appositamente etichettate come sane o patologiche.

Una volta allenato, il modello si è dimostrato in grado di generalizzare le conoscenze acquisite, classificando in modo automatico sequenze sconosciute provenienti dal campo e prevedendo in anticipo, come previsto, situazioni di potenziale spegnimento con obiettivo di rilevare, come richiesto, un evento critico prima che accada ed avere il tempo necessario per intervenire evitando che si verifichi.

Il lavoro, ha previsto un’analisi dei risultati ottenuti, comparando le accuratezze conseguite da tre diverse reti neurali al variare dell’orizzonte di predizione, della lunghezza in giorni delle sequenze e della combinazione delle grandezze date in ingresso.

Il lavoro di tesi è stato strutturato in diverse sezioni: Nella sezione 2 è riportata una breve introduzione teorica alle tecniche di Machine Learning ed in particolare al Deep Learning; una sezione 3 nella quale diamo una definizione del concetto di manutenzione predittiva e ne diamo lo stato dell’arte nel quadro dell’industria contemporanea, per poi presentare il caso di studio sviluppato presso Staer Sistemi; una sezione 3.3 in cui viene dettagliata la realizzazione del *dataset*; la sezione 4 in cui sono presentate le reti neurali utilizzate; la sezione 5 nella quale sono riportati i risultati conseguiti e le considerazioni che ne derivano; la sezione 6 in cui sono presentate le conclusioni finali, oltre ai potenziali ulteriori sviluppi.

# Deep Learning

## Recursive Neural Network - RNN

Il Deep Learning è una branchia del Machine Learning che s’inspira alla struttura del cervello umano, in particolare al neurone, per creare degli algoritmi di apprendimento. Il neurone è una cellula che si divide in neuroni sensori, motori e intermedi, in grado di raccogliere e condurre gli impulsi nervosi. Ce ne sono più di 100.000 nel nostro sistema nervoso e sono fondamentali per ricevere e trasmettere segnali. Una rete neurale cerca di riprodurre il funzionamento del neurone umano, ovvero tutti quei processi che avvengono nel cervello durante la fase di apprendimento e quella successiva del riconoscimento. Come nel cervello, questi algoritmi hanno una struttura a strati gerarchica (*layers*), capace di scomporre la complessità del problema. Gli strati più bassi permettono a quelli più alti l’apprendimento di concetti man mano sempre più astratti. Se immaginiamo che una rete neurale controlli il comportamento di un semplice organismo posto nell'ambiente, le unità di input della rete emulano gli organi sensoriali dell'organismo e le unità di output gli organi motori. L'insieme dei livelli di attivazione delle unità di input codifica lo stato dei diversi tipi di agenti fisici e chimici (luminosi, sonori, meccanici, termici, odori, sapori) presenti in un dato momento nell'ambiente circostante e a cui l'organismo è sensibile. L'attivazione si propaga dalle unità di input alle unità interne e da queste alle unità di output. L'insieme dei livelli di attivazione delle unità di output codifica i movimenti con cui l'organismo risponde all'informazione sensoriale proveniente dall'esterno. Ogni rete ha un'architettura costituita dal numero di unità e dallo schema delle interconnessioni.

In Figura 2‑1 vediamo la struttura di un neurone artificiale (13):

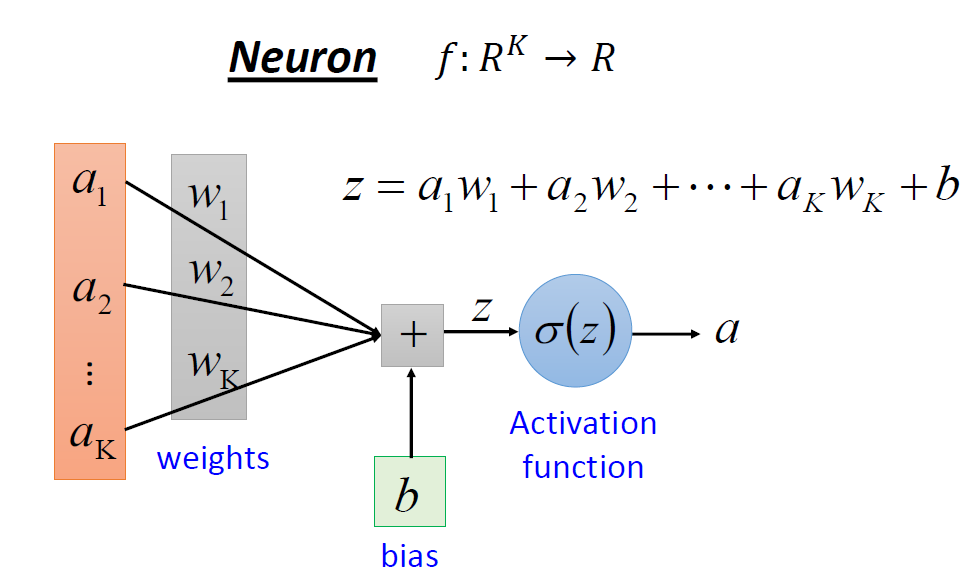


Figura 2‑1: Neurone artificiale

Vediamo che l’uscita del neurone è la combinazione lineare degli ingressi moltiplicati per dei coefficienti che vengono chiamati pesi. A questo termine va sommato un termine di *bias*. La funzione di attivazione che si trova dopo l’uscita ha il compito di pesare l’output del neurone prima che venga passato ad un altro elemento della rete.

Fra i vari tipi di architetture, distinguiamo le architetture *feedforward* (a propagazione in avanti) e le architetture ricorrenti. Un'architettura feedforward (Figura 2‑2) è formata da uno strato di unità di input (input layer), da uno strato di unità di output (*output layer*) e da uno o più strati intermedi di unità interne dette nascoste (*hidden layers*). Ogni unità di ciascuno strato è connessa con tutte le unità dello strato successivo, a partire dallo strato di input fino a quello di output. In ogni ciclo, l'attivazione si propaga in avanti dalle unità di input ai successivi strati di unità interne fino a raggiungere le unità di output (12).

In Figura 2‑2 vediamo lo schema di una rete feedforward con 2 layers nascosti (13):

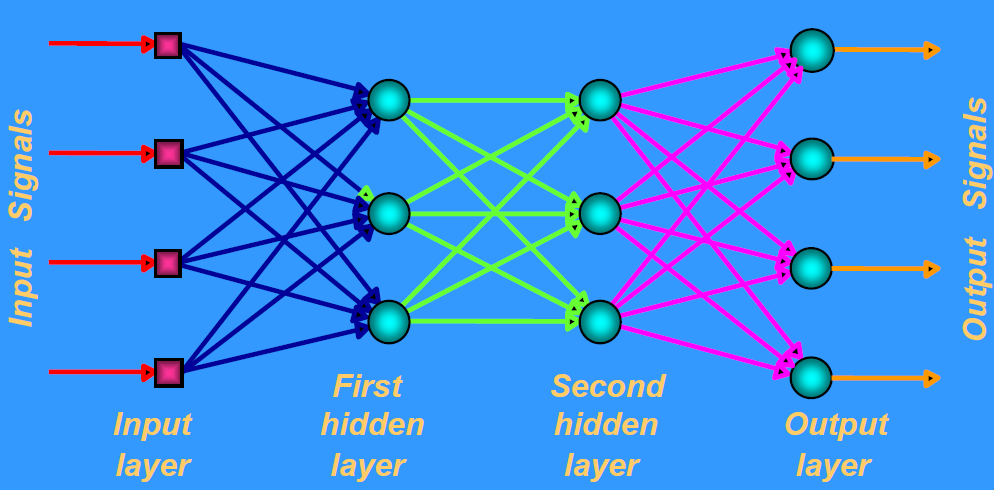


Figura 2‑2: Rete feedforward

A differenza delle architetture feedforward, le architetture ricorrenti presentano connessioni che partono da una data unità e ritornano, direttamente o indirettamente, alla stessa unità. Un tipo di architettura ricorrente può servire a dotare una rete neurale di una memoria che conserva una traccia di quanto è avvenuto nei cicli precedenti. In ogni ciclo input-output, il pattern di attivazione delle unità interne viene memorizzato in uno speciale insieme di unità chiamate [unità di memoria](https://www.treccani.it/enciclopedia/unita-di-memoria_(Enciclopedia_della_Scienza_e_della_Tecnica)/), collegate alle unità interne tramite normali connessioni. Pertanto, in ogni ciclo, il livello di attivazione delle unità interne, e quindi anche l'output della rete, viene a dipendere non solo dall’input in quell’istante, ma anche dalla traccia dello stato della rete nel ciclo precedente, conservata nelle unità di memoria (12).

In Figura 2‑3 vediamo lo schema di una rete ricorrente con un layer nascosto (13):

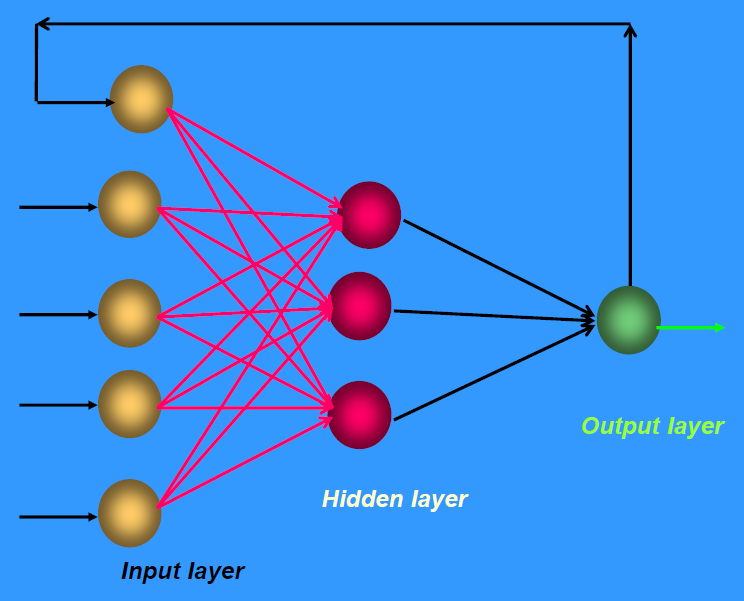


Figura 2‑3: Recurrent Neural Network

In Figura 2‑3 osserviamo la presenza di un feedback rappresentato dalla freccia che parte dallo strato di uscita per tornare allo strato di ingresso. L’informazione portata dal feedback permette di contestualizzare i nuovi dati in ingresso rispetto agli istanti precedenti.

Per creare una rete ben dimensionata è necessario fare attenzione alla complessità della rete portata dal numero di neuroni in ogni strato. In particolare, il numero di neuroni corrisponde alla quantità di informazioni memorizzate ad ogni step. Aumentandone il numero aumenta la complessità della rete in termini di numeri di parametri. Creare una rete neurale complessa, può portare la rete ad avere risultati eccellenti in termini di accuratezza sul set di dati su cui è stata allenata, e di avere risultati pessimi su un set di dati sconosciuti. Questo fenomeno si chiama overfitting, ed è causato dal fatto che durante l’allenamento la rete ha un numero di parametri tale da permettergli da imparare “a memoria” la relazione tra i dati del training set e le sue etichette di riferimento, senza però sviluppare la capacità di generalizzare le conoscenze acquisite a discapito dell’accuratezza su dati sconosciuti (13).

In Figura 2‑4 vediamo uno schema dettagliato di una rete RNN srotolata (14):

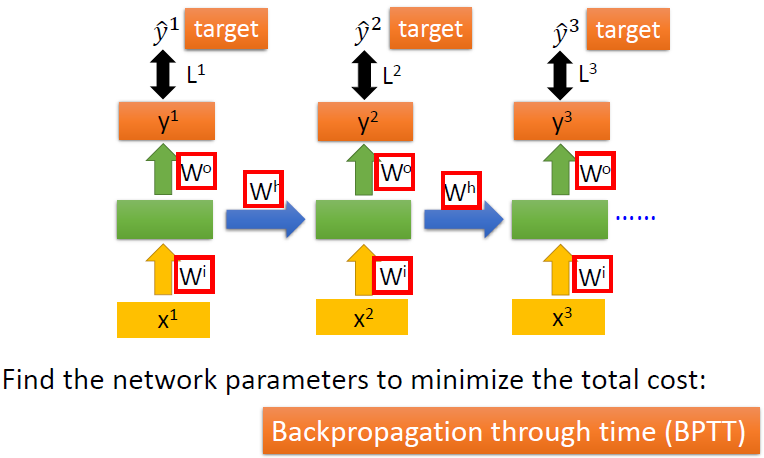


Figura 2‑4: Schema RNN srotolato

In Figura 2‑4, vediamo la stessa cella rappresentata a diversi istanti di tempo, in cui l’uscita al tempo diventa l’ingresso al tempo . Si dice che la rete è srotolata. Con intendiamo gli ingressi, con gli output stimati dalla rete e con i pesi/parametri per l’input, l’output e lo stato. I target rappresentano l’etichetta di riferimento del campione e il discostamento in modulo tra l’output della rete e il riferimento. Vediamo che ad ogni istante di tempo la rete ha due ingressi: . La memoria della rete è rappresentata dallo stato al tempo , ciò implica che dipende da tutti gli ingressi compresi tra l’istante zero e l’istante i.

In queste reti, ricoprono un ruolo di fondamentale importanza i suddetti parametri/pesi. Per ottimizzarli, essi vengono aggiornati ad ogni epoca, ossia ogni volta che tutti i dati contenuti nel training sono stati analizzati dalla rete. Il feedback consiste nel comparare il valore di riferimento della sequenza () con il valore predetto dalla rete () e fare le correzioni ai parametri di conseguenza. Più specificatamente, ad ogni epoca ogni parametro del modello riceve un aggiornamento proporzionale alla [derivata parziale](https://it.wikipedia.org/wiki/Derivata_parziale) della funzione di costo rispetto al parametro stesso (13)(15). L’obbiettivo che si vuole raggiungere è quello di è trovare il minimo globale della funzione di costo. Il costo può essere una distanza Euclidea oppure la cross entropia dell‘uscita della rete rispetto al riferimento (14). Per minimizzare la funzione di costo è molto importante che tutti i parametri vengano ottimizzati tramite il meccanismo di retro-propagazione del gradiente.

A tal proposito, uno dei principali colli di bottiglia delle reti RNN riguarda il meccanismo di retro-propagazione del gradiente, ed è il fenomeno della scomparsa del gradiente (15). Questo fenomeno si presenta quando si usano funzioni di attivazione non lineari classiche (la [tangente iperbolica](https://it.wikipedia.org/wiki/Tangente_iperbolica) o la f[unzione logistica](https://it.wikipedia.org/wiki/Funzione_logistica)) che hanno gradiente a valori nell'intervallo [-1,1] o [0,1], e si hanno dei dati costituiti da sequenze temporali di più di 1000 campioni. Se si verificano queste due condizioni, nell'algoritmo di retro-propagazione, quando i gradienti ai vari strati vengono moltiplicati tramite la [regola della catena](https://it.wikipedia.org/wiki/Regola_della_catena), dopo un certo numero di strati, il gradiente si approssima a zero e quindi non apporta correzione ai pesi. Questo è dovuto al fatto che, il prodotto di n numeri in [0,1] decresce esponenzialmente rispetto ad n (la profondità della rete). Questo fenomeno porta all’annullamento dell’effetto di correzione dei pesi negli strati più vicini all’inizio della sequenza e fa sì che la rete non riesca ad imparare le correlazioni tra due istanti di tempo distanti. Quando invece il gradiente delle funzioni di attivazione assume valori elevati, un problema analogo che può manifestarsi è quello dell'esplosione del gradiente (15). Il problema della scomparsa del gradiente limita fortemente la capacità di apprendimento della rete. Questo problema può essere risolto parzialmente tramite l’uso di funzioni di attivazioni lineari come la ReLU (Rectified Linear Unit) (17).

Un’importante innovazione per le reti RNN fu introdotta nel 1997 da Mike Schuster e Kuldip K. Paliwal che presentarono per la prima volta il concetto di RNN bidirezionale (18). Il concetto di queste reti è di avere un ingresso che arriva dall’istante successivo, al tempo (t+1), e uno dall’istante precedente, al tempo (t-1), oltre che all’input al tempo t.

La figura esplicita il concetto (14):

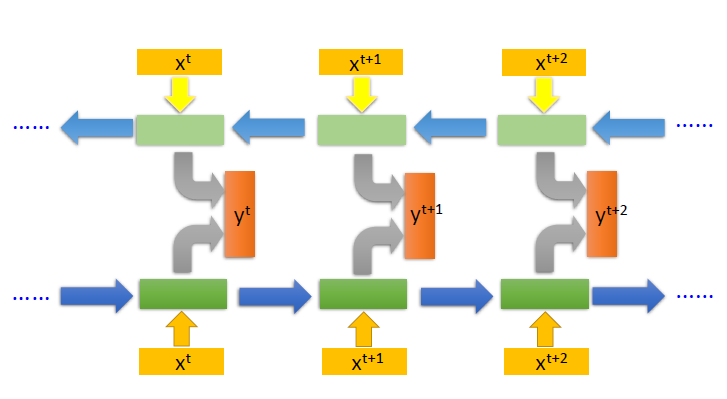


Figura 2‑5: Rete RNN bidirezionale

In Figura 2‑5 abbiamo due reti RNN sovrapposte che prendono in ingresso le stesse sequenze ma nell’ordine opposto. Le sequenze vengono lette dal primo all’ultimo e contemporaneamente dall’ultimo al primo elemento. Ciò permette di classificare l’elemento all’istante t grazie ai campioni precedenti ma anche a quelli futuri. Per poter avere una rete bidirezionale bisogna avere accesso a tutta la sequenza al momento della previsione.

Come vediamo nel seguente esempio, una rete bidirezionale può essere utile nel caso in cui si debba prevedere le parole in una frase (20):

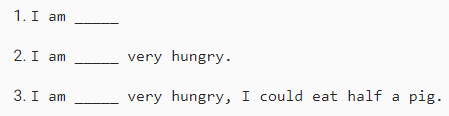


Figura 2‑6: Previsione delle parole in una frase in base al resto della frase

In Figura 2‑6, le soluzioni sono “hungry”, “not” e “very”. Per saperlo, nella prima riga ci bastano le informazioni precedenti, mentre nei due casi seguenti diventa fondamentale il resto della frase per poter contestualizzare.

## Long Short Term Memory - LSTM

Per risolvere il problema della scomparsa del gradiente, nel 1997 [Sepp Hochreiter](https://it.wikipedia.org/w/index.php?title=Sepp_Hochreiter&action=edit&redlink=1) e [Jürgen Schmidhuber](https://it.wikipedia.org/wiki/J%C3%BCrgen_Schmidhuber" \o "Jürgen Schmidhuber) creano le reti LSTM (Long-Short Term Memory) (20).

In Figura 2‑7**Errore. L'origine riferimento non è stata trovata.** vediamo uno schema esplicativo di un neurone LSTM (19):

Immagine che contiene testo, orologio

Descrizione generata automaticamente

Figura 2‑7: Cella LSTM

In Figura 2‑7, vediamo lo schema di un singolo neurone, anche chiamato cella, che presenta 3 ingressi e 2 uscite. La cella prende in ingresso l’uscita della cella al tempo t-1 (), l’input al tempo t () e lo stato della cella al tempo t-1 (), e ritorna in uscita lo stato al tempo t () e l’uscita (hidden state) al tempo t (). Inoltre, la cella fa uso di funzioni di attivazione non lineari come la sigmoide (rossa) e la tangente iperbolica (blu) che formano quelli che vengono chiamati gate. I gate permettono di controllare quale informazione viene propagata e quale no. È da notare, come i vettori e vengono concatenati prima di entrare nei vari gate.

A differenza delle reti RNN, le celle LSTM sono dotate di uno stato interno, , che gli consente di tenere memoria delle informazioni rilevanti nella totalità dei campioni precedenti, e che verrà dato in ingresso al neurone all’istante di tempo successivo.

Vediamo le funzionalità dei gate della cella LSTM (19):

* Il *Forget Gate* prende in ingresso l’input al tempo t e l’output della cella al tempo (t-1). Grazie ad una funzione di attivazione sigmoide nell’intervallo [0,1] stabilisce quali informazione salvare o meno nello stato della cella. In particolare, tramite una moltiplicazione con la sigmoide, l’informazione viene cancellata (sigmoide uguale a 0), salvata (sigmoide uguale a 1), oppure attenuata (sigmoide con valore compreso tra 0 e 1). Ciò permette alla rete di selezionare quali sono le informazioni rilevanti e quali invece vanno scartate, cosa che non accade nelle RNN. Il coefficiente risultante va moltiplicato con lo stato della cella al tempo (t-1).
* L’*Input Gate* prende in ingresso l’input al tempo t e l’output della cella al tempo (t-1). Essi passano in due rami diversi, uno con una sigmoide e uno con una funzione di attivazione tangente iperbolica. I risultati di questi due rami vengono moltiplicati tra loro. La tangente iperbolica normalizza i valori in [-1,1] così da non subire l’esplosione di valori che durante il processo iterativo potrebbero essere incrementati/moltiplicati diverse volte andando a sregolare il processo di apprendimento. La sigmoide seleziona i valori normalizzati. L’output si somma allo stato della cella al tempo (t-1) per creare lo stato corrente.
* L’*Output Gate* prende in ingresso l’input al tempo t e l’output della cella al tempo (t-1) e li passa attraverso una sigmoide. La risultante viene moltiplicata per lo stato della cella normalizzato da una tangente iperbolica e forma l’output della cella, anche detto hidden state*.*

Le equazioni che legano i rami tra loro sono le seguenti (19):

Vediamo che l’equazione dello stato corrente ( della cella dipende dai parametri del Forget Gate (, dell’Input Gate (, e dello stato precedente (. L’output corrente della cella è il prodotto dell’output gate con la tangente iperbolica dello stato corrente.

Le equazioni dei vari gate sono le seguenti (19):

Dove e rappresentano rispettivamente il peso e il bias per ogni gate. La rappresenta la funzione di attivazione sigmoide.

I gate sono un’innovazione significativa, che permette alla rete LSTM di ricordare solo le informazioni rilevanti contenute all’interno di una sequenza. Filtrando in questo modo le informazioni, la rete riesce ad apprendere correlazioni tra istanti di tempo lontani all’interno di una sequenza.

Come per le RNN anche le reti LSTM possono essere bidirezionali ed apportano lo stesso tipo di vantaggio che le RNN. Nel caso delle reti BiLSTM avremmo due layers LSTM consecutivi e quindi il numero di parametri generati dalla rete sarà doppio rispetto al caso LSTM semplice.

# Manutenzione predittiva

## Uso e definizione

Negli ultimi anni l’avvento dell’IoT ha permesso, grazie a sensori posizionati sempre più diffusamente su tutti i componenti di un sistema, di accedere a misurazioni e dati in grado di definire/descrivere in modo accurato tutte le fasi di un processo. Questa disponibilità di dati, campionati in modo discreto o continuo a seconda dei casi, sta portando una quantità tale di informazioni da influenzare fortemente il modo di prendere decisioni. L’informazione portata da questi dati, se sfruttata e decifrata correttamente, permette di avere un quadro migliore dello stato di salute dei componenti di un sistema e dunque di ottimizzarne la manutenzione.

Nella fattispecie, la norma UNE-EN 13306. 2018 definisce la manutenzione come una “Combinazione di tutte le azioni tecniche, amministrative e gestionali, durante il ciclo di vita di un’entità, destinate a mantenerla o riportarla in uno stato in cui possa eseguire la funzione richiesta”(21). A partire da questa definizione, possiamo differenziare i diversi tipi di manutenzione come in Figura 3‑1 (4):

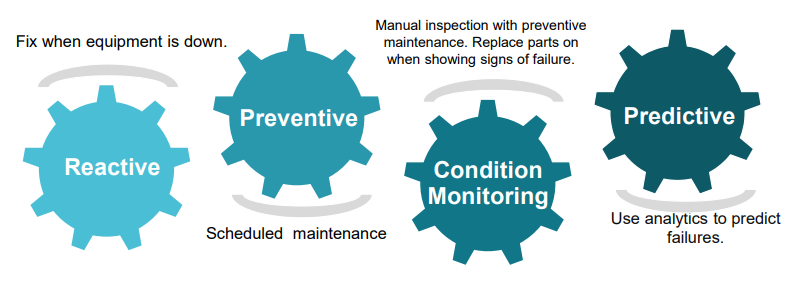


Figura 3‑1: Diversi tipi di manutenzione

In Figura 3‑1 vediamo i 4 tipi di manutenzione: reattiva, preventiva, via monitoraggio dei componenti e predittiva. Le prime tre tecniche di manutenzione sono convenzionali, poiché non vengono effettuate in modo automatico e consistono nella manutenzione di un componente, dopo che si sia rotto, prima che si rompa anche se non dà segni di malfunzionamento (manutenzione programmata), oppure consiste nel monitoraggio da parte di un utente delle condizioni del componente per rilevare comportamenti anomali e sostituirlo/ripararlo prima che si rompa. Questo tipo di approccio è caratterizzato da perdite economiche dovute alla sostituzione di componenti ancora funzionanti, da periodi di fermo degli apparati mediamente lunghi, e nei casi di manutenzione reattiva può portare anche a danni a persone e/o cose. Per evitare questo tipo di imprevisti è stato introdotto l’uso di strumenti analitici automatici per la manutenzione predittiva in grado di sfruttare la grande mole di dati altrimenti difficilmente interpretabile da un utente. Inoltre, l’utilizzo di sistemi automatizzati permette di eliminare l’incertezza e l’errore portato dal lavoro umano poiché implementa delle procedure sistematiche.

A seguire uno schema dell’elaborazione dei dati per effettuare manutenzione predittiva (4):

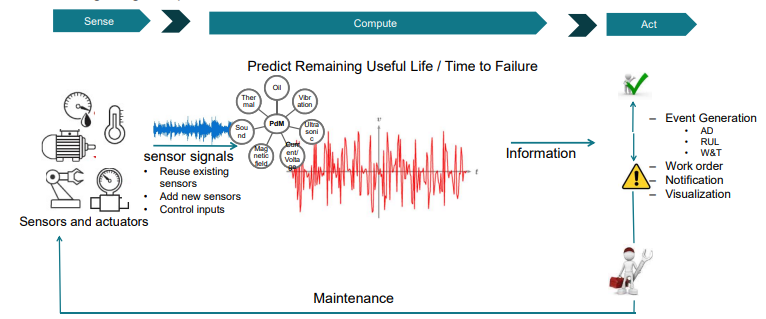


Figura 3‑2: Schema di manutenzione predittiva

Come vediamo in Figura 3‑2, i dati vengono raccolti dal campo, passati all’algoritmo di previsione ed elaborati. Una volta estratte le informazioni necessarie dai dati, delle notifiche vengono mandate all’utente per informarlo sullo stato di salute dei componenti.

È stato dimostrato che la manutenzione facente uso del machine learning fornisce sempre più soluzioni efficaci grazie alla crescita delle funzionalità hardware dei sensori, alla potenza di calcolo accessibile via cloud e all’introduzione di nuovi algoritmi all'avanguardia (6). Per questo motivo il machine learning sta diventando essenziale per una gestione efficiente delle attività di manutenzione e per ridurre i costi associati ai tempi di fermo e ai prodotti difettosi.

Tra i vari approcci alla manutenzione predittiva distinguiamo tre categorie: La manutenzione predittiva basata su un modello fisico del sistema, quella basata sui dati (data-driven) e quella ibrida. Il primo si basa sulla conoscenza del sistema per costruire una modello matematico/fisico dell’andamento nel tempo della degradazione del componente. In questo caso è facile comprendere il significato fisico dei risultati forniti, ma è difficile implementare un modello accurato in sistemi complessi perché diventa molto difficile conoscere e modellizzare tutte le correlazioni tra le variabili. L’approccio data-driven si basa sull’analisi di dati storici descriventi il funzionamento del sistema. Confrontando questi dati con i dati ricevuti si può predire lo stato futuro del sistema, senza però averne un modello fisico. Questo tipo di approccio si basa su metodi statistici, funzioni di affidabilità e metodi di intelligenza artificiale. Un vantaggio dei sistemi data-driven è che sono più abbordabili per sistemi complessi poiché permettono di ottenere risultati utili anche se non si ha una completa conoscenza della complessità del sistema. Tuttavia, per questo stesso motivo, le correlazioni tra i risultati ottenuti ed il problema sono più difficili da interpretare da un punto di vista fisico. La maggiore accessibilità ai dati, favorita dalla crescita tecnologica dell’IoT, sta permettendo ai sistemi data-driven di guadagnare popolarità dato che permettono di avere accesso a quella grande mole di dati che richiedono per essere affidabili. In particolare, la branchia facente uso di reti di deep learning sta emergendo poiché riesce ad estrarre correlazioni complesse dai dati. Infine, l'approccio ibrido combina i due approcci sopra citati.

L’introduzione di tali sistemi implica l’impiego di risorse e una serie di costi che riassumiamo in Figura 3‑3 (5):

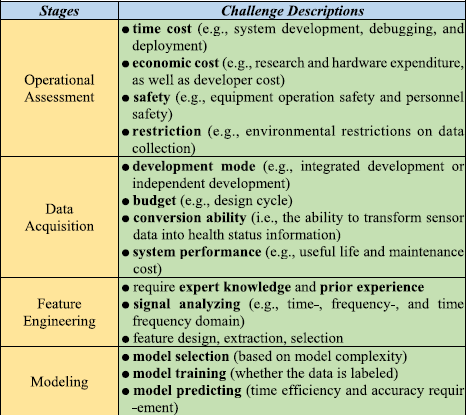


Figura 3‑3: Costi dell'implementazione di un sistema di manutenzione predittiva

Vediamo come la transizione da una manutenzione tradizionale ad un sistema di manutenzione predittiva porti ad un impiego di risorse economiche, di tempo e di persone. Le spese economiche vanno dal design del progetto, all’acquisto dell’hardware (sensori, microcontrollori, etc..), all’acquisizione e l’elaborazione dei dati, fino allo sviluppo del modello e al test. Dato che l’approccio è data-driven, l’acquisizione del dato è cruciale per il corretto funzionamento del sistema. Il trade-off tra i costi dello sviluppo del sistema di manutenzione predittiva e le perdite economiche dovute a guasti dei componenti è solitamente positivo e porta un valore aggiunto al processo. In particolare, esso permette di avere una panoramica più chiara dello stato di salute del sistema.

Per giustificare maggiormente l’investimento che rappresenta una sistema di manutenzione predittiva, forniamo qualche esempio di casi gravi che si sarebbero potuti prevenire, ed in cui il guasto di uno o più componenti abbia portato a gravi conseguenze. Nel giugno del 2009, una linea della metropolitana di Washington DC si è schiantata uccidendo 9 persone e ferendo dozzine di persone. Il guasto è stato attribuito ad anomalie della sensoristica posta sotto i binari della metropolitana (7). In Brasile, nel novembre del 2009, diversi blackout hanno causato disagi a più di 60 milioni di persone e arrestato diversi servizi, tra cui la metropolitana e l’illuminazione pubblica. Nonostante la causa di questi imprevisti sia stata attribuita a fattori metereologici come fulmini, vento e pioggia, è chiaro che è stato complice anche un guasto tecnico (7). Un ultimo caso si riscontra nel famoso malfunzionamento delle automobili Toyota che subivano accelerazioni involontarie e repentine causando, problemi per la sicurezza degli utenti, diversi morti, ed il ritiro di circa 10 milioni di automobili dal mercato da parte dell’azienda, senza contare i danni d’immagine (8).

## Stato dell’arte

Andiamo ora a presentare lo stato dell’arte della manutenzione predittiva per approcci data-driven facenti uso del machine learning. Come detto, l’approccio data-driven permette di analizzare dei sistemi senza averne un modello fisico e per questo motivo sta trovando sbocchi in diversi settori.

Nel settore dell’agricoltura, troviamo un caso pratico in cui la manutenzione predittiva viene usata per monitorare lo stato di salute sia di una piantagione di patate sia di una piantagione di pomodori, con l’obbiettivo di predire ed evitare una possibile infezione della piantagione che potrebbe portare alla perdita del raccolto. Tramite sensori di temperatura sparsi nella piantagione, lo stato di salute viene monitorato di modo da predire, 8 giorni prima del suo avvenimento, un’infezione della coltura. Facendo una classificazione, è possibile risalire all’entità del problema che si sta prevedendo tramite dei livelli di gravità e quindi è possibile decidere con precisione l’operazione da eseguire. I risultati ottenuti in termini di accuratezza sono del 99,97% (22).

Nel settore ferroviario, troviamo un caso di studio riguardante il rilevamento di imperfezioni/anomalie dei binari dei treni che possano portare a guasti, e la loro classificazione in ordine di gravità. Il caso è stato ampiamente studiato, dato che i modelli fisici dei binari non bastano per prevedere la natura stocastica di alcune anomalie. Per risolvere il problema viene implementato un monitoraggio continuo delle condizioni dei binari. Questo studio presenta un nuovo approccio per prevedere i difetti delle rotaie e della loro geometria e pianificare le attività d’ispezione e di manutenzione. In questo caso, l’approccio è un ibrido tra l’approccio fisico e l’approccio data-driven. La rete usata è una rete RNN. I risultati ottenuti riguardano la previsione del numero e del tipo di difetti presenti sul binario rispetto all’ultima manutenzione. I risultati mostrano un’accuratezza del 82%. Questo valore relativamente basso è stato attribuito ad una rumorosità dei dati formanti il dataset (23).

Un caso di particolare interesse potrebbe essere quello della manutenzione predittiva applicata a motori industriali (24). L’applicazione in questione, fa uso della grande mole di dati fornita dai sensori elettrochimici per allenare una rete di deep learning a classificare le sequenze come patologiche o sane. Il dataset fornisce una serie di misurazioni relative a varie turboventole di un motore per aerei (temperatura, pressione, velocità della turboventola, etc…) in diverse condizioni ambientali, al fine di monitorarne le condizioni di degradazione. L’obbiettivo è quello classificare le sequenze in ingresso per predire un guasto con largo anticipo. I risultati ottenuti variano in base all’anticipo con cui si vuole prevedere il guasto. L’intervallo di tempo in questo caso viene specificato in numero di cicli del motore rimanenti prima del guasto. Usando una rete LSTM e posizionandoci 50 cicli prima del guasto, otteniamo un’accuratezza del 91%, mentre a 5 cicli dal guasto l’accuratezza è del 100% (25). Altri tipi di approccio per predire i guasti sono quelli di creare una rete in grado di calcolare il tempo di vita utile rimanente del componente, anche detto RUL (Remaining Useful Life), oppure di creare una rete in grado di prevedere i futuri valori delle grandezze in base ai dati storici (24).

Un’altra applicazione industriale, viene proposta per lo studio dello stato di funzionamento delle macchine utensili a controllo numerico computerizzato (CNC). Esse svolgono un ruolo importante nel settore moderno manufatturiero. Queste macchine sono utilizzate per la sagomatura o la lavorazione di materiali rigidi come il metallo, di solito tagliando, alesando, rettificando, o deformando in altre forme. A causa del prolungato funzionamento, sul lungo termine le abrasioni e lo stress dei suoi componenti meccanici sono inevitabili, con conseguente scarsa qualità del prodotto e bassa efficienza. Nell'industria moderna, quasi il 79,6% dei tempi di inattività di una macchina utensile è causata da guasti meccanici. Per ridurre al minimo il tempo di fermo degli apparati si misurano e analizzano le vibrazioni dei componenti riuscendo a prevedere quando un componente è in procinto di rompersi. Il problema che si riscontra all’atto pratico, è che le vibrazioni corrispondenti ad un cambio di modalità o condizioni di lavoro del macchinario spesso possono essere confuse con quelle dovute a guasti. A tal proposito, in questo articolo viene allenata una rete di deep learning per distinguere questi due tipi di vibrazioni in modo automatico. I risultati ottenuti dalla classificazione sono pari ad un’accuratezza del 98,1% (24).

## Caso d’uso - architettura del sistema

### Acquisizione e trasmissione dei dati

Gli apparati presenti sul campo sono dei sistemi di monitoraggio a isola alimentati unicamente da pannelli solari. Prelevano i dati e li trasmettono al cloud tramite un sistema di telecomunicazioni. Sono suddivisi in tre tipologie (A, B e C). Le tipologie A e B sono costituite da un pannello solare che alimenta un pacco batterie, sensori e stazione meteo, mentre la terza non possiede i pannelli solari e le batterie, ma non possiede né i sensori né la stazione meteo. Per contro i tipi B e C hanno pannelli fotovoltaici e batterie maggiori del tipo A.

A seguire vediamo uno schema a blocchi di un dispositivo di tipo A:

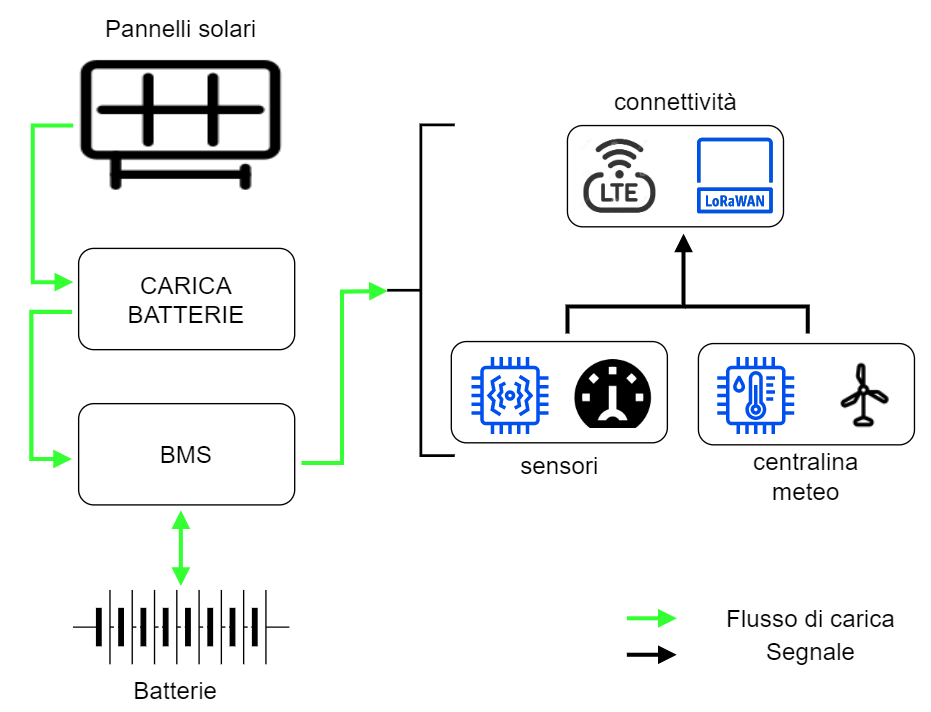


Figura 3‑4: Schema a blocchi di un dispositivo di tipo A

In Figura 3‑4, vediamo i vari componenti del dispositivo. Per l’alimentazione del sistema abbiamo un pannello solare collegato ad un caricatore. Tra il caricatore e il pacco batterie a litio abbiamo un BMS (Battery Management System) incaricato di ottimizzare la gestione e il tempo di vita delle celle (11). Questo sistema di produzione e stoccaggio dell’energia è incaricato di fornire la carica necessaria al corretto funzionamento dei sensori, della centralina meteo e della connettività.

Come si evince in Tabella 3‑1, i dispositivi di tipo A sono dotati di un pacco batterie contenente 4 celle, mentre i dispositivi di tipo B e C possiedono 8 celle. Il BMS è il sistema che ne gestisce la carica e la scarica in modo da garantirne il corretto funzionamento e la miglior durata di vita. Inoltre, calcola lo stato di carico (SOC), ossia la percentuale della piena capacità di carica del pacco batterie ancora disponibile (11). È il BMS a decidere se e quando caricare o scaricare la batteria. Nel nostro caso, la decisione viene presa prendendo in considerazione la potenza fornita dal pannello solare e il SOC della batteria: Se il SOC non è al 100% e il pannello sta fornendo una potenza maggiore rispetto al consumo dal sistema, il BMS passa direttamente al sistema la carica necessaria e stocca il surplus nel pacco batterie. Nel caso in cui il pannello non stia producendo abbastanza potenza per alimentare da solo il sistema, la potenza necessaria viene prelevata dal pacco batterie. Inoltre, qualora il pannello stia producendo carica in eccesso ma il SOC stia già al 100%, il pacco batterie non viene ricaricato dal BMS.

Come si evince dalla Figura 3‑4, il consumo del dispositivo è dovuto ai sensori, la centralina meteo e alla connettività. Le considerazioni fatte sui dispositivi di tipo A si applicano ai dispositivi di tipo B, e anche a quelli di tipo C se non si considerano i sensori e la centralina meteo.

A seguire vediamo una foto di un dispositivo di tipo B con tutti i suoi componenti:

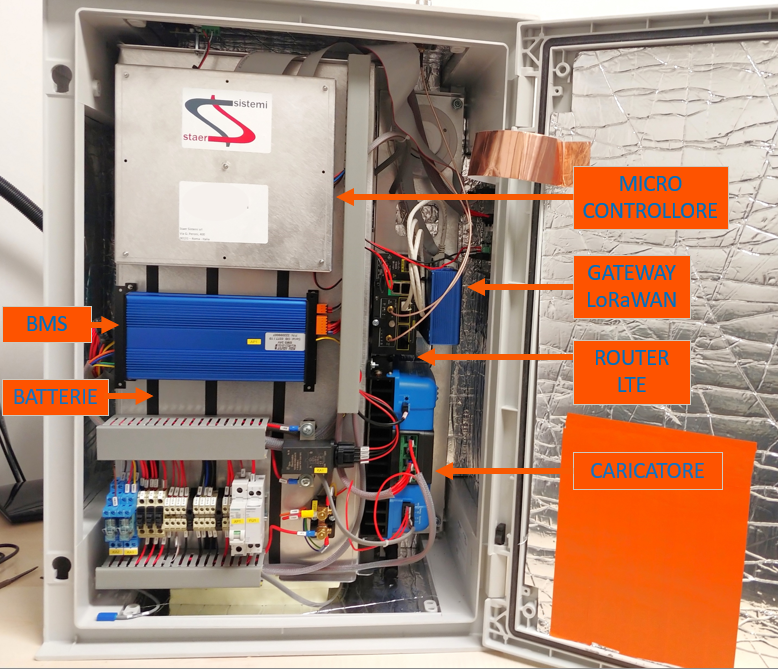


Figura 3‑5: Implementazione hardware di un dispositivo di tipo B

In Figura 3‑5 vediamo un’implementazione reale dello schema a blocchi presentato in Figura 3‑4. Dato che i dispositivi saranno collocati in ambienti difficilmente raggiungibili e che saranno esposti alle intemperie, vengono posizionati in un armadietto protettivo.

A seguire vediamo una tabella riepilogativa delle caratteristiche dei dispositivi:

Tabella 3‑1: Caratteristiche hardware degli apparati

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | A | B | C |
| BATTERIE  (Litio) | Tensione nominale | 12.8 V | 25.6 V | 25.6 V |
| Carica elettrica | 100 Ah | 160 Ah | 160 Ah |
| Energia immagazzinabile | 1280 Wh | 4096 Wh | 4096 Wh |
| Celle per unità | 4 | 8 | 8 |
| PANNELLI SOLARI | Potenza nominale | 115 W | 400 W | 400 W |
| Dimensioni | 1015 x 668 x 30 mm | 1690 x 1046 x 40 mm | 1690 x 1046 x 40 mm |
| SENSORI | | SI | SI | NO |
| CENTRALINA METEO | | SI | SI | NO |
| PROTOCOLLI DI COMUNICAZIONE | | LORAWAN e LTE | LTE | LTE |

Dalla Tabella 3‑1 possiamo osservare come i dispositivi A abbiano due protocolli di comunicazione: il primo è il protocollo LoRaWAN (Long Range Wide Area Network), un protocollo di trasmissione dei dati ad ampio spettro e a lunga portata che si basa sulla tecnologia LoRa. Quest’ultima è una tecnica di modulazione che deriva dalla tecnologia Chirp Spread Spectrum, utilizzata fin dal secondo dopoguerra nel campo militare. L’architettura include standard di protocollo e funzionalità che supportano la comunicazione bidirezionale a basso costo, mobile e sicura ed ottima per dispositivi IoT. L’architettura LoRaWAN è ottimizzata per un basso consumo energetico ed è progettata per scalare da un’installazione di un singolo gateway, fino a grandi reti globali con miliardi di dispositivi (9). La distanza massima che può essere coperta da questa connessione è di 15 km, nel nostro caso i dispositivi si trovano a non più di 5 km di distanza. Il protocollo LoRaWAN è usato di default dai dispositivi A per ridurre i consumi poiché hanno pacchi batterie più piccoli rispetto agli altri dispositivi. Il secondo è il protocollo LTE (Long Term Evolution), che viene usato come soluzione provvisoria qualora si dovesse verificare un problema temporaneo sulla rete LoRaWAN. Quest’ultimo è uno standard di connessione che permette l'accesso a internet a banda larga anche su reti mobili e ha prestazioni migliori di LoRaWAN a fronte di un consumo energetico maggiore (10).

I dispositivi B e C invece, fanno uso solamente della rete LTE poiché possiedono pacchi batterie e pannelli solari di dimensioni maggiori. In particolare, come vediamo in Tabella 3‑1, le batterie e i pannelli solari montati sui dispositivi B e C sono identici. Notiamo inoltre, che i dispositivi C non possiedono né stazione meteo né sensori poiché assolvono solo il compito del ricevitore per i segnali LORAWAN dei dispositivi A nelle vicinanze.

Una considerazione importante è che tutti i dispositivi (A, B e C) hanno un consumo approssimativamente costante causato sia dalla trasmissione dei dati sia, per i dispositivi A e B, dalla centralina meteo e dai sensori. Questo ci permette di valutare a priori il consumo energetico del dispositivo. Un’eccezione ala regola, si verifica nel caso in cui in cui un dispositivo di tipo A dovesse passare dalla connessione via LoRaWAN alla connessione via LTE.

Vediamo ora una rappresentazione della rete che collega i dispositivi in campo al database in cui vengono salvati i dati che in seguito andremmo ad analizzare:



Figura 3‑6: Connessione bidirezionale tra i dispositivi in campo e il databse

I dispositivi di tipo A sono rappresentati dai triangoli mentre i riquadri contenenti i gateway LoRaWAN e i router LTE sono i dispositivi di tipo B e/o C. Dalle frecce bidirezionali si evince che la comunicazione tra i dispositivi è bidirezionale, ossia in un senso vengono mandati al database i dati dal campo, e nell’altro vengono mandati ai dispositivi in campo i segnali di controllo remoto. Vediamo che i dispositivi di tipo A si connettono tramite la rete LoRaWAN ai dispositivi di tipo B o C nei dintorni e mandano i dati dal campo. I dati raccolti dai gateway LoRaWAN vengono poi passati al router LTE. I dati passati al router LTE vengono poi mandati al cloud in cui viene fatta una cernita dei dati da salvare, e poi vengono scritti nel database SQL Dallo schema, notiamo che i dispositivi di tipo A possono collegarsi a diversi gateway nelle vicinanze, in questo modo ci si accerta che, anche se un gateway non è disponibile, il segnale può comunque essere trasmesso usandone un altro. È da notare, che anche i dispositivi di tipo B e C mandano dati al database, chiaramente in questo caso le misure dal campo vengono passate direttamente al router LTE, spediti al cloud e poi salvati nel database, senza passare dalla connessione LoRaWAN.

### Salvataggio dei campioni nel database

Per semplicità di ragionamento, facendo riferimento a Figura 3‑4, andiamo a suddividere le misurazioni in 2 pacchetti/categorie: il pacchetto energy management contenente tutti i dati provenienti dal caricatore, pannello solare, BMS e pacco batterie; il pacchetto sensoristica contenente i dati della centralina meteo e dei sensori.

La frequenza di campionamento è impostata a 15 minuti. Dato l’alto numero di sensori e dato che il protocollo LoRaWAN è a banda stretta, i dati non vengono trasmessi tutti insieme, ma bensì sfalsati. Tutti i dati provenienti dal pacchetto energy management vengono mandati contemporaneamente, mentre sfalsati di 7,5 minuti, vengono mandati quelli del pacchetto sensoristica e della centralina metereologica. Inoltre, Al fine di minimizzare l’occupazione su disco del database, i valori provenienti dal campo vengono storicizzati solo su variazione rispetto i precedenti. Tutte le misurazioni vengono inviate ogni 15 minuti dai dispositivi, e solo in seguito vengono filtrate dal cloud in modo che i campioni invariati rispetto al precedente non vengano scritti nel database.

Nello specifico, qualora il dato non venga ricevuto dal cloud entro un intervallo massimo di 20 minuti (5 minuti dopo l’arrivo previsto del dato), viene effettuata una nuova scrittura nel database con bit di diagnostica alto, in modo da storicizzare la mancata comunicazione. Questa condizione riflette la mancata ricezione del campione, ma non ne determina la causa. La variabile di diagnostica viene disattivata quando si riceve un nuovo campione. Nella maggior parte dei casi ciò accade per un problema di connessione. Qualora il problema di connessione non si risolva nel breve termine, una procedura automatizzata viene implementata in base alla tipologia di apparato. I dispositivi di tipo A in questi casi funzionano con il router finché non torna la connessione LoRaWAN, mentre i dispositivi di tipo B e C riavviano il router LTE.

In conclusione, nel dataset la scrittura del campione non avviene in modo regolare dato che essa dipende dalla sua variazione. Per esempio, la potenza del pannello viene campionata solamente nelle ore di luce poiché la notte la sua potenza è sempre nulla. Per questo motivo, durante la notte non sono presenti campioni riguardanti la potenza del pannello solare. Inoltre, le misurazioni vengono trasmesse in due trance diverse dato che la banda del protocollo LoRaWAN è stretta.

# Elaborazione dei dati

## Presentazione dei dati per costruire il dataset

I dati vengono acquisiti dal database (SQL Server) tramite Database Toolbox, il plugin di Matlab che permette di effettuare delle query direttamente dall’editor. Questo perché altrimenti per esportare i dati dal database dovremmo, per ogni query, passare per l’interfaccia di SQL server, esportare i dati in csv e poi caricarli manualmente dentro Matlab. Il plugin invece, permette di scrivere direttamente da Matlab la query come da SQL server e poi la converte in codice Matlab. Una volta lanciata la routine i dati vengono salvati direttamente nel workspace.

È stata fatta la scelta di progetto di prelevare per ogni dispositivo i dati partendo dal 01-05-2021fino alla data di spegnimento. Questa scelta è stata fatta per coprire sia la stagione estiva, in cui vi sono più ore di luce, e quindi ci sono meno spegnimenti, sia la stagione invernale in cui si verificano maggiormente i problemi dovuti a scarso irraggiamento. La query ritorna i dati sotto forma di una tabella contenenti 4 colonne, il tempo di ricezione, valore della variabile, il codice identificativo della variabile ed il bit di diagnostica.

Le variabili prelevate sono le seguenti:

* **Irraggiamento**: L’irraggiamento istantanea acquisita dalla stazione meteo ogni 15 minuti.
* **Tensione della cella minima**: Estrazione del valore di tensione della cella componente il pacco con tensione più bassa.
* **SOC (State of Charge)**: Una stima effettuata dal BMS in Watt/ora (Wh) del livello di carica dell’insieme del pacco batterie. Presenta un errore cumulativo sulla stima che si azzera quando la batteria raggiuge lo 0% oppure il 100%.
* **Battery current in, Battery current out:** Le correnti istantanee in ingresso e in uscita alla batteria. Le useremo per calcolare il bilancio delle correnti della batteria.
* **Potenza del pannello:** Potenza istantanea del pannello solare.

Inoltre, abbiamo prelevato il bit relativo alla diagnostica associato ad ogni campione. Esso viene posto alto dal ricevitore quando il campione atteso non viene ricevuto entro il tempo massimo consentito. Viene rimesso basso quando si riceve un nuovo campione. Permette la distinzione tra una mancanza di inserzione nel database dovuta alla non variazione della grandezza, da una situazione in cui la variabile non è pervenuta dal campo entro il tempo massimo consentito.

## Preelaborazione dei dati (Sincronizzazione)

Per prima cosa andiamo a creare una struttura formata da 4 celle: nella prima abbiamo il time-stamp, nella seconda il valore della variabile, nella terza il codice identificativo della variabile e nella quarta il bit di diagnostica. Il tempo viene convertito in formato Excel, un formato che esprime il tempo in numero di giorni a partire dal 01/01/1900. Tramite una tabella identificativa facciamo la corrispondenza tra i codici identificativi delle variabili ed il loro nome.

I dati appartenenti allo stesso pacchetto vengono campionati allo stesso istante di tempo. Questo comporta che diverse colonne consecutive presentano lo stesso istante di tempo. Per ovviare a questo problema andiamo a creare una struttura che contenga una cella per ogni variabile di modo da rendere gli istanti di tempo univoci (vedi codice in appendice Figura 8‑1).

Una volta creata questa struttura dati, dobbiamo andare ad elaborarlo per correggerne le problematiche. In particolare, un problema si pone quando abbiamo dei problemi in ricezione. Come detto in precedenza, in questi casi viene alzato un bit di diagnostica. Nell’intervallo in cui il bit rimane alto, la nostra scelta di progetto è di interpolare linearmente tra il campione precedente e quello successivo l’entrata in diagnostica.

Vediamo un esempio d’interpolazione:

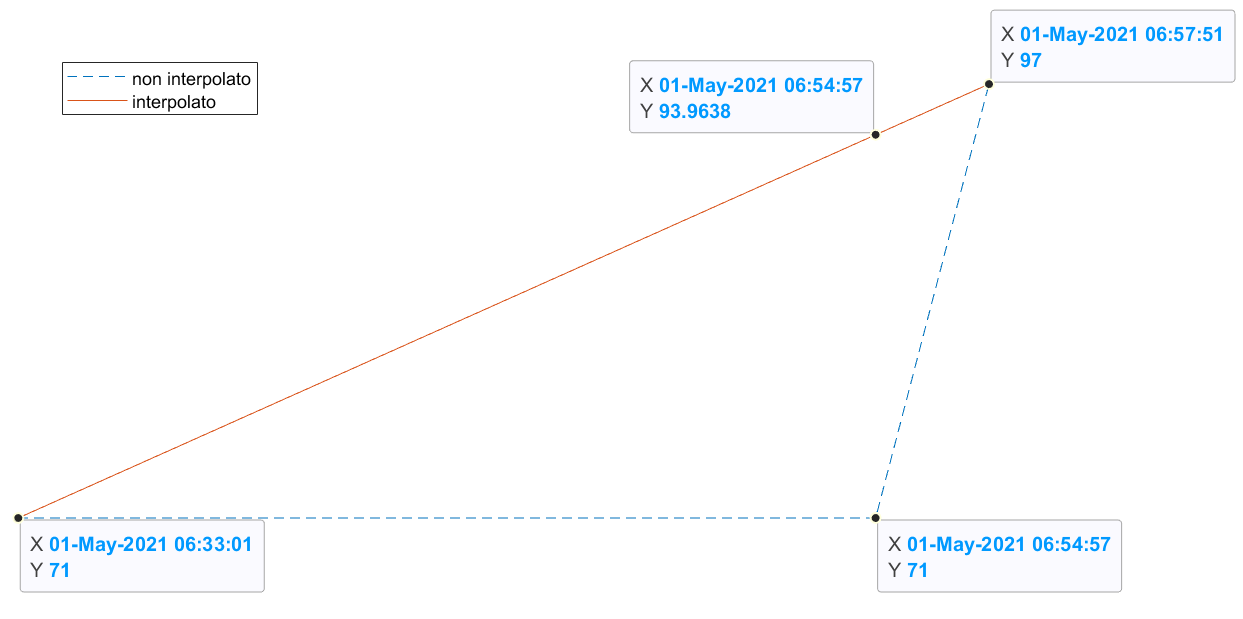


Figura 4‑1: Interpolazione

Come vediamo in Figura 4‑1, il campione interpolato è quello intermedio in data “01-05-2021 06:54:57”. La retta blu tratteggiata rappresenta i campioni non interpolati e quella rossa rappresenta i campioni interpolati. Andando a tracciare una retta tra i due campioni validi ed ignorando il campione in diagnostica, si vuole approssimare al meglio l’andamento dei dati nel periodo in cui non si hanno dati. Questo tipo di approssimazione è tanto più importante perché altrimenti il sistema interpreta che, tra il primo e il secondo punto in Figura 4‑1, l’andamento della grandezza è costante e, che tra il secondo ed il terzo punto subisce uno sbalzo improvviso. Abbiamo scelto l’interpolazione lineare, piuttosto che altri modelli più complessi, per rimanere il più fedeli possibile ai dati a nostra disposizione assicurandoci che la retta passi esattamente per i punti rappresentati dai campioni reali. Non abbiamo inserito effetti di addolcimento della curva, dato che questi effetti introducono, in certi casi, valori fisicamente impossibili, come valori di potenza inferiori a zero. Purtroppo, non introducendo questi effetti, vediamo la presenza di gradini o cuspidi che non rappresentano realisticamente l’evoluzione naturale dell’evento (vedi Figura 8‑2 in appendice per il codice dell’interpolazione).

Andiamo a vedere un esempio di problema generato dall’errata interpretazione dei campioni di diagnostica. Quando il ricevitore non riceve dati, in alcuni casi particolari, scrive i valori delle variabili come nulli come si vede osservando la retta blu tratteggiata tra i due campioni evidenziati in Figura 4‑2. In questi casi si verificavano picchi che falsano completamente l’andamento della grandezza. Ciò si verifica per esempio per le variabili relative alla batteria, come la tensione della cella minima.

In Figura 4‑2 la grandezza in questione sembra andare a zero nel lasso di tempo relativo alla diagnostica:

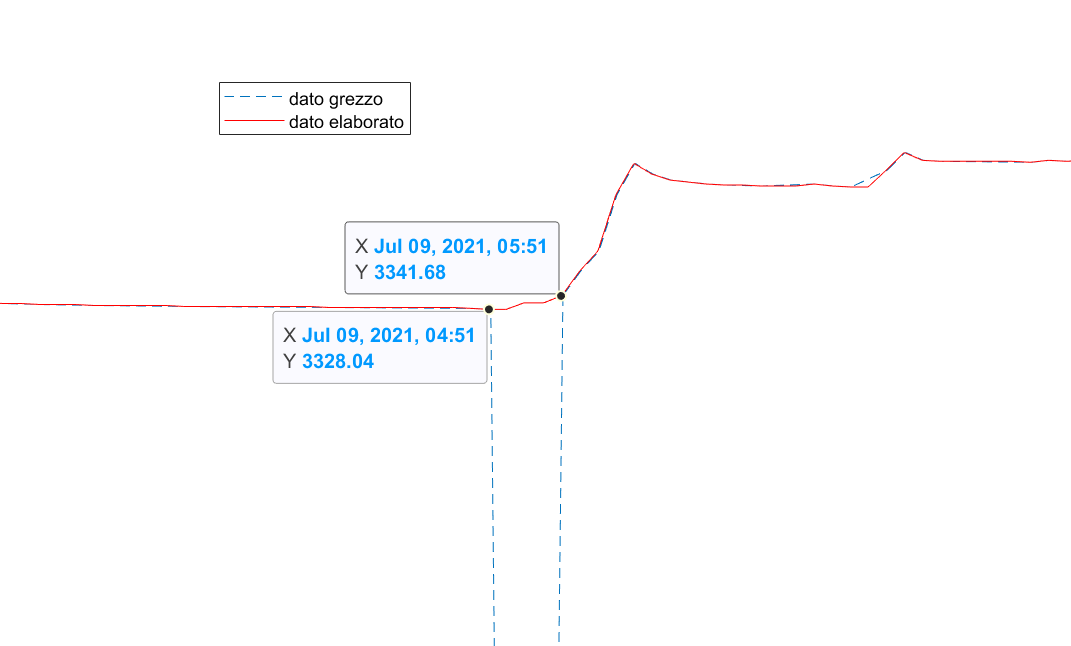


Figura 4‑2: Correzione del valore nullo in diagnostica

Osservando la curva rossa in Figura 4‑2 vediamo che tramite la procedura di interpolazione andiamo a correggere tali andamenti.

Una volta interpolati i valori nei casi in cui viene alzato il bit di diagnostica, possiamo considerare tutti i dati presenti come correttamente ricevuti e quindi validi. A questo punto, qualora l’intervallo di tempo tra due campioni consecutivi sia maggiore di 20 minuti consideriamo che la mancanza del dato è dovuta al fatto che esso è rimasto invariato rispetto al campione precedente. Per sopperire all’assenza del campione andiamo ad inserire il valore invariato, 15 minuti prima del campione successivo ricevuto.

A seguire, in Figura 4‑3 vediamo una dimostrazione del suddetto concetto:

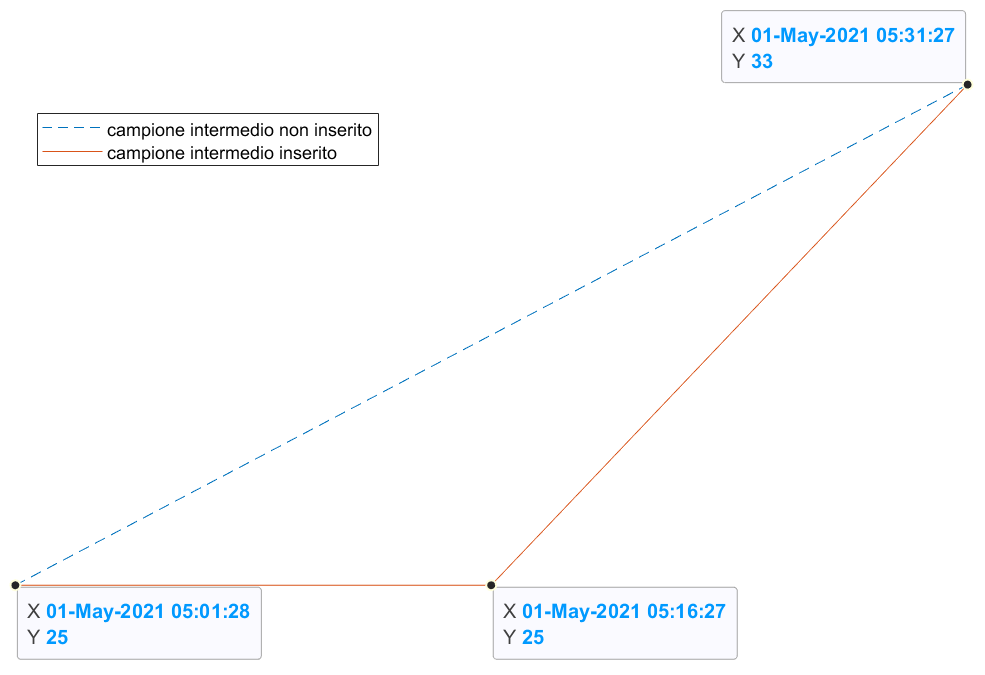


Figura 4‑3: Traslazione del dato

In Figura 4‑3 vediamo la retta blu tratteggiata che rappresenta i campioni senza l’aggiunta del campione mancante mentre quella rossa rappresenta i campioni dopo l’aggiunta del campione mancante. In questo caso l’intervallo tra i due punti “01-05-2021 05:01:28” e “01-05-2021 05:31:27” è maggiore di 20 minuti (retta blu tratteggiata), e quindi vuol dire che il campione in data “01-05-2021 05:16:27” non è stato trascritto perché invariato rispetto al precedente. Al fine di sincronizzare il dataset andiamo ad inserire questo campione intermedio di valore uguale a quello in data “01-05-2021 05:01:28” (vedi appendice in Figura 8‑3 per il codice utilizzato).

Un esempio dell’importanza della procedura mostrata in Figura 4‑3 è mostrato in Figura 4‑4. Andando a riscrivere il dato quando non è presente andiamo a risolvere un problema che occorreva nel salvataggio dei dati durante il periodo notturno. Infatti, alcune variabili come la potenza del pannello solare hanno valori diversi da zero solo durante il giorno, mentre la notte rimangono costantemente a zero. Possiamo osservare questo fenomeno in Figura 4‑4:

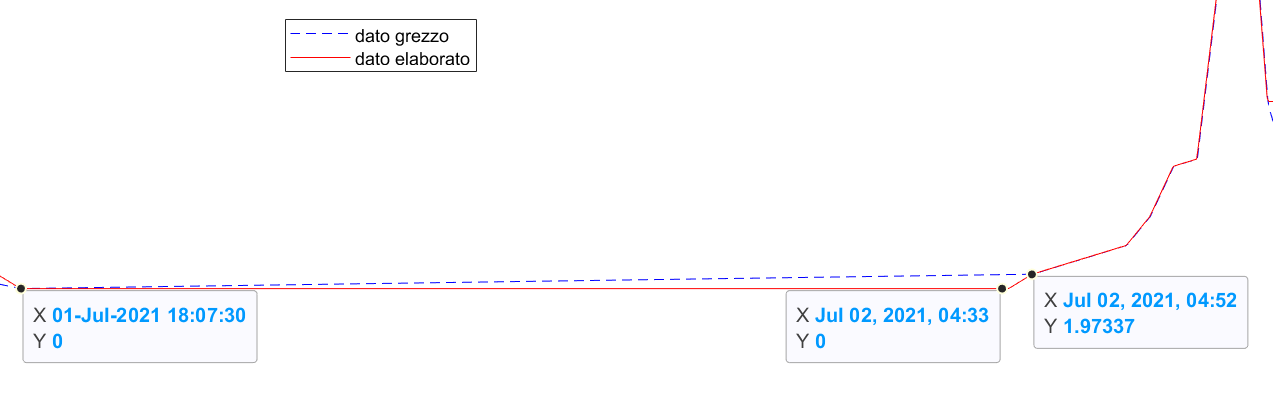


Figura 4‑4: Problema della non trascrizione del dato di notte per la corretta rappresentazione della potenza del pannello

Come si vede in Figura 4‑4 può accadere che l’ultimo campione in cui il pannello è attivo (“01-Jul-2021 18:07:30”) abbia valore nullo, mentre il primo valore campionato la mattina seguente sia già diverso da zero. Per questo motivo, non avendo campioni intermedi, si registra una leggera crescita di potenza durante la notte (retta blu tratteggiata). Ciò è stato corretto andando a riscrivere il campione con la procedura di traslazione (retta rossa).

Dopo aver fatto questa procedura di traslazione del dato, andiamo a fare un sovra campionamento alla frequenza del secondo di modo da riempire il dataset sovrabbondantemente, per poi estrarre un campione al minuto. In questo modo otterremo un dataset sincrono con 1 campione al minuto (vedi codice in appendice in Figura 8‑4).

Alla fine di questo procedimento troveremo, per gli stessi dati che in Figura 4‑3, la seguente distribuzione:

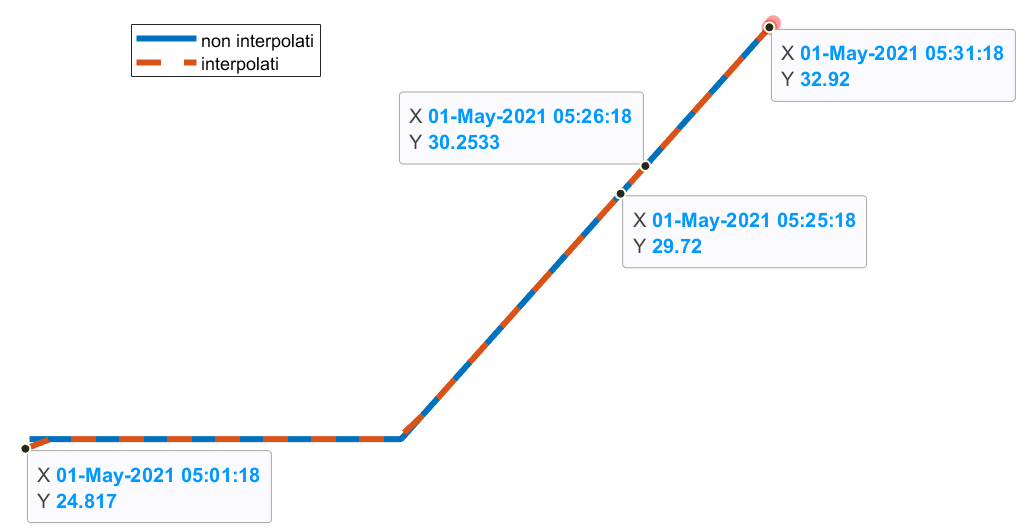


Figura 4‑5: Sovra campionamento al minuto

Possiamo osservare che ora abbiamo un campione al minuto. In questo modo il dato risulta sincrono e con una frequenza di campionamento 15 volte maggiore rispetto a quella dei dati provenienti dal campo.

Gli step descritti in Figura 4‑1, Figura 4‑3 e Figura 4‑5 vanno ripetuti per tutte le variabili prelevate. Una volta fatta quest’operazione abbiamo una struttura dati per ogni variabile, ognuna con un proprio time-stamp. Tuttavia, per poter creare un dataset, dobbiamo avere tutte le variabili sotto un unico time-stamp. Per fare ciò, andiamo a trovare, per tutte le variabili, l’istante corrispondente al primo e all’ultimo campione. Confrontando gli istanti di inizio e di fine ottenuti per ogni variabile, scegliamo il tempo di inizio massimo e quello di fine minimo, andando a definire così quale sarà l’intervallo su cui valuteremo i dati (vedi appendice in Figura 8‑5 per il codice).

In questo modo ora possiamo creare una struttura dati con un unico time-stamp e un campo per ogni variabile. Con questa struttura dati possiamo andare a creare un dataset sincronizzato da dare in ingresso alla rete neurale (vedi appendice per struttura dati in Figura 8‑6).

## Creazione di sequenze della durata di 3 giorni

In questo paragrafo andiamo a descrivere i parametri fondamentali nella costruzione delle sequenze che formeranno il dataset. Al fine di rendere il codice agile, questi parametri vengono definiti all’inizio del codice. Essi non vengono mai scritti successivamente, vengono solamente letti nelle varie routine (vedi appendice in Figura 8‑7 per il codice relativo).

I parametri sono:

* La lunghezza delle sequenze espressa in giorni. In questo caso è di 3 giorni
* Lo sfalsamento tra due sequenze consecutive in giorni. Al fine di avere il numero necessario di sequenze nel dataset ne abbiamo implementato una parziale sovrapposizione. Nella fattispecie, se una sequenza comprende l’intervallo tra l’1 e il 3 maggio, quella consecutiva comprende l’intervallo tra il 2 e il 5 maggio. Lo sfalsamento tra due sequenze consecutive è dunque di 1 giorno.
* L’anticipo di previsione che si vuole impostare in giorni. Nel nostro caso può avere 3 valori a seconda del caso: 1, 3 o 7 giorni.
* Il rapporto tra sequenze sane e patologiche nel dataset. Nel nostro caso avremmo una proporzione di tre sequenze sane per una sequenza patologica.
* La soglia critica per discriminare tra sequenze patologiche e sequenze sane. Nella fattispecie, si riferisce al valore della tensione della cella minima. Qualora un solo campione di una sequenza si trovi al di sotto della tensione nominale della cella (3.2 V) essa viene considerata come anomala.
* La proporzione delle sequenze ripartite tra test e train in percentuale. Si applica solo nel caso di partizione statica ed è di ¼ per il test set e ¾ per il traning set.

Maggiori approfondimenti sull’uso di tali variabili verranno fornite nelle sezioni seguenti.

Andiamo ora a vedere com’è strutturato il codice per la creazione del dataset. La procedura va ripetuta per ogni dispositivo:

* Scegliamo le variabili che vogliamo inserire nelle sequenze.
* Preleviamo le sequenze eleggibili per far parte del dataset evitando quelle contenenti intervalli temporali di diverse ore in cui non sono arrivati dati dal campo.
* Discriminiamo tra sequenze sane e patologiche. Le sequenze etichettate in questo modo verranno chiamate d’ora in poi sequenze di riferimento.
* Per prevedere l’evento di guasto dobbiamo spostare le etichette delle sequenze tot giorni prima dell’evento. Con questo intendiamo che se per esempio abbiamo un evento di spegnimento l’8 dicembre (evento patologico), e vogliamo prevedere i guasti 7 giorni prima, allora l’etichetta patologica relativa al guasto dell’8 dicembre va applicata alle sequenze contenenti il 1° dicembre. In questo modo l’algoritmo impara a riconoscere le condizioni dell’apparato, 7 giorni prima del guasto, come anomale. Chiaramente, questa procedura si applica a tutte le sequenze, sia sane che patologiche. Per semplicità chiameremo le sequenze etichettate in questo modo sequenze predittive.
* Normalizziamo i dati.
* Dividiamo le sequenze tra Train e Test. Questa operazione viene fatta solo nel caso si voglia fare una partizione statica del dataset.
* Salviamo i dati in una cartella relativa al dispositivo in questione

I dati salvati facendo questa procedura sono salvati in cartelle distinte per ogni dispositivo. Andiamo ora ad unificare questi dati salvati in un unico dataset finale che in seguito andremmo a passare all’algoritmo di classificazione (vedi appendice in Figura 8‑8 per il codice).

### Estrazione delle sequenze

Andiamo a suddividere in sequenze da poter dare in ingresso all’algoritmo di classificazione il dato ottenuto in 4.2. In questo esempio andiamo a ripartire il dato in sequenze di 3 giorni sfalsate di 1 giorno rispetto alla precedente. Abbiamo scelto di avere una sovrapposizione parziale delle sequenze per la scarsa casistica di guasto. In questo modo possiamo avere una maggiore consistenza del dataset.

Per prima cosa dobbiamo trovare un modo di isolare i singoli giorni per poi assemblare le sequenze di 3 giorni. Per fare ciò, usiamo una funzione il cui compito è di trovare ogni giorno la prima occorrenza di un campione nell’intervallo tra le 20.00 e le 05.00. Questa operazione la facciamo con i dati non sincronizzati e quindi aventi ancora assenza di dati laddove non sono stati ricevuti. In questo modo, qualora per un dato giorno non si siano ricevuti dati tra le 20.00 e le 05.00 esso viene escluso dai giorni selezionabili per creare delle sequenze.

A seguire un esempio delle sequenze che vogliamo evitare di selezionare:

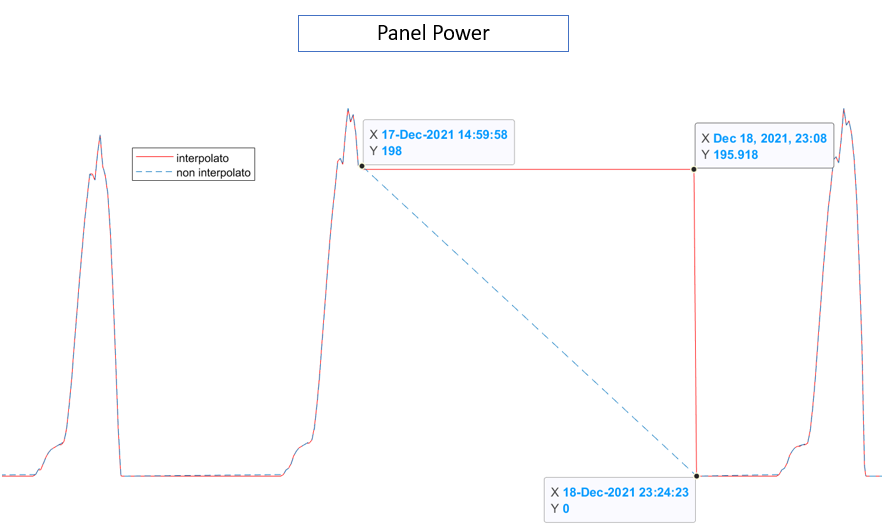


Figura 4‑6: Differenza di andamenti tra dato sovra campionato e non, là dove si verifica una mancata ricezione del dato

In Figura 4‑6 abbiamo un intervallo di diversi giorni. La grandezza osservata è la potenza del pannello solare. Vediamo che la curva rossa (grandezza interpolata), all’altezza del secondo picco, fa un gradino. Questo gradino è il frutto dell’interpolazione fatta in 4.2, e non è realistico dato che implicherebbe che nell’intervallo di tempo tra il “17-Dec-2021 14:59:58” e il “18-Dec-2021 23:24:23” la potenza del pannello sia rimasta costante. In questo caso, l’interpolazione lineare è per forza di cose imprecisa dato che non sono stati ricevuti dati per un intervallo lungo (approssimativamente 32 ore). Per rilevare automaticamente queste situazioni, e quindi evitare di selezionare il 17 e il 18 dicembre nelle sequenze, usiamo i dati grezzi provenienti dal campo (curva blu tratteggiata) poiché essi non contengono campioni in quell’intervallo, a differenza dei dati interpolati (vedi appendice in Figura 8‑9 per il codice relativo).

Una volta isolati i singoli giorni, quello che abbiamo ottenuto sono sequenze da 24 ore. Andiamo a concatenarle per creare sequenze con una durata di 72 ore sfalsandole ognuna di 24 ore rispetto alla precedente. Per fare ciò compariamo i time-stamp contenuti nelle sequenze (vedi appendice in Figura 8‑10 per il codice relativo).

In Figura 4‑7 osserviamo l’effetto del processo di selezione delle sequenze valide rispetto alla totalità del dato (in questo caso vediamo l’andamento della tensione della cella minima):

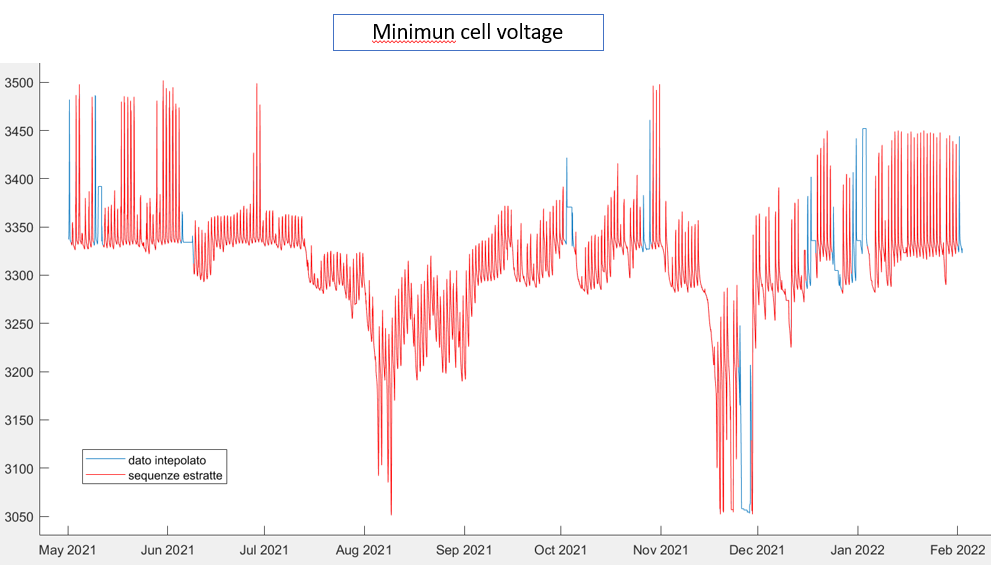


Figura 4‑7: Paragone tra le sequenze estratte e la totalità del dato sincronizzato

In Figura 4‑7 la curva rossa rappresenta le sequenze estratte per costituire il dataset, mentre in blu si vedono le parti del dato sincronizzato che non sono state selezionate perché contenenti buchi di dati e quindi approssimazione imprecise come nel caso di Figura 4‑6. Se andiamo a zoomare su una porzione, vediamo che le sequenze in blu (quelle che non faranno parte del dataset) presentano lunghi intervalli in cui il dato rimane costante, sintomo del fatto che non è stato ricevuto il dato per un lungo periodo di tempo e quindi è giusto escluderle.

Un altro caso simile a quello in Figura 4‑6 si può osservare in Figura 4‑8:

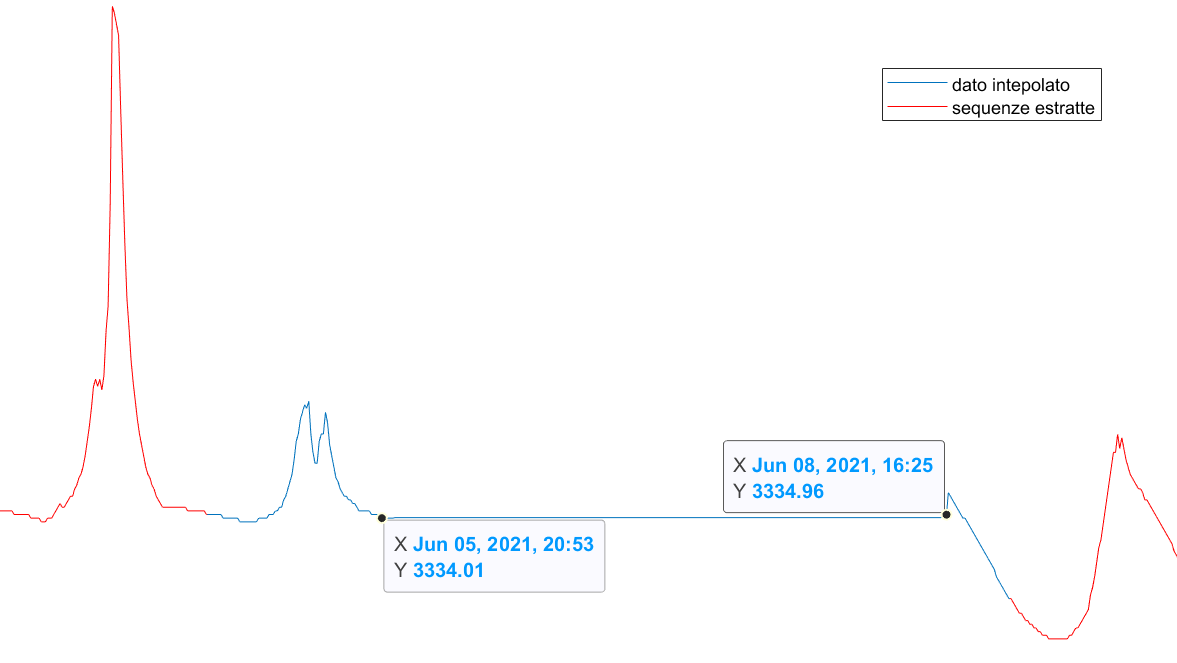


Figura 4‑8: Lasso temporale non valido ai fini del dataset

Vediamo che tra il “05-Jun-2021 20:53” e il “08-Jun-2021 16:25” abbiamo fatto un’interpolazione ma non ci sono stati dati ricevuti e la retta rimane piatta per quasi 3 giorni. Per questo motivo quest’intervallo viene escluso dal dataset.

Grazie alla sincronizzazione dei dati che abbiamo fatto in 4.2, ogni sequenza ha lo tesso numero di campioni.

### Identificazione delle sequenze sane e patologiche

Andiamo a differenziare le sequenze sane da quelle patologiche. Una sequenza viene detta patologica quando almeno un campione della tensione della cella minima si trova sotto il valore nominale della batteria a 3,2 Volt, mentre il resto delle sequenze viene considerato sano (vedi appendice in Figura 8‑11 per il codice).

Dopo aver etichettato le sequenze con il suddetto parametro andiamo a spostare le etichette come accennato nell’introduzione del capitolo 4.3. Per fare ciò usiamo il timestamp delle sequenze, e spostiamo le etichette delle sequenze di riferimento indietro del numero di giorni a cui vogliamo prevedere l’evento. In questo modo otteniamo le sequenze predittive. La stessa procedura vale per tutte le sequenze, sia sane che patologiche (vedi appendice in Figura 8‑12 per il codice).

Poiché vogliamo rispettare una proporzione tra il numero di sequenze sane e di sequenze patologiche, e tipicamente abbiamo un numero di sequenze sane molto maggiore di quelle patologiche, andiamo a sfoltire il numero di sequenze sane estraendone alcune casualmente. In questo modo le sequenze sane selezionate cambiano ogni volta che eseguiamo la routine (vedi appendice in Figura 8‑13 per il codice). Una volta eseguite queste operazioni otteniamo una cernita di sequenze classificate tra sane e patologiche.

A seguire un esempio di sequenza patologica di riferimento e la relativa sequenza predittiva 7 giorni prima:

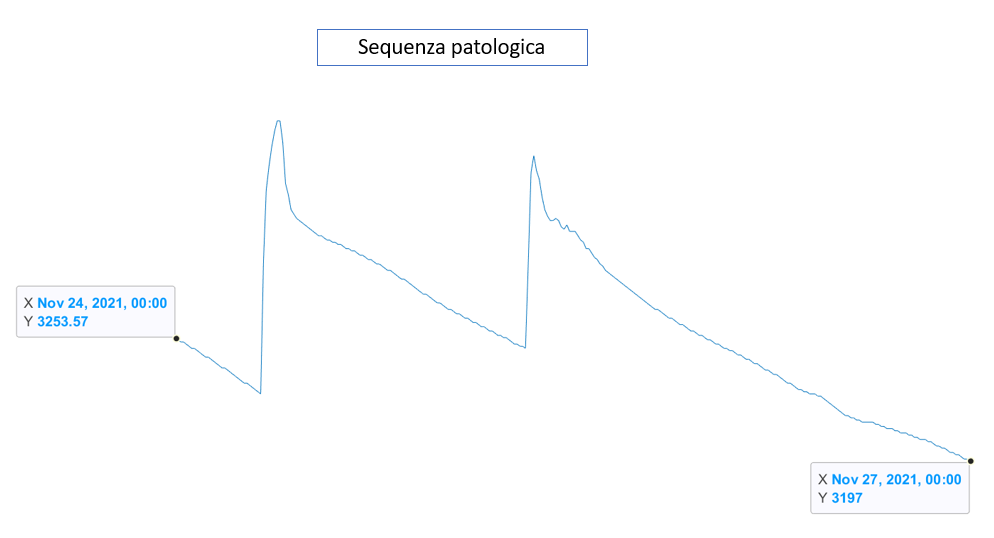


Figura 4‑9: Sequenza patologicac di riferimento



Figura 4‑10: Sequenza predittiva

Vediamo che in Figura 4‑9 la sequenza patologica di riferimento è considerata tale perché nell’ultima parte scende al di sotto d 3200 [mV]. La sequenza predittiva viene selezionata calcolando che l’ultimo campione della sequenza predittiva deve essere 7 giorni precisi prima dell’ultimo campione della sequenza patologica, come vediamo dai marker.

A seguire vediamo come viene prelevata la sequenza predittiva:

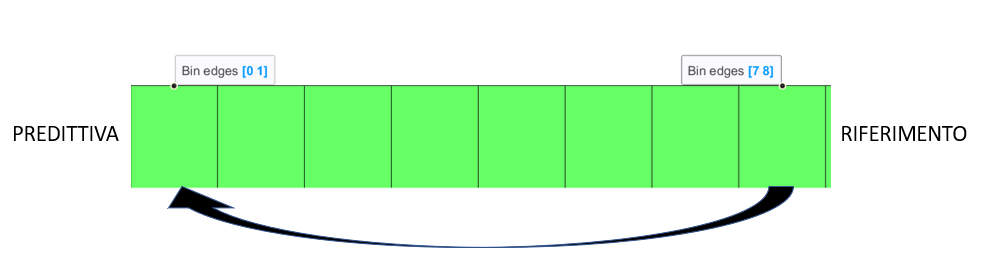


Figura 4‑11: Selezione della sequenza predittiva con predzione a 7giorni e sequenza di riferimento sana

In questo caso la previsione è a 7 giorni, quindi la sequenza predittiva è quella 7 sequenze prima. La sequenza predittiva viene selezionata allo stesso modo, sia che si riferisca ad una sequenza sana che patologica.

Vediamo il caso in cui la sequenza di riferimento sia patologica:

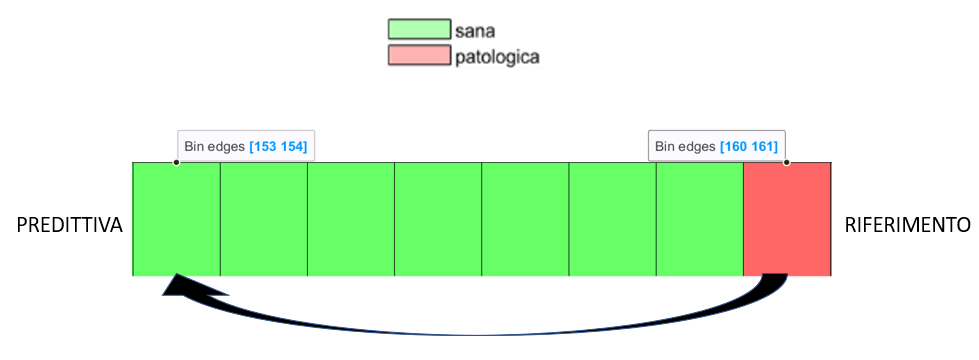


Figura 4‑12:Selezione della sequenza predittiva con predzione a 7giorni e sequenza di riferimento patologica

Il meccanismo di selezione è lo stesso tra Figura 4‑11 e Figura 4‑12. Da notare che la sequenza predittiva acquisisce l’etichetta (sana o patologica) della sequenza di riferimento. Per questo motivo la sequenza predittiva in Figura 4‑12 verrà considerata patologica ai fini del dataset per la classificazione predittiva a 7 giorni. Ciò è possibile perché sappiamo per certo che dopo 7 giorni si verificherà un evento patologico.

Nel seguente grafico possiamo vedere le sequenze patologiche di riferimento (rosso) rispetto alle sequenze patologiche predittive (arancioni):

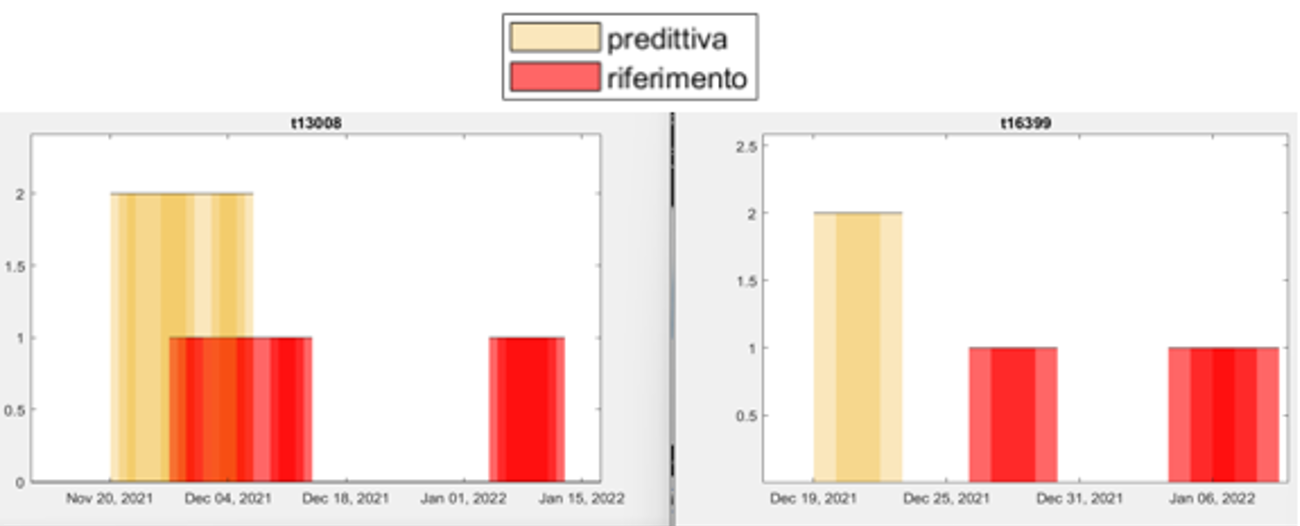


Figura 4‑13: Disposizione tra sequenze di riferimento e predittive a 7 giorni per ogni dispositivo

In Figura 4‑13 notiamo come le sequenze predittive siano sfalsate a sinistra rispetto a quelle di riferimento, questo perché sono predittive. In teoria dovrebbero essere in numero uguali a quelle di riferimento, in pratica però si può verificare il caso in cui la sequenza predittiva cada in un intervallo dove abbiamo una mancanza di dati e quindi non venga generata nessuna sequenza predittiva per una data sequenza di riferimento. Ciò si verifica in entrambi i grafici in Figura 4‑13 in cui rispettivamente le sequenze patologiche nell’intorno del 6 gennaio e nell’intervallo tra il 1° e il 15 gennaio ci sono sequenze patologiche di riferimento ma non predittive.

Inoltre, possiamo notare come in alcuni casi le sequenze predittive e di riferimento si sovrappongano tra loro. Ciò si verifica per il grafico a sinistra perché le sequenze di riferimento sono spalmate in un intervallo maggiore di 7 giorni e quindi, le sequenze predittive delle ultime sequenze di riferimento si sovrappongono con le prime sequenze di riferimento.

A seguire diamo una panoramica dell’andamento globale della tensione della cella minima per ogni dispositivo selezionato. Le sequenze verdi sono quelle sane mentre le sequenze rosse sono quelle patologiche:

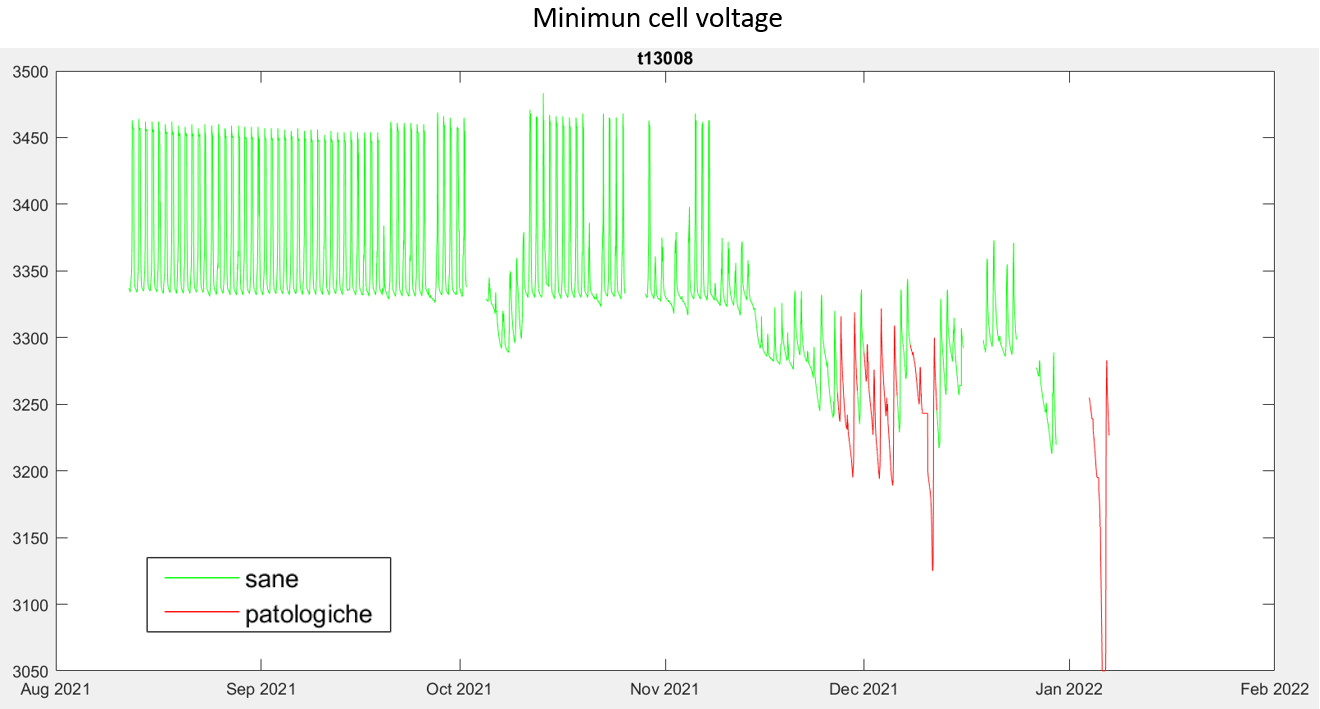


Figura 4‑14: Dispostivo 13008

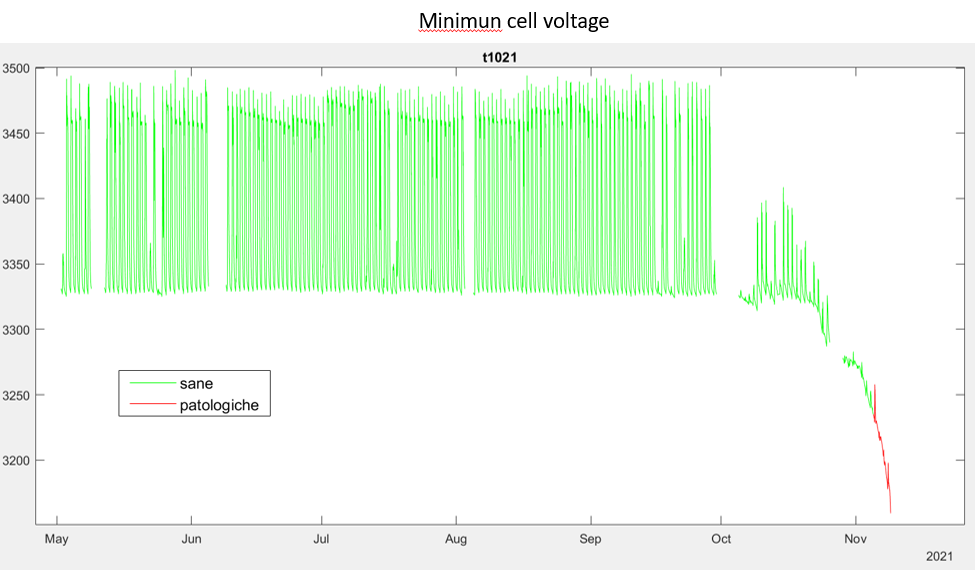


Figura 4‑15: Dispositivo 1021

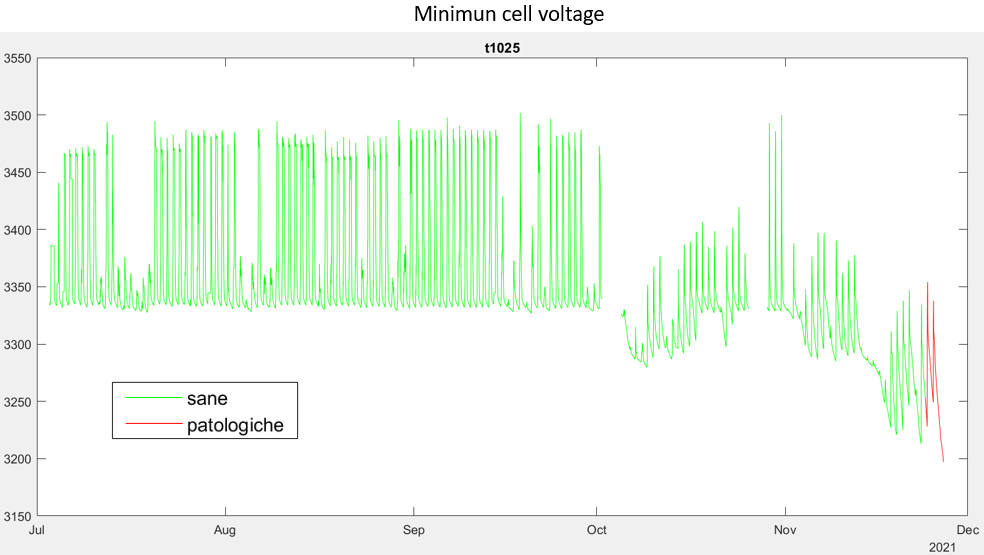


Figura 4‑16: Dispositivo 1025

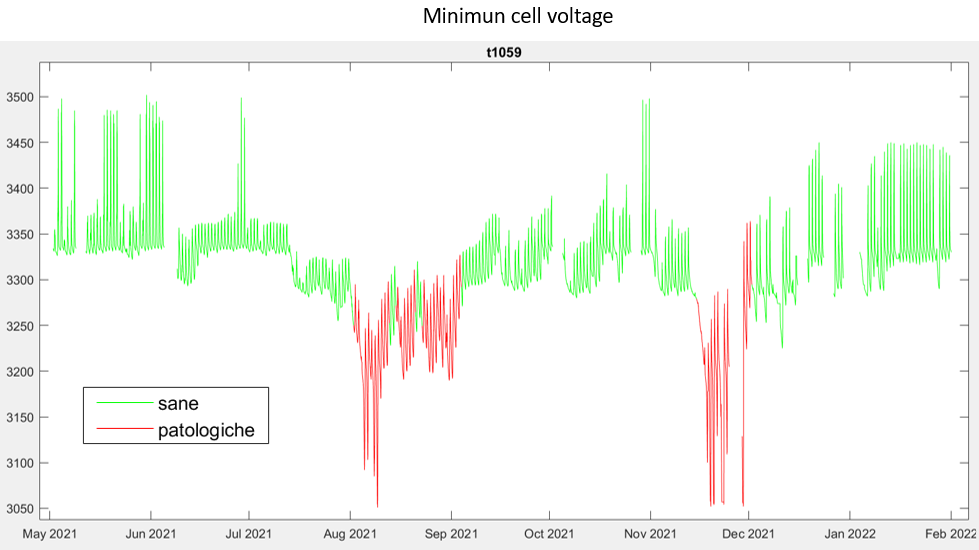


Figura 4‑17: Dispositivo 1059

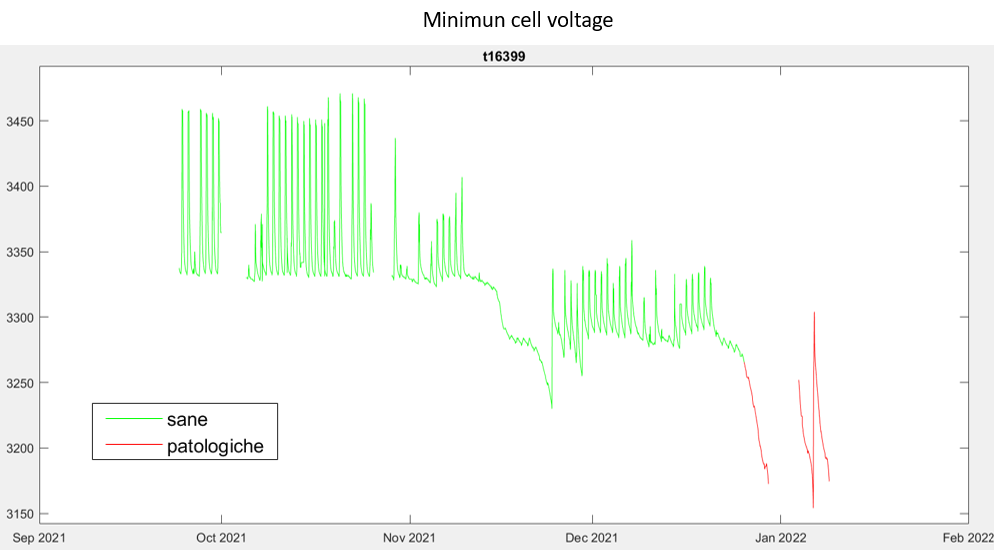


Figura 4‑18: Dispositivo 16399

Vediamo che la casistica è molto eterogenea: In Figura 4‑15 Figura 4‑16 e Figura 4‑18 le sequenze patologiche sono le ultime sequenze prima di uno spegnimento definitivo. Vediamo invece, che nel caso di Figura 4‑14 Figura 4‑17 le sequenze patologiche rappresentano un periodo di forte scarica della batteria, che non sfocia sempre in uno spegnimento (tensione della cella minima a 3050 volt), e che poi viene seguito da una ripresa. Questo perché andando a fissare una soglia fissa per le sequenze patologiche non abbiamo la capacità di discriminare tra questo tipo di situazioni.

Alcuni apparati non coprono tutto l’intervallo temporale perché potrebbero essersi spenti definitivamente (Figura 4‑15, Figura 4‑16), oppure perché la data di implementazione sul campo potrebbe essere successiva al “01-05-2021” (Figura 4‑14, Figura 4‑18).

Un altro punto rilevante è che dato che le sequenze si sovrappongono parzialmente ogni giorno è presente in 3 sequenze. Per questo motivo una sequenza patologica può condividere giorni con sequenze sane. A seguire, vediamo un esempio di tale concetto. Prendiamo in considerazione il dispositivo in Figura 4‑15 nel periodo in cui l’andamento della tensione della cella minima diventa patologico:



Figura 4‑19: Periodo in cui l’andamento della tensione della cella minima diventa patologico

Andiamo a fare uno zoom sul periodo inquadrato in Figura 4‑19 che contiene tre sequenze, due sane e una patologica:

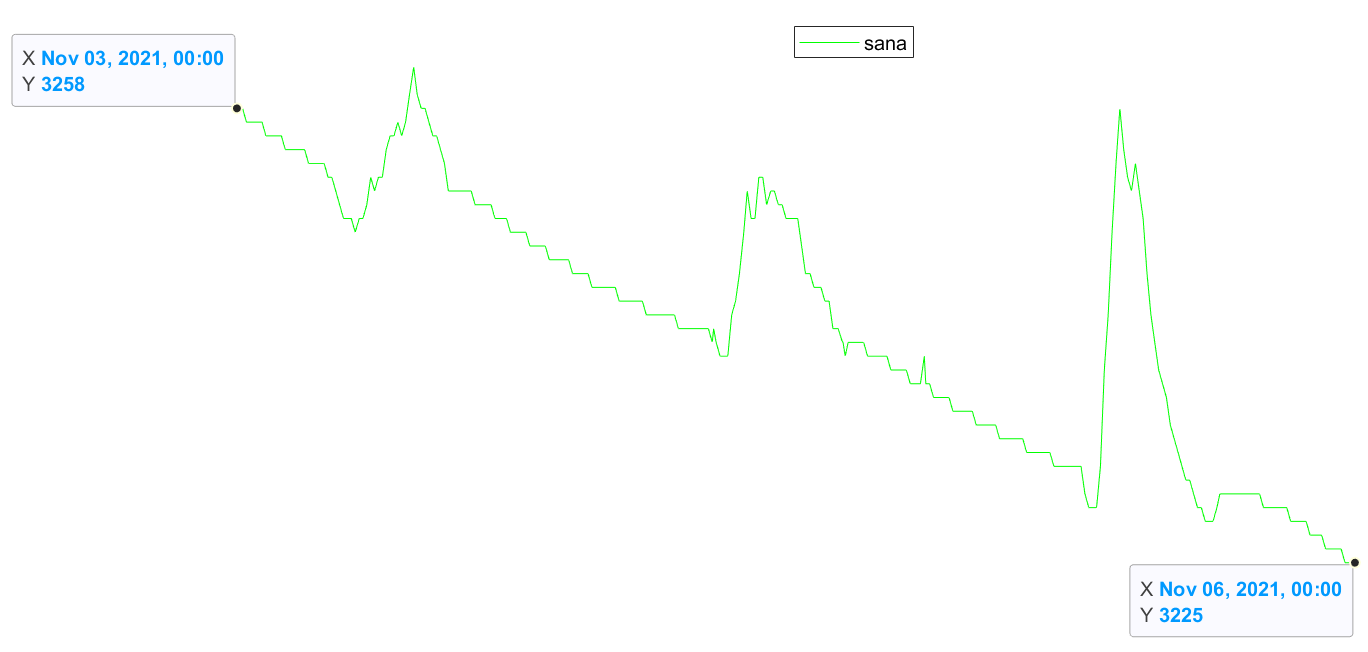


Figura 4‑20: Prima sequenza

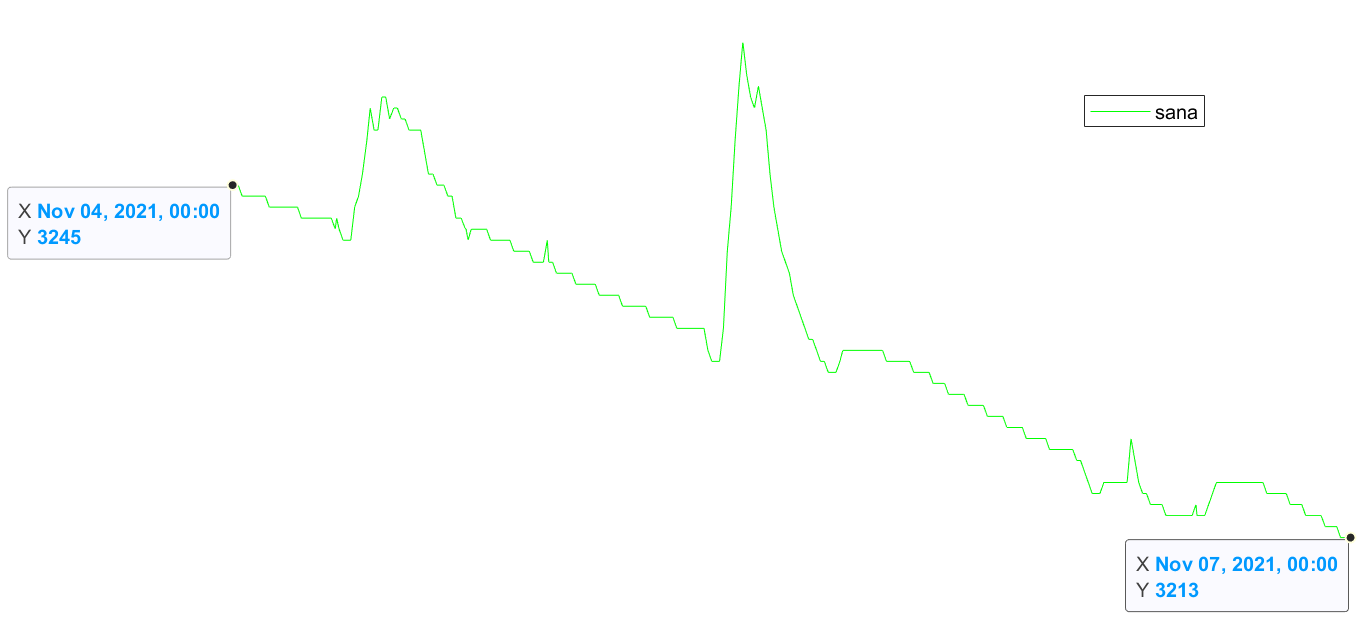


Figura 4‑21: Seconda sequenza

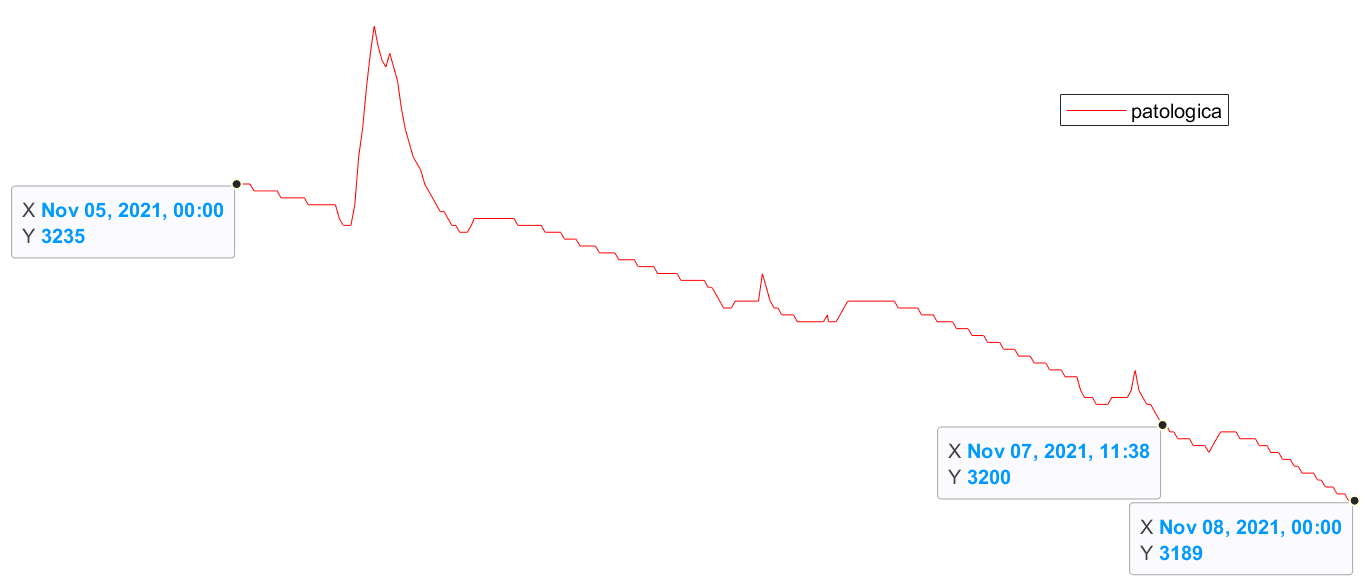


Figura 4‑22: Terza sequenza

Come vediamo in Figura 4‑22 l’evento patologico avviene il 7 novembre, ma la sequenza contiene anche il 5 e il 6 novembre che sono giorni in cui il valore della tensione della cella minima è superiore alla soglia critica. In Figura 4‑21, l’intervallo va dal 4 al 7 novembre e in Figura 4‑20 l’intervallo va dal 3 al 6 novembre. Vediamo che queste tre intervalli comprendono il 5 novembre. Va notato, che il 5 novembre di per sé non contiene un comportamento anomalo, ma fa parte di una sequenza patologica poiché basta che la tensione scenda sotto la soglia in uno dei tre giorni compresi dalla sequenza. Al contrario, il 7 novembre la tensione scende sotto la soglia e quindi è il giorno effettivo che contiene l’evento critico, motivo per cui apparterrà solo a sequenze patologiche e non potrà appartenere a sequenze sane.

In Figura 4‑23 sintetizziamo il concetto:

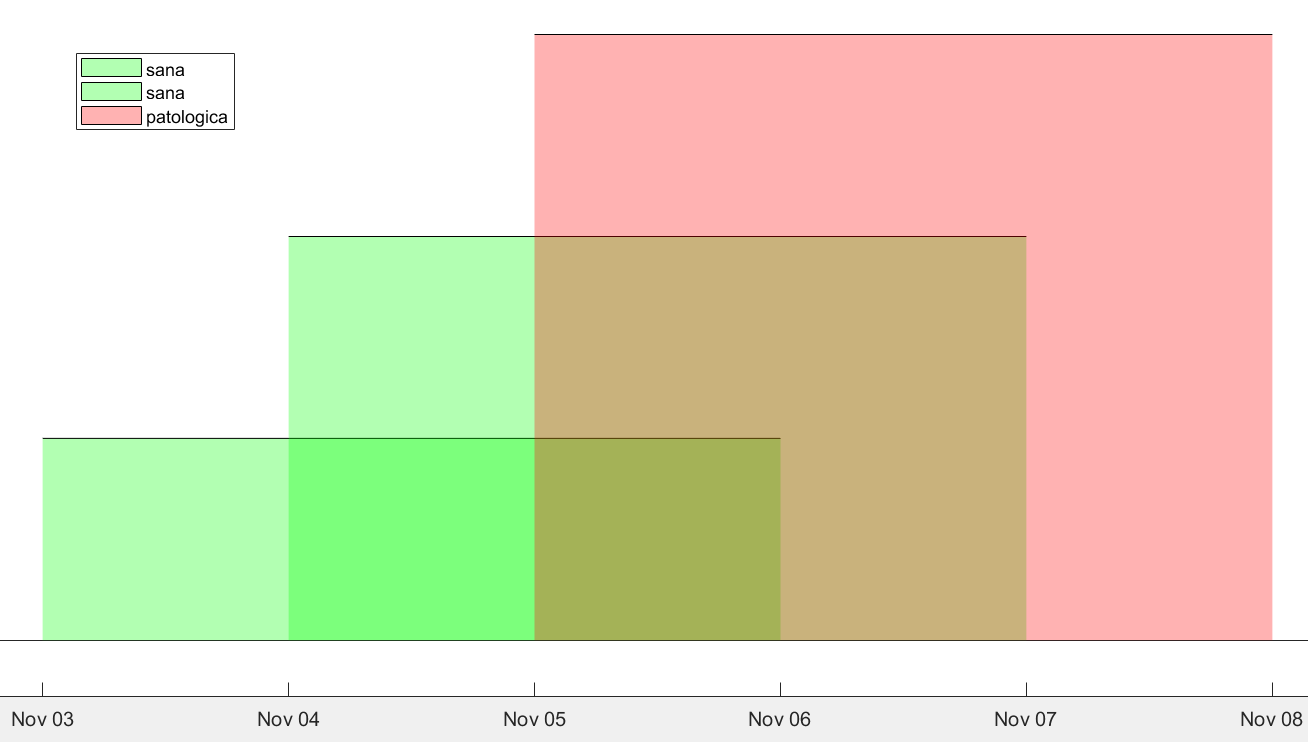


Figura 4‑23: Sfalsamento delle sequenze

In Figura 4‑23 vediamo come il 5 novembre sia compreso in tutte e tre le sequenze. Ciò è vero per tutti i giorni dell’intervallo salvo quelli agli estremi, dato che le sequenze consecutive sono sfalsate di un giorno

In sintesi, il sistema di identificazione degli eventi di guasto non si basa sul giorno dell’evento ma bensì sulla sequenza di appartenenza. Ogni sequenza che contiene un valore inferiore alla soglia viene definita automaticamente come patologica mentre le altre sono sane.

### Normalizzazione delle sequenze

Andiamo ora a normalizzare le sequenze per rendere le grandezze adimensionali. La normalizzazione delle sequenze viene effettuata sottraendo ad ogni campione la media del valore della variabile e dividendo per la deviazione standard. Sia la media che la deviazione standard vengono fatte su tutto l’intervallo di campionamento.

Di seguito il grafico della tensione della cella minima normalizzata:

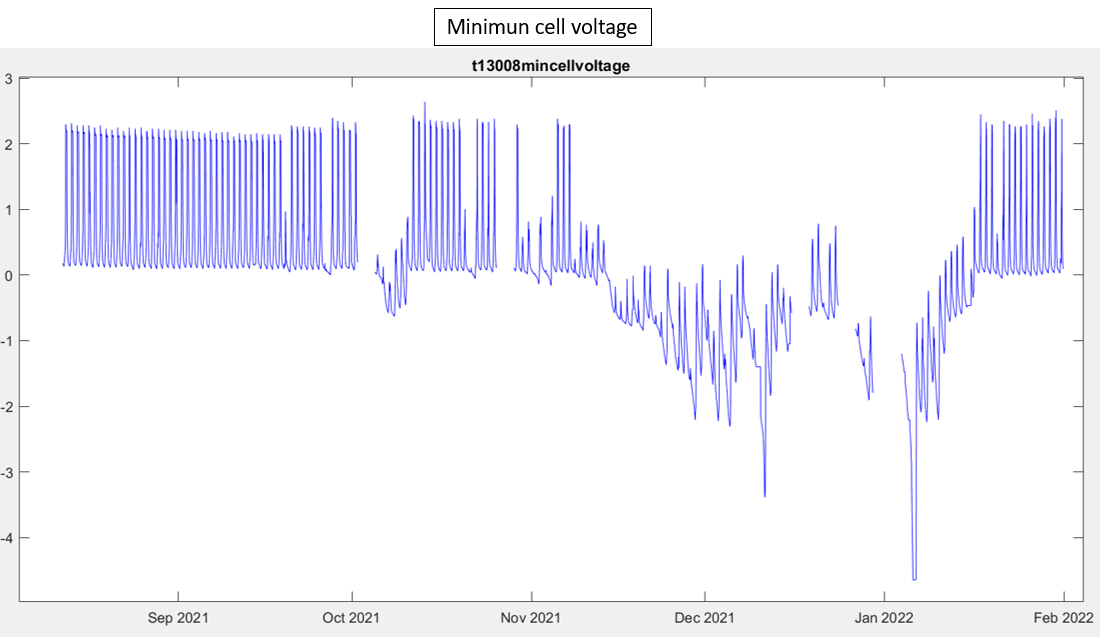


Figura 4‑24: Andamento della tensione della cella minima normalizzata per il dispositvo 13008

Vediamo che rispetto a Figura 4‑14 l’andamento rimane lo stesso, mentre vediamo che il valore va da un massimo approssimativo di 2 fino ad un minimo approssimativo di -4,5, e che viene centrato nell’intorno dello zero.

In seguito, andiamo ad assegnare formalmente le etichette alle sequenze. Dato che facciamo una classificazione binaria, le etichette sono solo due:

* ‘1’: la sequenza è sana
* ‘0’: la sequenza è patologica

### Formazione del dataset finale inglobando le sequenze di tutti i dispositivi

Le operazioni descritte nei paragrafi precedenti (vedi paragrafi dal 4.3.1 al 4.3.3) vengono ripetute per ogni dispositivo. In questo modo otteniamo per ogni dispositivo un insieme di sequenze soddisfacenti le condizioni richieste dai parametri (vedi paragrafo iniziale 4.3). Andiamo ora a mischiare le sequenze relative ai vari dispositivi (vedi appendice in Figura 8‑16).

In questo modo non potremmo più risalire al tipo di dispositivo d’appartenenza di una data sequenza. Ciò non è un problema poiché vogliamo allenare la rete a prevenire i guasti sui vari tipi di dispositivi, nonostante le differenze nei consumi (vedi 3.3.1).

Andiamo a graficare le sequenze sane e patologiche comprese nel dataset finale nel caso in cui abbiamo un intervallo di previsione a 7 giorni, una durata delle sequenze di 3 giorni e uno sfalsamento tra sequenze di 1 giorno:

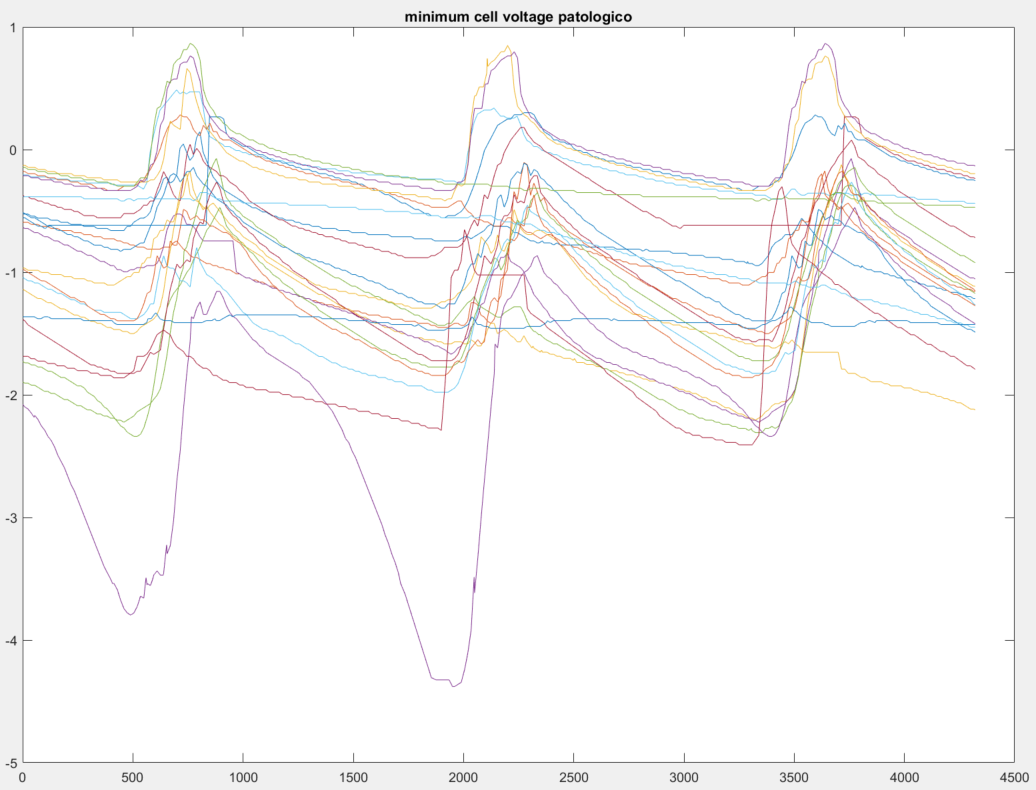


Figura 4‑25: Sequenze patologiche della tensione della cella minima

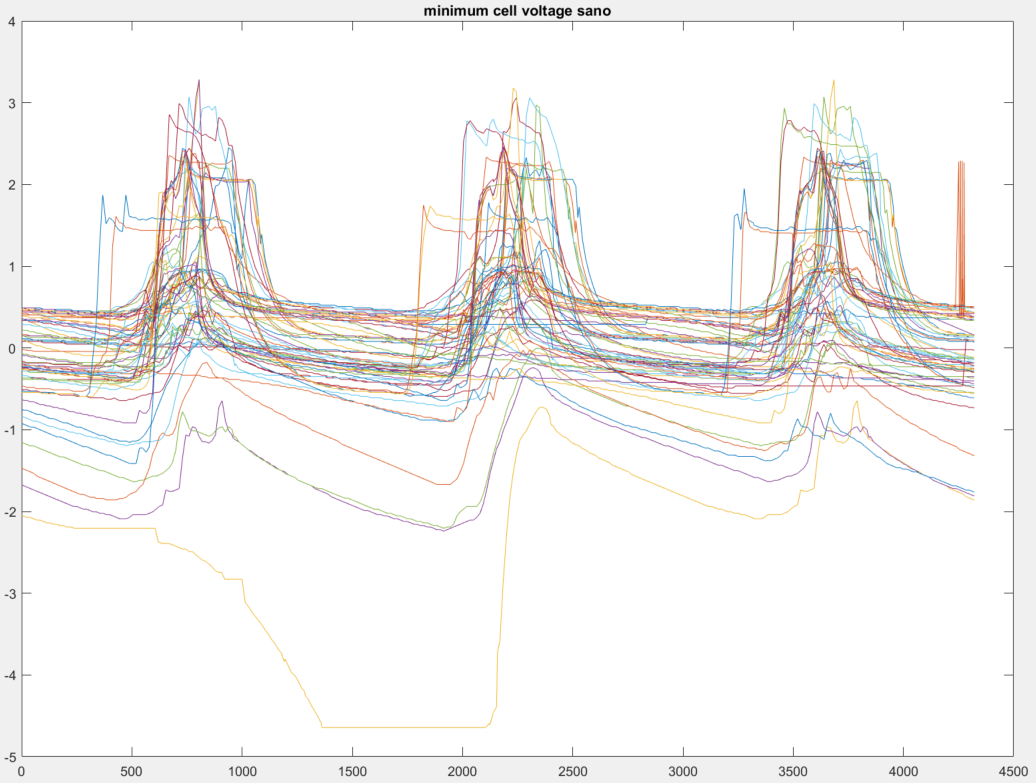


Figura 4‑26: Sequenze sane della tensione della cella minima

In Figura 4‑25 e Figura 4‑26 possiamo osservare l’andamento della tensione della cella minima normalizzata nelle sequenze passate al dataset finale. In Figura 4‑25 vediamo le sequenze patologiche, mentre in Figura 4‑26 vediamo le sequenze sane. Possiamo notare che le sequenze sane incrociano le ordinate soprattutto nell’intorno dello zero e hanno un andamento ciclico che va da 0 ad un massimo di 3. Dato che le sequenze sono normalizzate, il valor medio globale è in 0, e quindi il fatto che le sequenze buone stiano globalmente sopra lo 0 è coerente con la definizione di sequenza sana. Notiamo anche qualche curva che può arrivare anche oltre -4, questo perché stiamo facendo una previsione a 7 giorni e quindi può accadere che una data sequenza sana sia stata preceduta 7 giorni prima da un evento patologico. Questo fenomeno tende a presentarsi maggiormente con l’aumentare dell’intervallo di previsione. Notiamo comunque un andamento ciclico e abbastanza regolare delle curve in Figura 4‑26 che ci permette di individuare a occhio i tre giorni contenuti nelle sequenze. Questa regolarità è una caratteristica che ricerchiamo in sequenze sane poiché è indice di un corretto funzionamento del sistema. Andando invece a vedere Figura 4‑25 vediamo che tale regolarità non si riscontra e che la densità di curve non è concentrata in un punto ma sia mediamente costante tra 0 e -2. Globalmente vediamo che le sequenze sono comprese tra 1 e -4. Questo è dovuto al fatto che le sequenze patologiche hanno un andamento molto più scostante e altalenante a causa delle anomalie.

Andiamo ora ad osservare le sequenze sia patologiche che sane corrispondenti alla potenza del pannello solare:

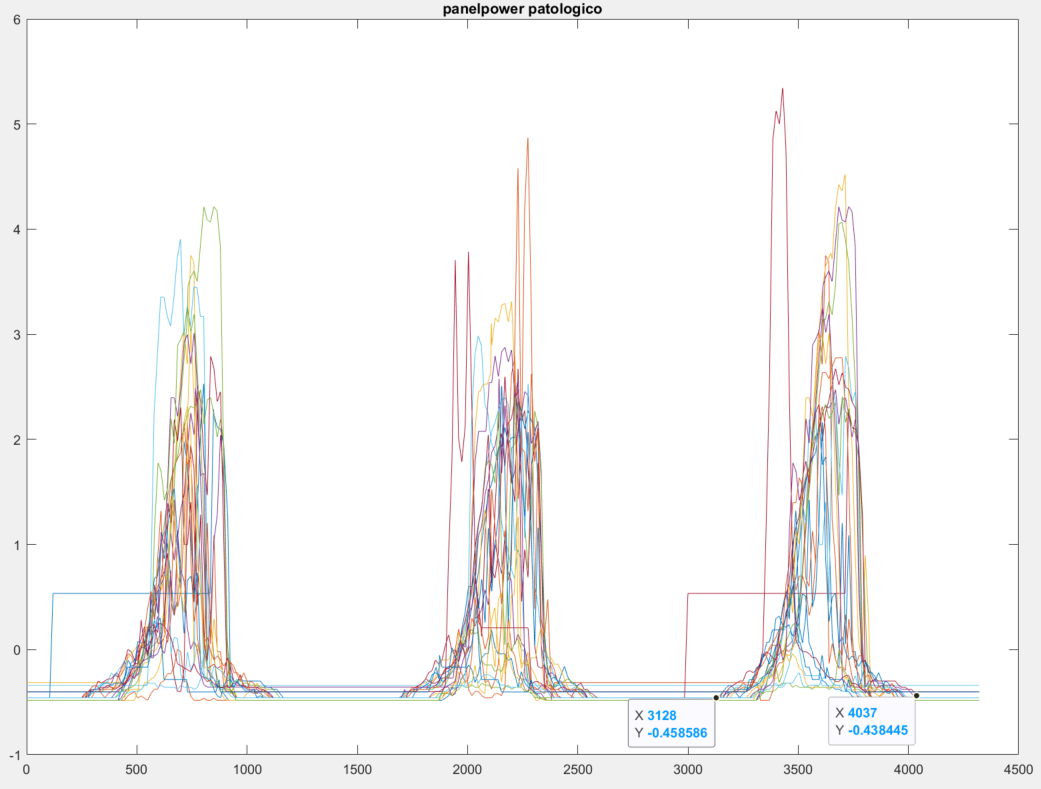


Figura 4‑27: Sequenze patologiche della potenza del pannello

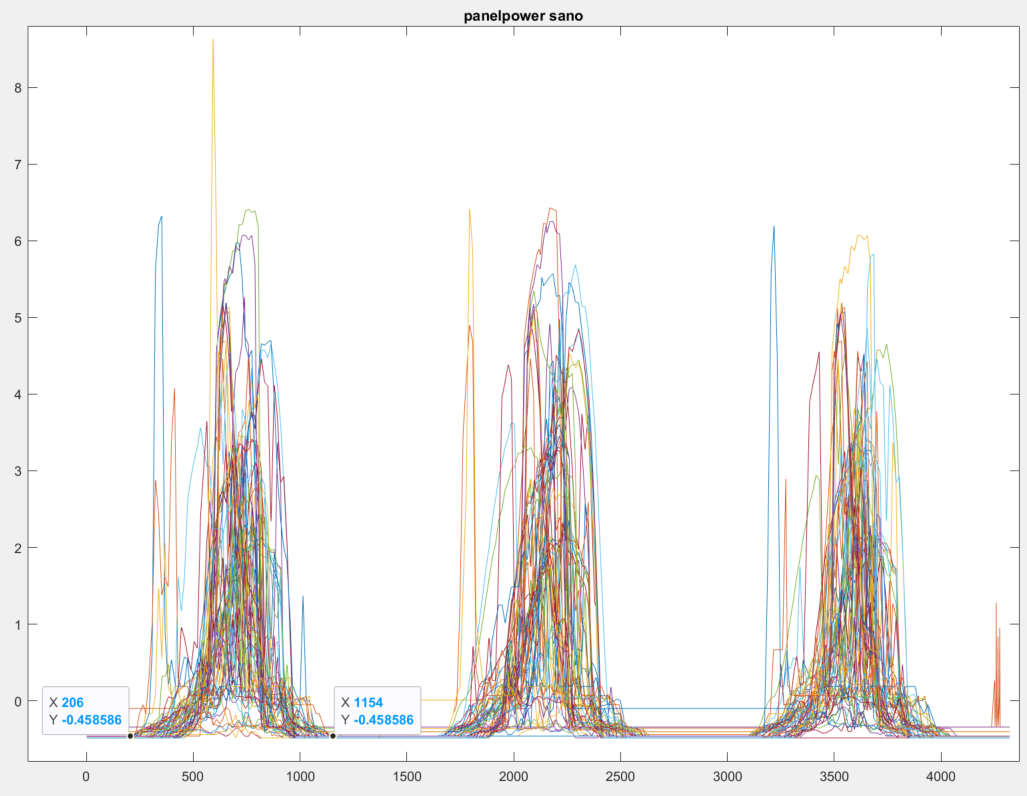


Figura 4‑28: Sequenze sane della potenza del pannello

In Figura 4‑27 vediamo che il valore della potenza del pannello va dall’intorno di 0 fino ad un picco di 7, restando mediamente tra 0 e 4. In Figura 4‑28 l’ampiezza va dall’intorno di 0 fino ad un massimo di 10, restando mediamente tra 0 e 6. Vediamo dunque che l’ampiezza è leggermente maggiore in media per le sequenze sane. Ciononostante, la differenza in termine di ampiezza non è molto marcata. Una differenza che si nota maggiormente è la larghezza dei picchi. Per le sequenze patologiche abbiamo una larghezza massima di 600 punti, mentre per quelle sane di 1000 punti. Inoltre, in Figura 4‑28 vediamo che il lobo sembra essere più ripido sin da subito, mentre in Figura 4‑27 sembra avere una pendenza meno ripida ai margini.

A seguire vediamo l’andamento del SOC:

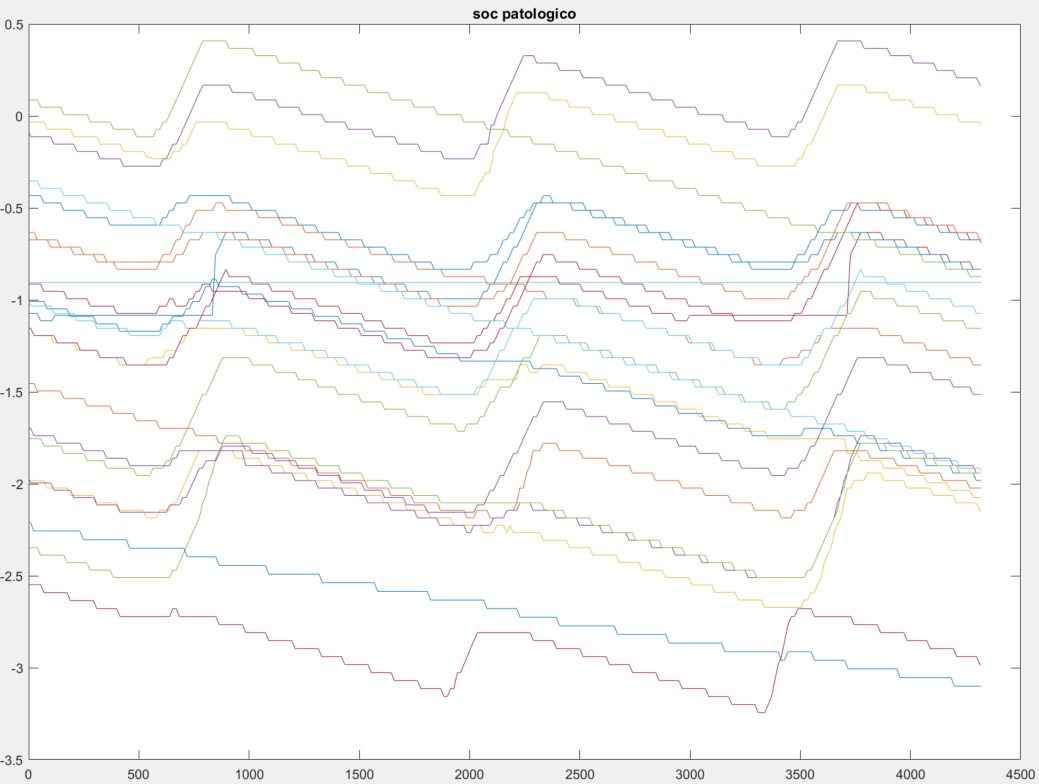


Figura 4‑29: Sequenze patologiche del SOC

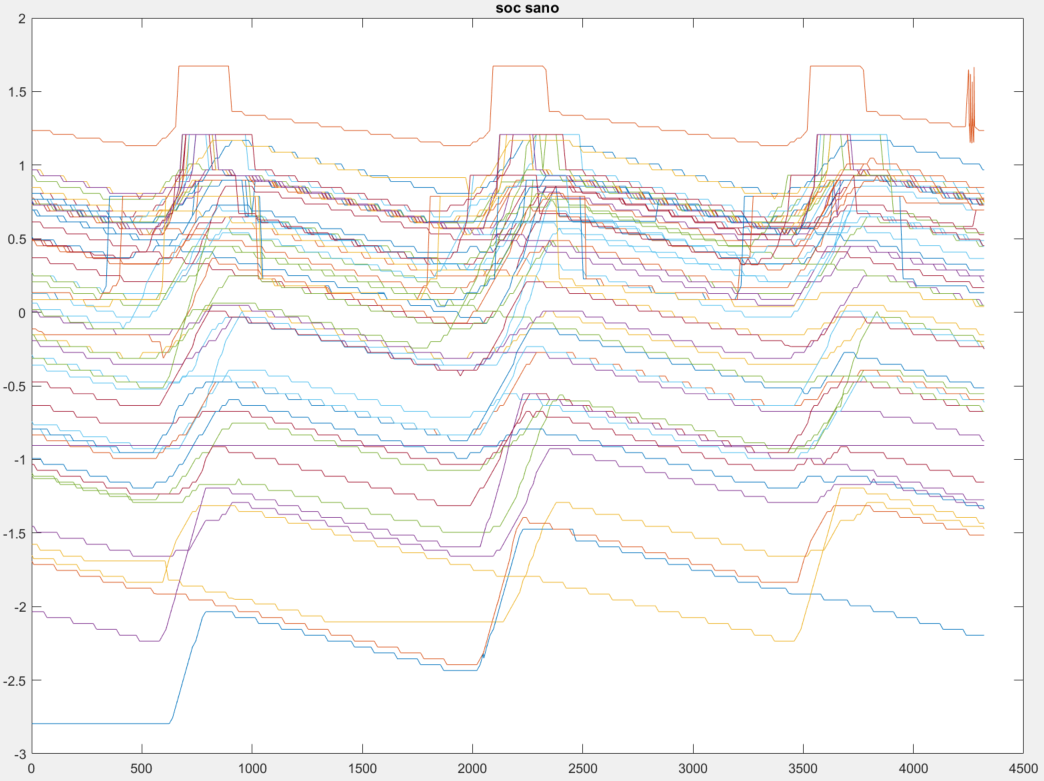


Figura 4‑30: Sequenze sane del SOC

Similmente alla tensione della cella minima (Figura 4‑25, Figura 4‑26) le sequenze sane incrociano le ordinate soprattutto nell’intorno di un punto, in questo caso 0,5, e hanno mediamente un andamento ciclico che va da 0 ad 1. Osserviamo però, che rispetto a Figura 4‑26 è maggiore il numero di curve fuori da questo range, con dei minimi a -2 e dei massimi a 1,5. In Figura 4‑29 le sequenze patologiche incrociano le ordinate in un intervallo che va da 0,5 a -2,5 in modo omogeneo. Notiamo che rispetto a Figura 4‑30, si possono ugualmente individuare i 3 giorni nelle sequenze, anche se meno facilmente. Inoltre, osserviamo che il SOC ha un andamento decrescente in Figura 4‑29.

## Partizione statica e dinamica del dataset

Andiamo ad implementare due tipi di partizioni, una partizione statica e una partizione dinamica. La partizione statica consiste nel separare, durante la creazione del dataset, le sequenze e le relative etichette tra train e test set. In questo modo, tutte le volte che si darà questo dataset in ingresso alla rete essa avrà sempre le stesse sequenze per l’addestramento e per il test dando quindi risultati simili. Al contrario, la partizione dinamica non prevede la divisione tra train e test durante la creazione del dataset. Questa divisione viene fatta durante l’esecuzione della rete da un apposito algoritmo, che si incaricherà di ripartire sempre differentemente le sequenze tra train e test. In questo modo, facendo girare varie volte la rete neurale, essa varierà le sequenze presenti nei due set sfruttando al massimo le combinazioni di dati per creare un modello più performante.

Qualora volessimo una partizione statica andiamo a dividere manualmente i dataset tra training e test set. Secondo la nostra scelta di progetto, il test set contiene il 25% del dataset, e il restante 75% fa parte del training set (vedi appendice in Figura 8‑17 per il codice).

## Creazione di sequenze della durata di 1 giorno

Un caso alternativo a quello appena introdotto in 4.3 è quello in cui le sequenze abbiano una durata di 1 solo giorno invece che 3. Questo caso è rilevante poiché essendo le sequenze sfalsate di 1 giorno tra l’una e l’altra, non ci sarebbe più sovrapposizione tra una sequenza e l’altra (vedi Figura 4‑20, Figura 4‑21, Figura 4‑22 e Figura 4‑23). Per di più, dato che una sequenza viene detta patologica anche quando un solo campione della tensione della cella minima è sotto la soglia di 3,2V, prendendo sequenze di lunghezza minore (1 giorno) otteniamo sequenze con maggiore granularità rispetto all’evento. Per implementare questa modifica ci basta cambiare il parametro riguardante la lunghezza in giorni di cui si fa cenno nella parte introduttiva del paragrafo 4.3.

A seguire vediamo come le sequenze positive e negative non si sovrappongono più:

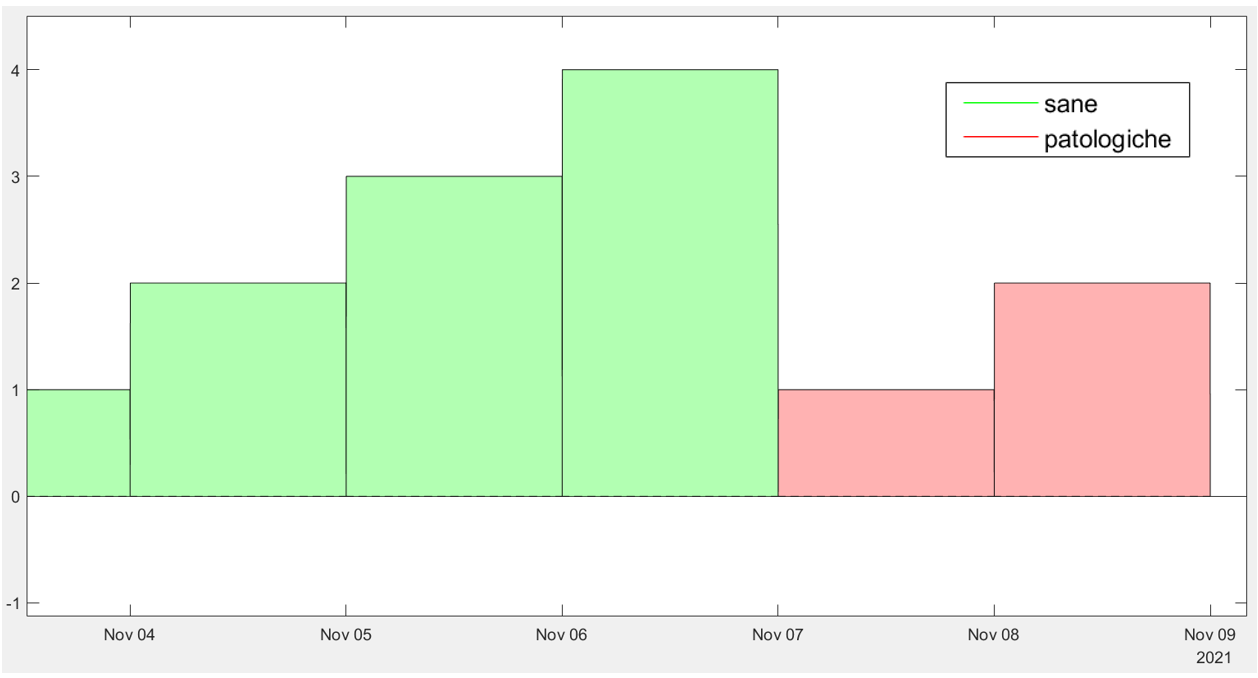


Figura 4‑31: Rappresentazione delle sequenze consecutive lunghe 1 giorno

In Figura 4‑31 vediamo come dove si ferma una sequenza ne comincia un’altra senza però mai sovrapporsi. Il cambio di ampiezza delle barre viene fatto solo per rendere più chiaro dove finisce una sequenza e dove ne comincia un’altra.

Nel capitolo riguardante i risultati andremmo a vedere come i risultati possano variare in base alla lunghezza in giorni delle sequenze.

# Reti neurali proposte

## Composizione della rete

Andiamo a vedere come abbiamo composto la rete neurale. Dato che abbiamo a disposizione l’interezza delle sequenze, usiamo un layer BiLSTM per estrarre le correlazioni sia con gli istanti precedenti che con quelli successivi. Come spiegato nel paragrafo 2.2, il layer LSTM è indicato quando si hanno sequenze con più di 1000 campioni. Nel nostro caso, le sequenze lunghe 1 giorno contengono ognuna 1441 campioni, mentre le sequenze lunghe 3 giorni contengono ognuna 4321 campioni, il che giustifica la nostra scelta di usare un layer LSTM. Inoltre, vogliamo che il layer LSTM sia bidirezionale poiché vogliamo estrarre le correlazioni dei dati usando il contesto passato ma anche quello futuro. La rete BiLSTM produce 2 output per ogni unità nascosta dato che è bidirezionale e quindi abbiamo 30 uscite (vedi Tabella 5‑1). Dopo la rete BiLSTM mettiamo un layer fully connected con 1 neurone in uscita e con una funzione d’attivazione non lineare softmax (16). I livelli fully connected in una rete neurale sono caratterizzati dal fatto che tutti gli input di un livello sono collegati a ogni unità di attivazione del livello successivo. Nei modelli di machine learning più diffusi, gli ultimi livelli sono spesso costituiti da questo tipo di layer perché compilano i dati estratti dai livelli precedenti per formare l'output finale. Il nostro layer prende in ingresso la totalità delle uscite ritornate dal layer BiLSTM e produce due uscite che passa al classification layer (26). La funzione di attivazione softmax fornisce le probabilità che data sequenza appartenga ad una delle 2 classi (16). Il classification layer assegna un’etichetta alla sequenza in base alle probabilità fornite dalla funzione di attivazione softmax e calcola la funzione di cross entropia di perdita. Il classificatore impiegato fa uso di un discriminante di Fisher (27) (vedi appendice in Figura 8‑19 per il codice).

Inoltre, dopo aver ottenuto i risultati per la suddetta rete neurale (che chiameremo rete neurale di base), abbiamo deciso di crearne due varianti aggiungendo opportunamente dei layers, nell’ottica di rendere più completa l’analisi svolta. In particolare, è stata proposta una prima rete alternativa, provvista di un layer BiLSTM in cascata a quello già presente, mentre nel secondo caso sono stati aggiunti due layers fully connected, con funzione di attivazione ReLU. Ulteriori dettagli e chiarimenti riguardo queste due reti di approfondimento sono forniti nell’introduzione dei paragrafi 6.2 e 6.3.

## Parametri

I parametri specifici della nostra rete sono i seguenti:

Tabella 5‑1: Parametri della rete

|  |  |
| --- | --- |
| Numero di unità nascoste | 15 |
| Numero di classi | 2 |
| Numero massimo di epoche | 8 |
| Fattore di apprendimento iniziale | 0.04 |

In Tabella 5‑1, il primo parametro rappresenta il numero di neuroni presenti nel layer BiLSTM che passiamo alla rete. Il numero di unità nascoste corrisponde alla quantità di informazioni ricordate tra i passaggi temporali (hidden state). È importante non inserire un numero di unità nascoste troppo grande poiché la rete potrebbe andare in overfitting. Dopo diverse prove, troviamo che i risultati migliori con il tempo di addestramento minore si trovano con 15 unità nascoste. Il numero di classi rappresenta il numero di valori che può avere un’etichetta. Nel nostro caso abbiamo solo 2 classi poiché facciamo una classificazione binaria. Per quanto riguarda il numero massimo di epoche, dopo diverse prove ci siamo indirizzati su 8 epoche. Ponendo un maggiore numero di epoche troviamo che la curva dell’accuratezza assume un andamento periodico e manda in overfitting il modello offrendo dei risultati di accuratezza peggiori. Il learning rate iniziale influisce sulla velocità di apprendimento della rete e dunque sulla velocità di convergenza della rete verso un minimo della funzione di perdita (14). Dopo diverse prove abbiamo concordato che un valore di 0,04 portasse ai risultati migliori. Scegliendone uno minore ottenevamo lo stesso risultato ma in più epoche, oppure ottenevamo risultati peggiori. Uno maggiore invece faceva divergere la curva di apprendimento.

## Training Options

Andiamo ora ad impostare le opzioni della rete:

Tabella 5‑2: Training options

|  |  |
| --- | --- |
| Solver | Adam |
| Learn rate schedule | Piecewise |
| Learn rate drop factor | 0.5 |
| Learn rate drop period | 2 |
| Soglia massima del gradiente | 1 |
| Shuffle | Ad ogni epoca |
| Rappresentazione grafica dell’andamento dell’accuratezza e della funzione di perdita | |

Il solver che andiamo ad utilizzare è “Adam”(29). Scegliamo di usare un fattore di decrescimento del learning rate iniziale pari al 50 % ogni 2 epoche di modo che all’avanzare del training l’apprendimento scali. Per evitare l’esplosione del gradiente andiamo a scegliere 1 come valore massimo del gradiente. Le dimensioni dei mini-batch varieranno in base alle dimensioni del dataset tenendo in conto che vogliamo mantenere il mini-batch di una dimensione che sia un divisore delle dimensioni dei set (Training e Test). Nel nostro caso varia tra 10 e 30 elementi. Infine, scegliamo di impostare uno shuffle ad ogni epoca poiché le sequenze tra loro non presentano legami di consequenzialità. Abilitando questa opzione vogliamo evitare che il modello resti in un minimo locale non raggiungendo un minimo globale della funzione di perdita. Inoltre, nel caso in cui il mini-batch size non sia un divisore della dimensione del set, evitiamo che siano sempre le stesse sequenze ad essere escluse dal set.

A seguire il grafico relativo alla lunghezza delle sequenze:

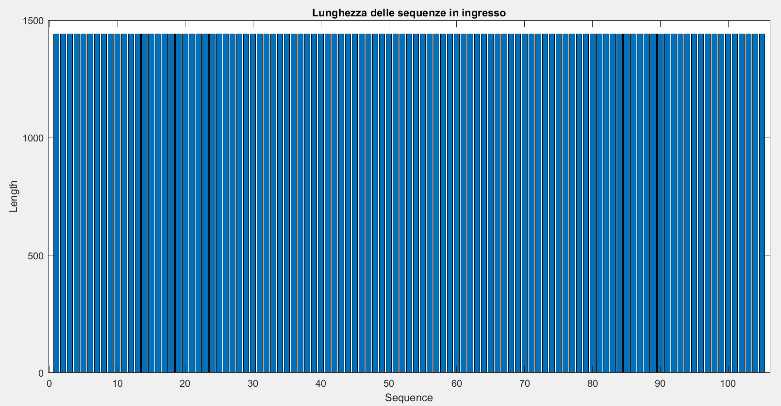


Figura 5‑1: Lunghezza delle sequenze uniforme

Grazie alla sincronizzazione dei dati, le sequenze hanno tutte la stessa lunghezza e quindi non abbiamo necessità di introdurre né padding né sorting.

## Training e classificazione

Dopo aver impostato i parametri possiamo avviare il training e la classificazione. A seguire un andamento dell’accuratezza e della funzione di perdita con le suddette opzioni della rete:

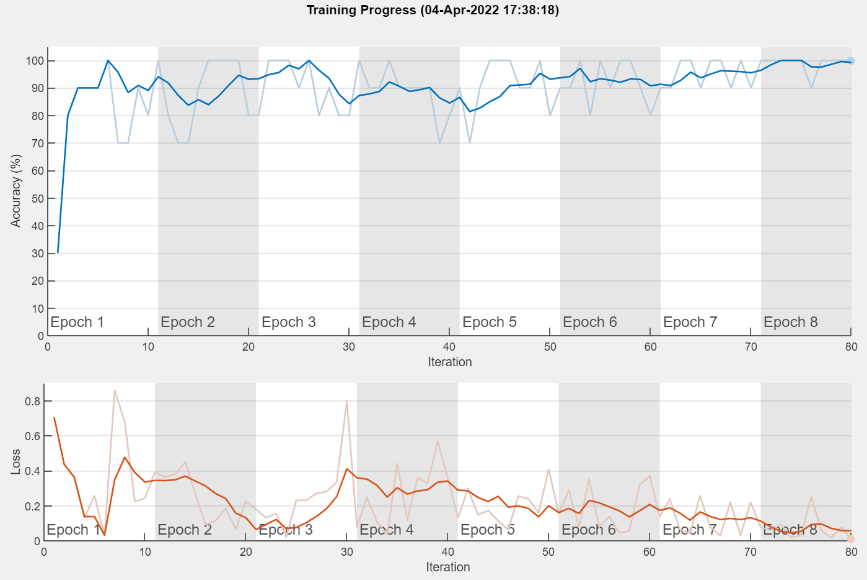


Figura 5‑2: Andamento tipico dell'accuratezza e della funzione di perdita con le suddette opzioni della rete

Come vediamo in Figura 5‑2 dopo 5/6 epoche l’andamento dell’accuratezza (curva blu) tende a stabilizzarsi sopra il 90% e la loss (curva rossa) rimane sotto lo 0,2.

Una volta effettuata la classificazione calcoliamo l’accuratezza e grafichiamo la confusion chart:



Figura 5‑3: Confusion chart

In Figura 5‑3, vediamo la confusion chart che rappresenta come sono stati classificati gli elementi del dataset rispetto alle proprie etichette di riferimento. Sulla diagonale principale ci sono gli elementi correttamente classificati, in (1,0) troviamo i falsi negativi e in (0,1) i falsi positivi. Nel riquadro di destra invece troviamo per ogni classe le percentuali di elementi classificati correttamente e la percentuale di quelli classificati in modo errato rispetto alla classe di riferimento.

## K-Folding

In questo paragrafo, andiamo a spiegare che cos’è il k-folding e perché scegliamo di farne uso al fine di aumentare l’affidabilità dei risultati che otteniamo. Il k-folding è una procedura di ri-campionamento utilizzata per valutare i modelli di apprendimento automatico su un campione di dati limitato. La procedura ha un unico parametro chiamato k che si riferisce al numero di gruppi in cui deve essere suddiviso il dataset. Viene utilizzato principalmente per stimare l'abilità di un modello su dati sconosciuti. In prima battuta, la procedura generale è di mescolare il set di dati in modo casuale e dividerlo in k gruppi. Dopodiché si ripete k volte la seguente procedura:

* Prendere 1 gruppo e usarlo come test set
* Usare i restanti gruppi come set di addestramento
* Allenare il modello sul training set e valutarlo sul test set
* Conservare il punteggio di valutazione ed eliminare il modello

Ripetere la procedura k volte usando a rotazione sempre un gruppo diverso per il test set. Alla fine, riassumiamo l'abilità del modello facendo una media delle accuratezze ottenute (28).

È buona norma ripetere la procedura di k-folding per un numero n di volte per avere il risultato più fedele possibile sulle capacità di apprendimento della rete su tali dati.

Nel nostro caso scegliamo di dividere il set in 4 fold e di ripetere la procedura 10 volte. In questo modo speriamo di ottenere un risultato di accuratezza che rispecchi il più possibile le capacità di apprendimento del modello con i dati a disposizione (vedi appendice in Figura 8‑18 per il codice). I 4 fold verranno usati a rotazione per formare il training set e il test set. In particolare, 3 fold formeranno il training set e 1 il test set. Da notare che durante questa suddivisione in gruppi viene mantenuta automaticamente la proporzione presente nel dataset originale tra sequenze sane e patologiche. Per evitare che le partizioni estratte siano le stesse, impostiamo ad ogni iterazione il seed del generatore pseudo random uguale al numero dell’iterazione in corso. Una volta creato un modello per ogni permutazione del dataset e fatta la classificazione, facciamo la confusion chart rispetto alla globalità del dataset e calcoliamo l’accuratezza come media delle accuratezze ottenute per le 4 permutazioni del dataset. Rispetto alla partizione statica abbiamo il vantaggio che sia la confusion chart che l’accuratezza siano rispetto a tutto il dataset e quindi siano più consistenti.

# Risultati sperimentali

Andiamo ora a presentare i risultati ottenuti. In 6.1.1 mostriamo i risultati ottenuti usando la rete neurale presentata in 5 (rete di base). In 6.2 e 6.3 vengono proposte 2 alternative alla rete di base su cui vengono eseguite le stesse prove effettuate in 6.1. Infine, in 6.4 confrontiamo i risultati ottenuti con le 3 reti neurali.

Le prove eseguite si distinguono per la combinazione di variabili che forniamo in ingresso alla rete. La prima prova, viene effettuata dando alla rete solo la tensione della cella minima e la potenza del pannello. La seconda combinazione, contiene oltre alla tensione della cella minima e la potenza del pannello, anche il SOC e l’irraggiamento. La terza combinazione, contiene tutte le variabili, ossia la tensione della cella minima, la potenza del pannello, il SOC, l’irraggiamento ed il bilancio delle correnti nella batteria.

Inoltre, vogliamo osservare i risultati prodotti dalle reti, ponendoci a diversi intervalli di tempo dall’evento che vogliamo prevedere. In particolare, alleniamo dei modelli a prevedere un guasto a distanza di 1 giorno, poi a distanza di 3 giorni, ed infine a distanza di 7 giorni.

Infine, variamo la lunghezza delle sequenze per constatare se ciò possa avere dei riscontri sull’affidabilità dei modelli. Nella fattispecie, distinguiamo i tentativi eseguiti fornendo in ingresso delle sequenze della durata di 1 giorno, da quelli in cui diamo in ingresso delle sequenze della durata di 3 giorni.

## Rete neurale di base

### Tensione della cella minima e potenza del pannello

#### Previsione ad 1 giorno con sequenze lunghe 3 giorni

Poniamoci nel caso in cui le sequenze durino 3 giorni, siano sfalsate di 1 giorno, e ci sia una proporzione di sequenze tale che per una sequenza patologica ce ne siano 3 sane. Vogliamo prevedere l’evento 1 giorno prima. La dimensione dei mini-batch è di 12 elementi con dei fold ognuno da 48 elementi, per un totale di 192 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza del 94% con 0,99% di deviazione standard. L’errore di classificazione è approssimativamente del 6%.

Per quanto riguarda la confusion matrix, le righe rappresentano le classi di riferimento e le colonne rappresentano le classi assegnate alle sequenze dal modello predittivo. La prima riga rappresenta la classe patologica e la seconda la classe sana:

Tabella 6‑1: Confusion Matrix per predizioni con anticpo di 1 giorno e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 43.4 ± 2.2 | 4.6 ± 2.2 |
| 1 | 6.8 ± 2.2 | 137.2 ± 2.2 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑1, evidenzia che sulla totalità del dataset mediamente 6,8 elementi vengono classificati erroneamente come patologici e 4,6 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard della confusion matrix ci dice la differenza di ogni elemento della confusion matrix tra una run e l’altra e quindi la variazione di prestazione in base alla scelta dei fold.

#### Previsione ad 1 giorno con sequenze lunghe 1 giorno

Poniamoci nelle stesse condizioni del caso precedente ma con sequenze di durata 1 giorno. La dimensione dei mini-batch è di 7 elementi con dei fold ognuno da 35 elementi, per un totale di 140 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza del 97% con 1,36% di deviazione standard. L’errore di classificazione è approssimativamente del 3%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑2: Confusion matrix per predizioni con anticipo di 1 giorno e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 34.6 ± 0.5 | 0.4 ± 0.5 |
| 1 | 4.0 ± 1.6 | 101.0 ± 1.6 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑2 evidenzia che sulla totalità del dataset mediamente 4 elementi vengono classificati erroneamente come patologici e 0.4 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi sani.

#### Previsione a 3 giorni con sequenze lunghe 3 giorni

In questo caso manteniamo gli stessi parametri ma impostiamo sequenze lunghe 3 giorni e intervallo di previsione uguale a 3 giorni.

La dimensione dei mini-batch è di 23 elementi con dei fold ognuno da 46 elementi, per un totale di 184 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 91% con 0,99% di deviazione standard. L’errore di classificazione è approssimativamente del 9%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑3: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 39.8 ± 1.6 | 6.2 ± 1.6 |
| 1 | 9.8 ± 1.4 | 128.2 ± 1.4 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑3 evidenzia che sulla totalità del dataset mediamente 9,8 elementi vengono classificati erroneamente come patologici e 6.2 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi patologici.

#### Previsione a 3 giorni con sequenze lunghe 1 giorno

In questo caso manteniamo gli stessi parametri ma impostiamo sequenze lunghe 1 giorno.

La dimensione dei mini-batch è di 16 elementi con dei fold ognuno da 32 elementi, per un totale di 128 sequenze nel dataset. Da notare che il numero di sequenze è minore rispetto al caso precedente.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 91% con 1,4% di deviazione standard. L’errore di classificazione è approssimativamente del 9%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑4: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 25.7 ± 2.1 | 6.3 ± 2.1 |
| 1 | 4.8 ± 1.8 | 91.2 ± 1.8 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑4 evidenzia che sulla totalità del dataset mediamente 4,8 elementi vengono classificati erroneamente come patologici e 6,3 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi patologici.

#### Previsione a 7 giorni con sequenze lunghe 3 giorni

In questo caso manteniamo gli stessi parametri e impostiamo l’intervallo di previsione uguale a 7 giorni,

La dimensione dei mini-batch è di 23 elementi con dei fold ognuno da 46 elementi, per un totale di 184 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 82% con 1,39% di deviazione standard. L’errore di classificazione è approssimativamente del 18%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑5: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 30.8 ± 1.7 | 15.2 ± 1.7 |
| 1 | 17.9 ± 2.9 | 120.1 ± 2.9 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑5 evidenzia che sulla totalità del dataset mediamente 17,9 elementi vengono classificati erroneamente come patologici e 15,2 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi sani.

#### Previsione a 7 giorni con sequenze lunghe 1 giorno

In questo caso manteniamo gli stessi parametri del paragrafo precedente impostando però la lunghezza delle sequenze di 1 giorno.

La dimensione dei mini-batch è di 10 elementi con dei fold ognuno da 31 elementi, per un totale di 124 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 84% con 1,14% di deviazione standard. L’errore di classificazione è approssimativamente del 16%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑6: Confusion matrixper predizioni con anticipo di 7 giorni e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 18.6 ± 1.1 | 12.4 ± 1.1 |
| 1 | 7.3 ± 1.3 | 85.7 ± 1.3 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑6 evidenzia che sulla totalità del dataset mediamente 7,3 elementi vengono classificati erroneamente come patologici e 12,4 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi sani.

### Tensione della cella minima, potenza del pannello, SOC e irraggiamento

Diamo in ingresso alla rete la combinazione di tensione della cella minima, potenza del pannello, lo stato di carica della batteria (SOC) e l’irraggiamento del pannello.

#### Previsione ad 1 giorno con sequenze lunghe 3 giorni

Poniamoci nel caso in cui le sequenze durino 3 giorni, siano sfalsate di 1 giorno, e ci sia una proporzione di sequenze tale che per una sequenza patologica ce ne siano 3 sane. La dimensione dei mini-batch è di 12 elementi con dei fold ognuno da 47 elementi, per un totale di 188 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 95% con 1,12% di deviazione standard. L’errore di classificazione è approssimativamente del 5%.

Per quanto riguarda la confusion matrix, le righe rappresentano le classi di riferimento e le colonne rappresentano le classi assegnate alle sequenze dal modello predittivo. La prima riga rappresenta la classe patologica e la seconda la classe sana:

Tabella 6‑7: Confusion matrix per predizioni con anticipo di 1 giorno e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 42.2 ± 2.2 | 4.8 ± 2.2 |
| 1 | 5.2 ± 0.9 | 135.8 ± 0.9 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑7 evidenzia che sulla totalità del dataset mediamente 5,2 elementi vengono classificati erroneamente come patologici e 4,8 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi patologici.

#### Previsione ad 1 giorno con sequenze lunghe 1 giorno

Poniamoci nelle stesse condizioni del caso precedente ma con sequenze di durata 1 giorno.

La dimensione dei mini-batch è di 17 elementi con dei fold ognuno da 34 elementi, per un totale di 136 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza del 97% con 1,59% di deviazione standard. L’errore di classificazione è approssimativamente del 3%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑8: Confusion matrix per predizioni con anticpo di 1 giorno e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 33.1 ± 1.0 | 0.9 ± 1.0 |
| 1 | 3.1 ± 1.5 | 98.9 ± 1.5 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑8 evidenzia che sulla totalità del dataset mediamente 3.1 elementi vengono classificati erroneamente come patologici e 0.9 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi sani.

#### Previsione a 3 giorni con sequenze lunghe 3 giorni

In questo caso manteniamo gli stessi parametri ma impostiamo sequenze lunghe 3 giorni e intervallo di previsione uguale a 3 giorni.

La dimensione dei mini-batch è di 15 elementi con dei fold ognuno da 45 elementi, per un totale di 180 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 91% con 1,49% di deviazione standard. L’errore di classificazione è approssimativamente del 9%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑9: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 37.5 ± 1.9 | 7.5 ± 1.9 |
| 1 | 8.1 ± 1.4 | 126.9 ± 1.4 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑9 evidenzia che sulla totalità del dataset mediamente 8,1 elementi vengono classificati erroneamente come patologici e 7,5 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi patologici.

#### Previsione a 3 giorni con sequenze lunghe 1 giorno

In questo caso manteniamo gli stessi parametri ma impostiamo sequenze lunghe 1 giorno.

La dimensione dei mini-batch è di 10 elementi con dei fold ognuno da 31 elementi, per un totale di 124 sequenze nel dataset. Da notare che il numero di sequenze è minore rispetto al caso precedente.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 91% con 1,32% di deviazione standard. L’errore di classificazione è approssimativamente del 9%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑10: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 26.9 ± 1.2 | 4.1 ± 1.2 |
| 1 | 7.6 ± 0.8 | 85.4 ± 0.8 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑10 evidenzia che sulla totalità del dataset mediamente 7,6 elementi vengono classificati erroneamente come patologici e 4,1 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi patologici.

#### Previsione a 7 giorni con sequenze lunghe 3 giorni

In questo caso manteniamo gli stessi parametri e impostiamo l’intervallo di previsione uguale a 7 giorni.

La dimensione dei mini-batch è di 15 elementi con dei fold ognuno da 46 elementi, per un totale di 184 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 82% con 0,96% di deviazione standard. L’errore di classificazione è approssimativamente del 18%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑11: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 31.2 ± 1.6 | 13.8 ± 1.6 |
| 1 | 18.3 ± 1.5 | 116.7 ± 1.5 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑11 evidenzia che sulla totalità del dataset mediamente 18,3 elementi vengono classificati erroneamente come patologici e 13,8 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi sani.

#### Previsione a 7 giorni con sequenze lunghe 1 giorno

In questo caso manteniamo gli stessi parametri del paragrafo precedente impostando però la lunghezza delle sequenze di 1 giorno.

La dimensione dei mini-batch è di 10 elementi con dei fold ognuno da 30 elementi, per un totale di 120 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 82% con 1,51% di deviazione standard. L’errore di classificazione è approssimativamente del 18%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑12: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 18.5 ± 1.3 | 11.5 ± 1.3 |
| 1 | 10.3 ± 1.5 | 79.7 ± 1.5 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑12 evidenzia che sulla totalità del dataset mediamente 10,3 elementi vengono classificati erroneamente come patologici e 11,5 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi sani.

### Tensione della cella minima, potenza del pannello, SOC, irraggiamento e bilancio delle correnti della batteria

Usiamo la combinazione di tensione della cella minima, potenza del pannello, lo stato di carica della batteria (SOC), l’irraggiamento del pannello e la corrente totale della batteria (bilancio tra quella in entrata e in uscita).

#### Previsione ad 1 giorno con sequenze lunghe 3 giorni

Poniamoci nel caso in cui le sequenze durino 3 giorni, siano sfalsate di 1 giorno, e ci sia una proporzione di sequenze tale che per una sequenza patologica ce ne siano 3 sane.

La dimensione dei mini-batch è di 12 elementi con dei fold ognuno da 47 elementi, per un totale di 188 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 93% con 0,76% di deviazione standard. L’errore di classificazione è approssimativamente del 7%.

Per quanto riguarda la matrice di confusione:

Tabella 6‑13: Confusion matrix per predizioni con anticipo di 1 giorno e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 40.2 ± 1.6 | 6.8 ± 1.6 |
| 1 | 5.6 ± 1.5 | 135.4 ± 1.5 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑13 evidenzia che sulla totalità del dataset mediamente 5,6 elementi vengono classificati erroneamente come patologici e 6,8 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard per la classificazione degli elementi patologici e sani è approssimativamente uguale.

#### Previsione ad 1 giorno con sequenze lunghe 1 giorno

Poniamoci nelle stesse condizioni del caso precedente ma con sequenze di durata 1 giorno.

La dimensione dei mini-batch è di 17 elementi con dei fold ognuno da 34 elementi, per un totale di 136 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza del 98% con 0,83% di deviazione standard. L’errore di classificazione è approssimativamente del 2%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑14: Confusion matrix per predizioni con anticipo di 1 giorno e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 32.6 ± 1.4 | 1.4 ± 1.4 |
| 1 | 1.8 ± 1.2 | 100.2 ± 1.2 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑14 evidenzia che sulla totalità del dataset mediamente 1.8 elementi vengono classificati erroneamente come patologici e 1.4 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi patologici.

#### Previsione a 3 giorni con sequenze lunghe 3 giorni

In questo caso manteniamo gli stessi parametri ma impostiamo sequenze lunghe 3 giorni e intervallo di previsione uguale a 3 giorni.

La dimensione dei mini-batch è di 15 elementi con dei fold ognuno da 45 elementi, per un totale di 180 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 90% con 1,29% di deviazione standard. L’errore di classificazione è approssimativamente del 10%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑15: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 37.9 ± 1.4 | 7.1 ± 1.4 |
| 1 | 10.4 ± 2.0 | 124.6 ± 2.0 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑15 evidenzia che sulla totalità del dataset mediamente 10,4 elementi vengono classificati erroneamente come patologici e 7,1 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi sani.

#### Previsione a 3 giorni con sequenze lunghe 1 giorno

In questo caso manteniamo gli stessi parametri ma impostiamo sequenze lunghe 1 giorno.

La dimensione dei mini-batch è di 10 elementi con dei fold ognuno da 31 elementi, per un totale di 124 sequenze nel dataset. Da notare che il numero di sequenze è minore rispetto al caso precedente.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 89% con 1,14% di deviazione standard. L’errore di classificazione è approssimativamente del 11%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑16: Confusion matrix per predizioni con anticpo di 3 giorni e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 27.2 ± 1.5 | 3.8 ± 1.5 |
| 1 | 9.9 ± 1.5 | 83.1 ± 1.5 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑16 evidenzia che sulla totalità del dataset mediamente 9,9 elementi vengono classificati erroneamente come patologici e 3,8 elementi vengono classificati erroneamente come sani. Inoltre, le deviazioni standard per la classificazione degli elementi patologici e la classificazione degli elementi sani sono approssimativamente uguali.

#### Previsione a 7 giorni con sequenze lunghe 3 giorni

In questo caso manteniamo gli stessi parametri e impostiamo l’intervallo di previsione uguale a 7 giorni.

La dimensione dei mini-batch è di 15 elementi con dei fold ognuno da 45 elementi, per un totale di 180 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 85% con 1,02% di deviazione standard. L’errore di classificazione è approssimativamente del 15%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑17: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 32.4 ± 1.4 | 12.6 ± 1.4 |
| 1 | 13.8 ± 1.3 | 121.2 ± 1.3 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑17 evidenzia che sulla totalità del dataset mediamente 13,8 elementi vengono classificati erroneamente come patologici e 12,6 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi patologici.

#### Previsione a 7 giorni con sequenze lunghe 1 giorno

In questo caso manteniamo gli stessi parametri del paragrafo precedente impostando però la lunghezza delle sequenze di 1 giorno.

La dimensione dei mini-batch è di 10 elementi con dei fold ognuno da 30 elementi, per un totale di 120 sequenze nel dataset.

I risultati che otteniamo sono i seguenti:

Otteniamo un’accuratezza approssimativa del 83% con 1,36% di deviazione standard. L’errore di classificazione è approssimativamente del 17%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑18: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 18.1 ± 1.1 | 11.9 ± 1.1 |
| 1 | 8.1 ± 1.3 | 81.9 ± 1.3 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑18 evidenzia che sulla totalità del dataset mediamente 8,1 elementi vengono classificati erroneamente come patologici e 11,9 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi sani.

### Confronto delle accuratezze in base alla durata delle sequenze

A seguire le accuratezze ottenute con la combinazione di tensione della cella minima e potenza del pannello:

Tabella 6‑19: Tabella delle accuratezze ottenute con la tensione della cella minima e la potenza del pannello

|  |  |  |
| --- | --- | --- |
|  | Sequenze da 3 giorni | Sequenze da 1 giorno |
| Previsione ad 1 giorno | 94.06% ± 0.66% | 96.86% ± 1.36% |
| Previsione a 3 giorni | 91.30% ± 0.99% | 91.33% ± 1.44% |
| Previsione a 7 giorni | 82.01% ± 1.39% | 84.11% ± 1.14% |

A seguire un diagramma a barre che rappresenta le accuratezze in Tabella 6‑19:

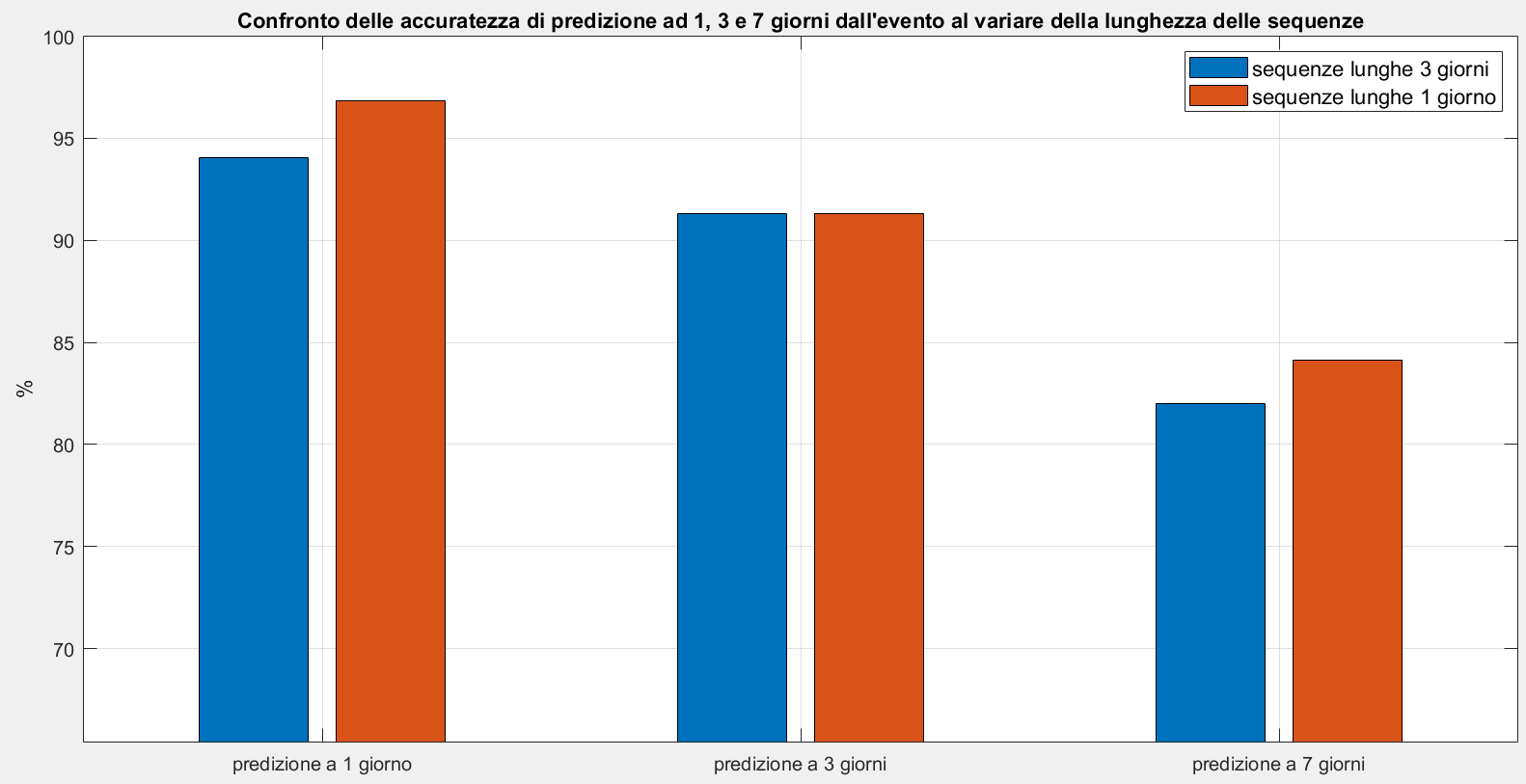


Figura 6‑1: Confronto delle accuratezze ad 1, 3 e 7 giorni dall'evento al variare della lunghezza delle sequenze. Variabili usate: tensione della cella minima e potenza del pannello

Osserviamo come l’accuratezza decresca a mano a mano che il numero di giorni dall’evento aumenta. L’accuratezza migliore in assoluto è uguale al 96,86% e viene ottenuta nel caso di previsione ad 1 giorno con sequenze lunghe 1 giorno. Il caso peggiore invece corrisponde a sequenze lunghe 3 giorni a distanza di 7 giorni dall’evento ed è 82,01%. Nel caso di previsione ad 1 giorno dell’evento l’accuratezza per sequenze lunghe 1 giorno è maggiore di quella per sequenze lunghe 3 giorni del 2,8%. A 3 giorni dall’evento l’accuratezza nel caso di sequenze lunghe 3 giorni è pressoché identica a quella di sequenze lunghe 1 giorno. A 7 giorni dall’evento, l’accuratezza per sequenze lunghe 1 giorno è maggiore dell’accuratezza per sequenze lunghe 3 giorni del 2,1%.

Andiamo ora a vedere le accuratezze ottenute con la combinazione di tensione della cella minima, potenza del pannello, SOC e irraggiamento:

Tabella 6‑20: Tabella delle accuratezze ottenute con la tensione della cella minima, la potenza del pannello, il SOC e l’irraggiamento

|  |  |  |
| --- | --- | --- |
|  | Sequenze da 3 giorni | Sequenze da 1 giorno |
| Previsione ad 1 giorno | 94.68% ± 1.12% | 97.06% ± 1.59% |
| Previsione a 3 giorni | 91.33% ± 1.49% | 90.56% ± 1.32% |
| Previsione a 7 giorni | 82.17% ± 0.96% | 81.83% ± 1.51% |

A seguire un diagramma a barre che rappresenta le accuratezze in Tabella 6‑20:

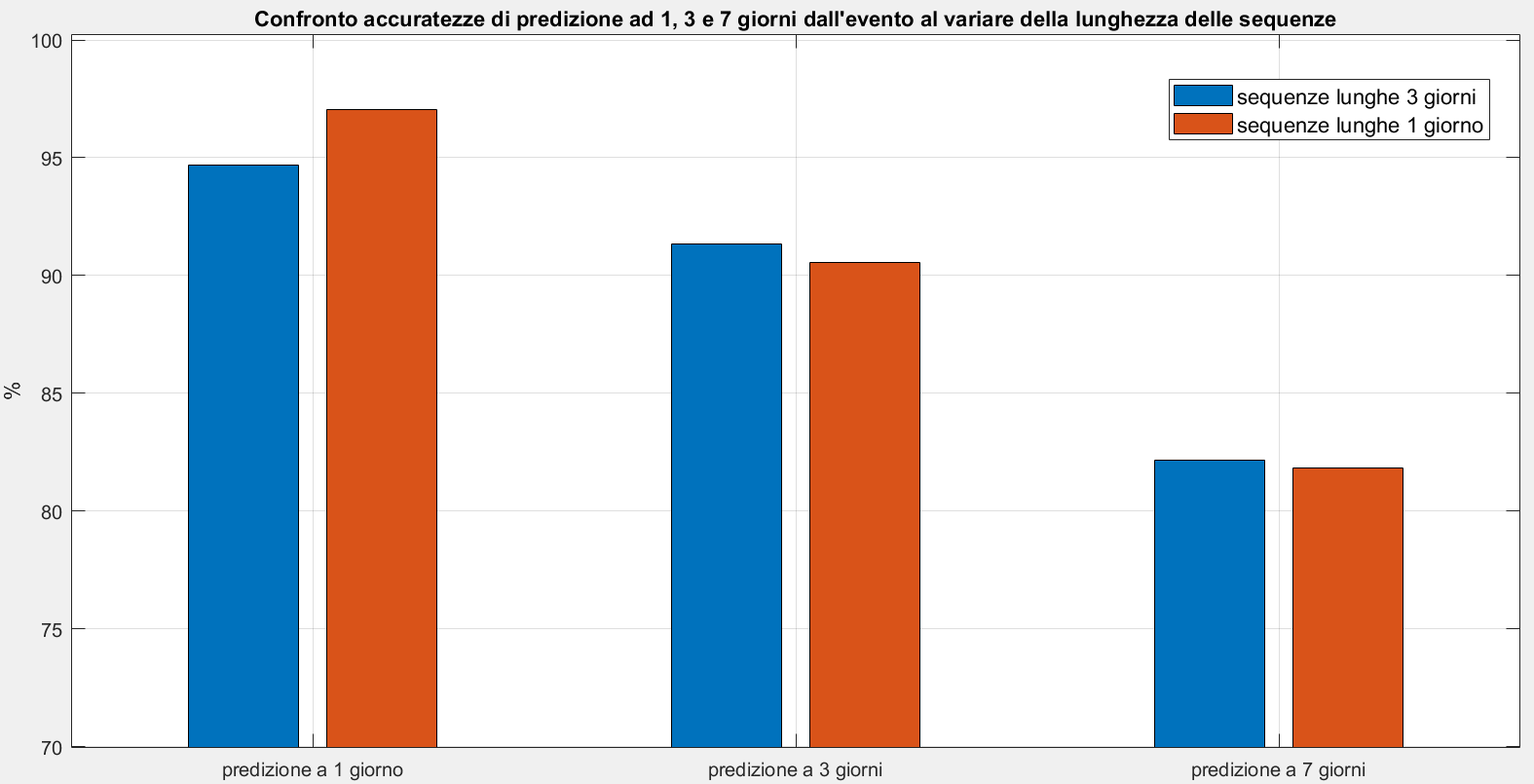


Figura 6‑2: Confronto delle accuratezze ad 1, 3 e 7 giorni dall'evento al variare della lunghezza delle sequenze. Variabili usate: tensione della cella minima, potenza del pannello, SOC e irraggiamento

Osserviamo come l’accuratezza decresca a mano a mano che il numero di giorni dall’evento aumenta. L’accuratezza migliore in assoluto è uguale al 97,06% e viene ottenuta nel caso di previsione ad 1 giorno con sequenze lunghe 1 giorno. Il caso peggiore invece corrisponde a sequenze lunghe 1 giorno a distanza di 7 giorni dall’evento ed è 81,83%. Nel caso di previsione ad 1 giorno dell’evento l’accuratezza per sequenze lunghe 1 giorno è maggiore di quella per sequenze lunghe 3 giorni del 2,38%. Inoltre, notiamo che a 3 giorni e a 7 giorni dall’evento le accuratezze nel caso di sequenze lunghe 3 giorni sono pressocché uguali a quelle ad 1 giorno.

Andiamo ora a vedere le accuratezze ottenute con la combinazione di tensione della cella minima, potenza del pannello, SOC, irraggiamento e bilancio delle correnti del pacco batterie:

Tabella 6‑21: Tabella delle accuratezze ottenute con la tensione della cella minima, la potenza del pannello, il SOC, l’irraggiamento e la corrente della batteria

|  |  |  |
| --- | --- | --- |
|  | Sequenze da 3 giorni | Sequenze da 1 giorno |
| Previsione ad 1 giorno | 93.40% ± 0.76% | 97.65% ± 0.83% |
| Previsione a 3 giorni | 90.28% ± 1.29% | 88.95% ± 1.14% |
| Previsione a 7 giorni | 85.33% ± 1.02% | 83.33% ± 1.36% |

A seguire un diagramma a barre che riporta le accuratezze in Tabella 6‑21:

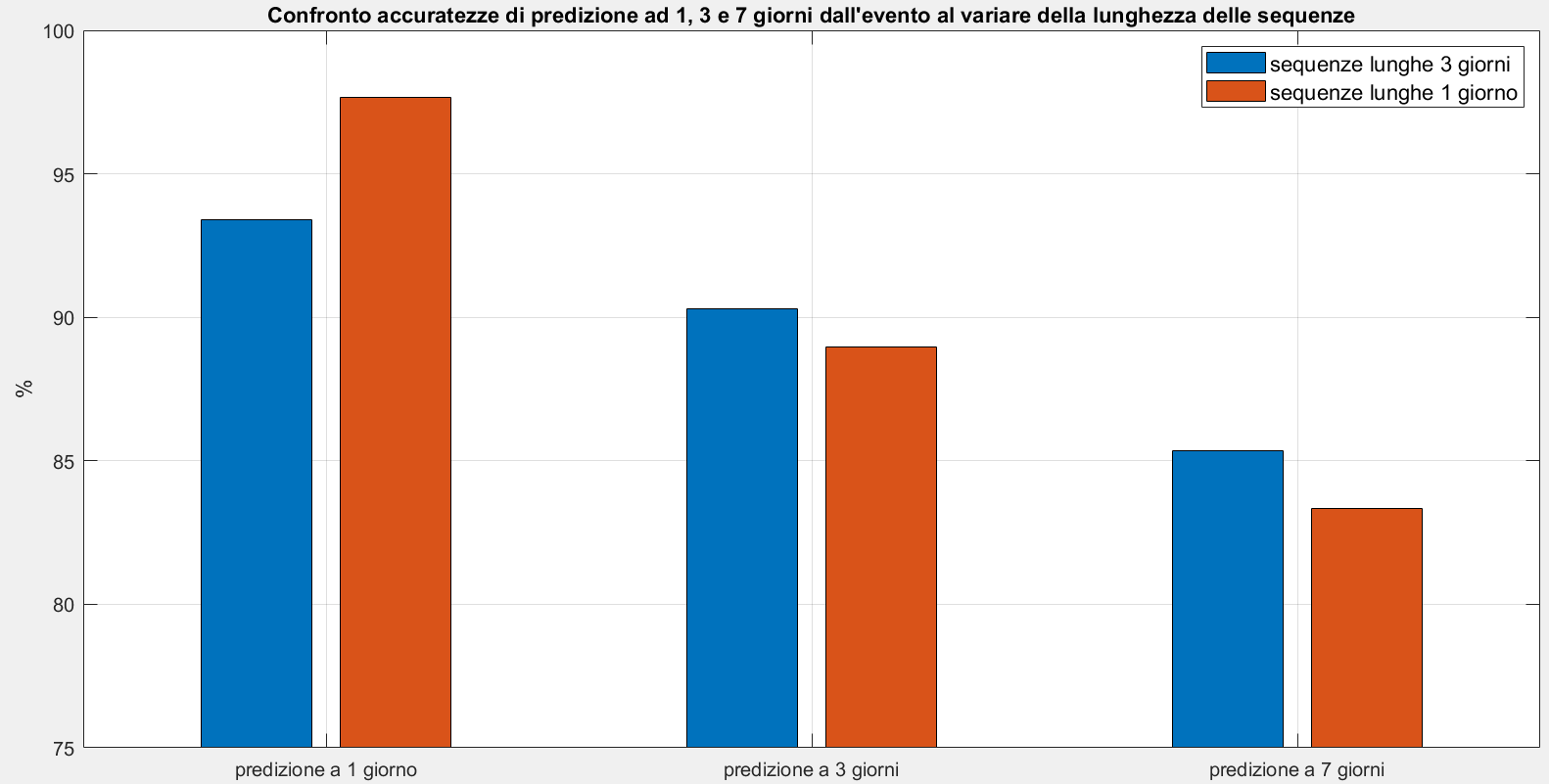


Figura 6‑3: Confronto delle accuratezze ad 1, 3 e 7 giorni dall'evento al variare della lunghezza delle sequenze. Variabili usate: tensione della cella minima, potenza del pannello, SOC, irraggiamento e bilancio delle correnti della batteria

Osserviamo come l’accuratezza decresca a mano a mano che il numero di giorni dall’evento aumenta. L’accuratezza migliore in assoluto è uguale al 97,65% e viene ottenuta nel caso di previsione ad 1 giorno con sequenze lunghe 1 giorno. Il caso peggiore invece corrisponde a sequenze lunghe 1 giorno a distanza di 7 giorni dall’evento ed è 83,33%. Nel caso di previsione ad 1 giorno dell’evento l’accuratezza per sequenze lunghe 1 giorno è maggiore di quella per sequenze lunghe 3 giorni del 4,24%. Inoltre, notiamo che a 3 giorni e a 7 giorni dall’evento le accuratezze nel caso di sequenze lunghe 3 giorni sono maggiori di quelle ad 1 giorno rispettivamente del 1,3% e del 2%.

Globalmente, osserviamo che a distanza di 1 giorno dall’evento le sequenze della durata di 1 giorno hanno un’accuratezza migliore rispetto a quelle di una durata di 3 giorni. Ciò non è vero quando ci poniamo ad una distanza di 3 o 7 giorni dall’evento. In questi casi sequenze della durata di 3 giorni possono avere risultati migliori in base alla combinazione di variabili che scegliamo di dare in ingresso alla rete neurale.

### Confronto delle accuratezze in base alla combinazione di grandezze

Andiamo ora a confrontare l’andamento delle accuratezze in base alla combinazione di dati per determinare quale dia risultati migliori. In Figura 6‑4 e in Figura 6‑5 vediamo dei diagrammi a barre. Le barre blu rappresentano le accuratezze ottenute quando in ingresso alla rete diamo la tensione della cella minima e la potenza del pannello. Le barre arancioni corrispondono alle accuratezze considerando la tensione della cella minima, la potenza del pannello, il SOC e l’irraggiamento. Infine, le barre gialle corrispondono alle accuratezze quando in ingresso alla rete diamo la tensione della cella minima, la potenza del pannello, il SOC, l’irraggiamento e il bilancio delle correnti delle batterie. Sulle ascisse troviamo l’anticipo, in giorni, con il quale si vuole predire l’evento.

In Tabella 6‑22 riportiamo le accuratezze al variare dell’intervallo di previsione. Le intestazioni delle colonne sono il colore delle barre in Figura 6‑4:

Tabella 6‑22: Accuratezze al variare dell'intervallo di previsione con varie combinazioni di variabili e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
|  | Blu | Arancione | Giallo |
| Previsione ad 1 giorno | 96.86% ± 1.36% | 97.06% ± 1.59% | 97.65% ± 0.83% |
| Previsione a 3 giorni | 91.33% ± 1.44% | 90.56% ± 1.32% | 88.95% ± 1.14% |
| Previsione a 7 giorni | 84.11% ± 1.14% | 81.83% ± 1.51% | 83.33% ± 1.36% |

Grafichiamo le accuratezze relative alle sequenze lunghe 1 giorno:

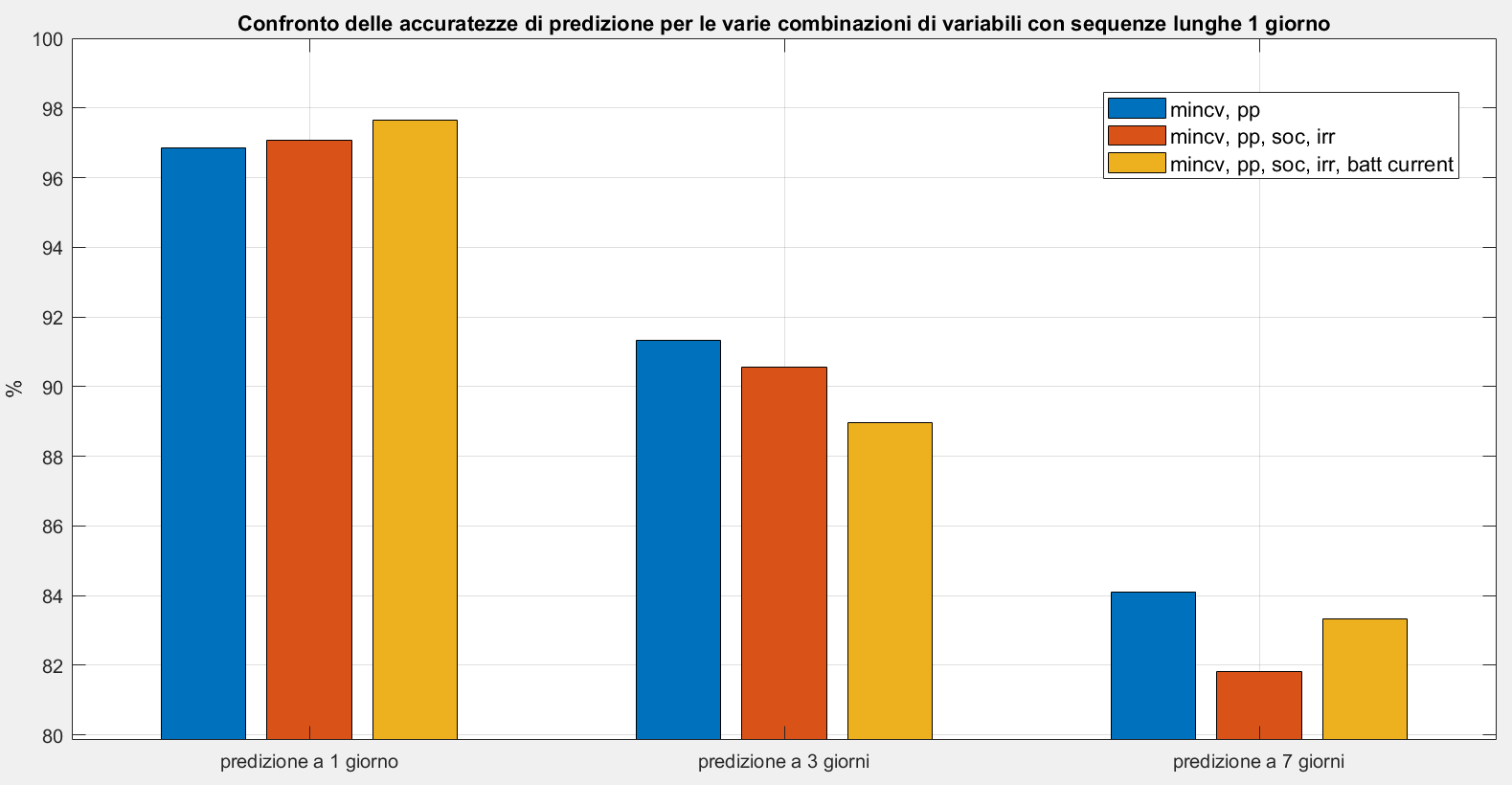


Figura 6‑4: Confronto delle accuratezze di previsione per le varie combinazioni di variabili con sequenze lunghe 1 giorno

Le accuratezze in Figura 6‑4 si riferiscono al caso in cui le sequenze abbiano una durata di 24 ore. Constatiamo che, con un intervallo di previsione di 1 giorno i risultati migliori di accuratezza sono rappresentati dalla barra gialla con un’accuratezza del 97,65%. Per un intervallo di previsione d 3 giorni l’accuratezza migliore è rappresentata dalla barra blu ed è del 91,33%. Per predizioni a 7 giorni l’accuratezza migliore è data dalla barra blu e vale 84,11%.

In Tabella 6‑23 riportiamo le accuratezze al variare dell’intervallo di previsione. Le intestazioni delle colonne sono il colore delle barre in Figura 6‑5 :

Tabella 6‑23: Accuratezze al variare dell'intervallo di previsione con varie combinazioni di variabili e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
|  | Blu | Arancione | Giallo |
| Previsione ad 1 giorno | 94.06% ± 0.66% | 94.68% ± 1.12% | 93.40% ± 0.76% |
| Previsione a 3 giorni | 91.30% ± 0.99% | 91.33% ± 1.49% | 90.28% ± 1.29% |
| Previsione a 7 giorni | 82.01% ± 1.39% | 82.17% ± 0.96% | 85.33% ± 1.02% |

Grafichiamo le accuratezze relative alle sequenze lunghe 1 giorno:

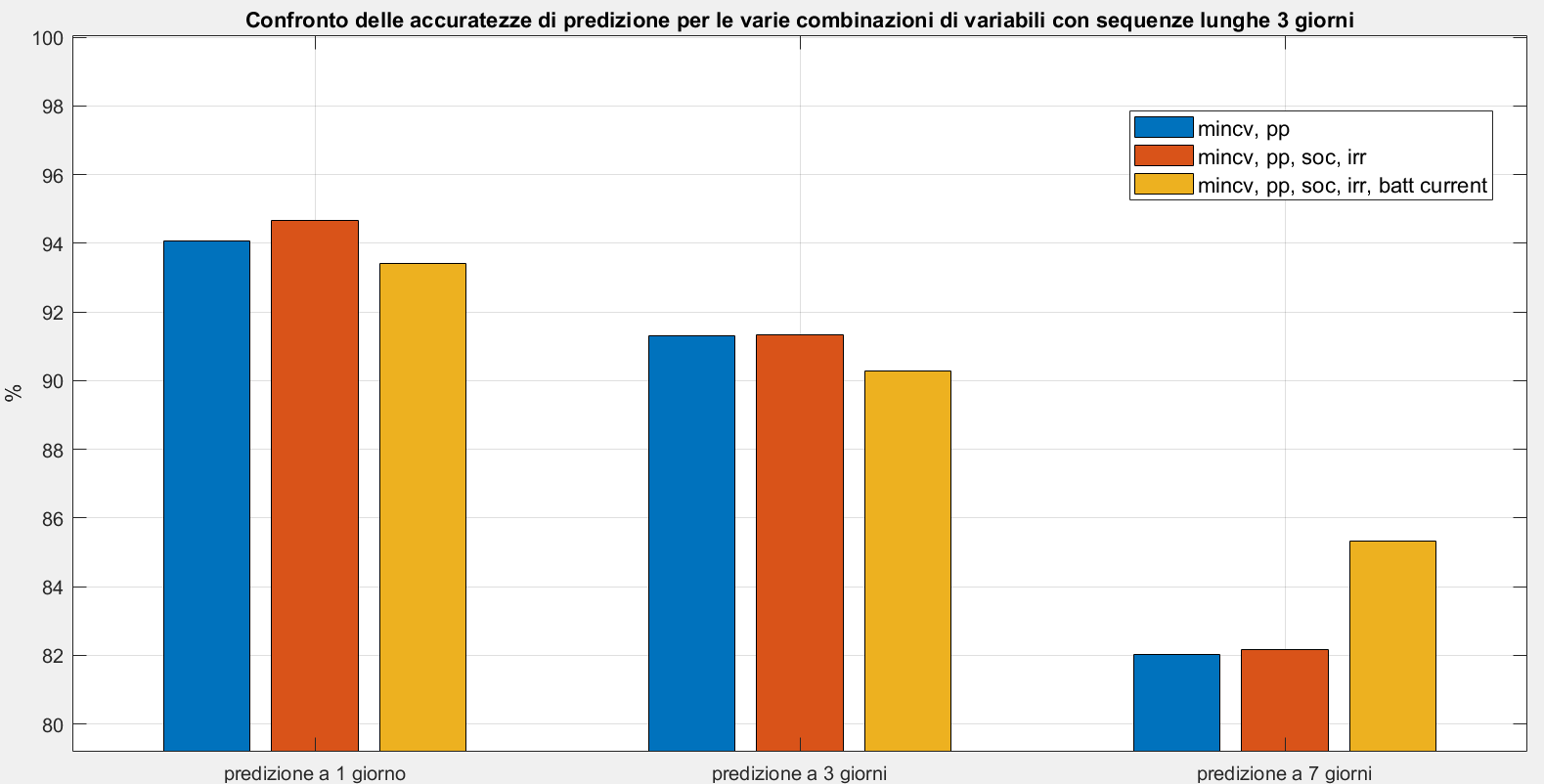


Figura 6‑5: Confronto delle accuratezze di previsione per le varie combinazioni di variabili con sequenze lunghe 3 giorni

Le accuratezze in Figura 6‑5 si riferiscono al caso in cui le sequenze abbiano una durata di 72 ore. Osserviamo che, con un intervallo di previsione di 3 giorni i risultati migliori di accuratezza sono rappresentati dalla barra arancione con un’accuratezza del 94,68%. Per un intervallo di previsione d 3 giorni l’accuratezza migliore è rappresentata dalla barra blu e dalla barra arancione ed è del 91,3%. Per predizioni a 7 giorni l’accuratezza migliore è data dalla barra gialla e vale 85,33%.

Facendo riferimento a Figura 6‑5 e a Figura 6‑4, constatiamo che, per un dato intervallo di previsione, non emerge una combinazione di variabili che dia in assoluto i migliori risultati in termini di accuratezza, in quanto i risultati dipendono anche dalla lunghezza in giorni delle sequenze. Globalmente possiamo osservare, per le predizioni ad 1 giorno, delle accuratezze maggiori nel caso di sequenze lunghe 1 giorno rispetto a quelle lunghe 3 giorni. Indipendentemente dalla durata delle sequenze, le accuratezze in questo intervallo di previsione rimangono sempre superiori al 93% fino ad un massimo del 98% approssimativamente. Nell’intervallo di previsione di 3 giorni, le accuratezze vanno da un minimo del 89% fino ad un massimo del 91%. Nell’intervallo di previsione di 7 giorni, le accuratezze vanno da un minimo del 82% fino ad un massimo del 85%.

## Prima rete alternativa

In questo paragrafo andremmo a vedere i risultati ottenuti con una rete più complessa di quella usata nel paragrafo 6.1. Facendo questo tentativo, vogliamo osservare il comportamento della rete, e valutare se ha maggiore capacità di estrarre correlazioni tra le misurazioni date in ingresso. L’obbiettivo è di avere un modello più robusto e con una maggiore affidabilità. Le prove verranno fatte con le stesse combinazioni di variabili usate nel paragrafo 6.1 e i risultati verranno presentati nello stesso modo. I parametri della rete rimarranno quelli impostati in Tabella 5‑1 e Tabella 5‑2.

La rete in questione, è composta da 2 layer BiLSTM consecutivi, seguiti da un layer fully connected con funzione di attivazione softmax, e da un classification layer. La scelta di aggiungere un altro layer BiLSTM, è motivata dal fatto che, vogliamo aumentare la capacità della rete di estrarre correlazioni tra i campioni delle sequenze. Rendendo la rete maggiormente complessa, speriamo di poter ottenere dei parametri, in grado di dare una capacità di generalizzazione maggiore alla rete. A fronte di una maggiore complessità, otterremo dei tempi di apprendimento del modello maggiori (vedi appendice in Figura 8‑20 per il codice).

### Tensione della cella minima e potenza del pannello

#### Previsione ad 1 giorno con sequenze lunghe 3 giorni

Poniamoci nel caso in cui le sequenze durino 3 giorni, siano sfalsate di 1 giorno, e ci sia una proporzione di sequenze tale che per una sequenza patologica ce ne siano 3 sane. Vogliamo prevedere l’evento 1 giorno prima dell’evento. La dimensione dei mini-batch è di 12 elementi con dei fold ognuno da 48 elementi, per un totale di 192 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza del 94% con 0,78% di deviazione standard. L’errore di classificazione è approssimativamente del 6%.

Per quanto riguarda la confusion matrix, le righe rappresentano le classi e sulle colonne troviamo le classi assegnate alle sequenze dal modello predittivo. La prima riga rappresenta la classe patologica e la seconda la classe sana:

Tabella 6‑24: Confusion Matrix per predizioni con anticpo di 1 giorno e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 43.4 ± 1.8 | 4.6 ± 1.8 |
| 1 | 7.8 ± 0.8 | 136.2 ± 0.8 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑24 evidenzia come sulla totalità del dataset mediamente 7,8 elementi vengono classificati erroneamente come patologici e 4,6 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi patologici.

#### Previsione ad 1 giorno con sequenze lunghe 1 giorno

Poniamoci nelle stesse condizioni del caso precedente ma con sequenze di durata 1 giorno. La dimensione dei mini-batch è di 7 elementi con dei fold ognuno da 35 elementi, per un totale di 140 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza del 95% con 2,13% di deviazione standard. L’errore di classificazione è approssimativamente del 5%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑25: Confusion matrix per predizioni con anticipo di 1 giorno e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 32.8 ± 2.1 | 2.2 ± 2.1 |
| 1 | 5.4 ± 1.7 | 99.6 ± 1.7 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑25 evidenzia che sulla totalità del dataset mediamente 5,4 elementi vengono classificati erroneamente come patologici e 2,2 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi patologici.

#### Previsione a 3 giorni con sequenze lunghe 3 giorni

In questo caso manteniamo gli stessi parametri ma impostiamo sequenze lunghe 3 giorni e intervallo di previsione uguale a 3 giorni.

La dimensione dei mini-batch è di 23 elementi con dei fold ognuno da 46 elementi, per un totale di 184 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 90% con 1,26% di deviazione standard. L’errore di classificazione è approssimativamente del 10%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑26: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 39.9 ± 4.3 | 6.1 ± 4.3 |
| 1 | 11.9 ± 2.9 | 126.1 ± 2.9 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑26 evidenzia che sulla totalità del dataset mediamente 11,9 elementi vengono classificati erroneamente come patologici e 6,1 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi patologici.

#### Previsione a 3 giorni con sequenze lunghe 1 giorno

In questo caso manteniamo gli stessi parametri ma impostiamo sequenze lunghe 1 giorno.

La dimensione dei mini-batch è di 16 elementi con dei fold ognuno da 32 elementi, per un totale di 128 sequenze nel dataset. Da notare che il numero di sequenze è minore rispetto al caso precedente.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 91% con 1,27% di deviazione standard. L’errore di classificazione è approssimativamente del 9%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑27: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 26.0 ± 2.6 | 6.0 ± 2.6 |
| 1 | 6.2 ± 1.8 | 89.8 ± 1.8 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑27 evidenzia che sulla totalità del dataset mediamente 6,2 elementi vengono classificati erroneamente come patologici e 6,0 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi patologici.

#### Previsione a 7 giorni con sequenze lunghe 3 giorni

In questo caso manteniamo gli stessi parametri e impostiamo l’intervallo di previsione uguale a 7 giorni,

La dimensione dei mini-batch è di 23 elementi con dei fold ognuno da 46 elementi, per un totale di 184 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 82% con 1,58% di deviazione standard. L’errore di classificazione è approssimativamente del 18%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑28: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 29.8 ± 4.4 | 16.2 ± 4.4 |
| 1 | 17.5 ± 3.0 | 120.5 ± 3.0 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑28 evidenzia che sulla totalità del dataset mediamente 17,5 elementi vengono classificati erroneamente come patologici e 16,2 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi patologici.

#### Previsione a 7 giorni con sequenze lunghe 1 giorno

In questo caso manteniamo gli stessi parametri del paragrafo precedente impostando però la lunghezza delle sequenze di 1 giorno.

La dimensione dei mini-batch è di 10 elementi con dei fold ognuno da 31 elementi, per un totale di 124 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 84% con 1,26% di deviazione standard. L’errore di classificazione è approssimativamente del 16%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑29: Confusion matrixper predizioni con anticipo di 7 giorni e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 18.6 ± 2.5 | 12.4 ± 2.5 |
| 1 | 7.3 ± 1.5 | 85.7 ± 1.5 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑29 evidenzia che sulla totalità del dataset mediamente 7,3 elementi vengono classificati erroneamente come patologici e 12,4 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi patologici.

### Tensione della cella minima, potenza del pannello, SOC e irraggiamento

#### Previsione ad 1 giorno con sequenze lunghe 3 giorni

Poniamoci nel caso in cui le sequenze durino 3 giorni, siano sfalsate di 1 giorno, e ci sia una proporzione di sequenze tale che per una sequenza patologica ce ne siano 3 sane.

La dimensione dei mini-batch è di 12 elementi con dei fold ognuno da 47 elementi, per un totale di 188 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 94% con 2.14% di deviazione standard. L’errore di classificazione è approssimativamente del 6%.

Per quanto riguarda la matrice di confusione:

Tabella 6‑30: Confusion matrix per predizioni con anticipo di 1 giorno e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 42.6 ± 2.8 | 4.4 ± 2.8 |
| 1 | 6.4 ± 3.3 | 134.6 ± 3.3 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑30 evidenzia come sulla totalità del dataset mediamente 6,4 elementi vengono classificati erroneamente come patologici e 4,4 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi sani.

#### Previsione ad 1 giorno con sequenze lunghe 1 giorno

Poniamoci nelle stesse condizioni del caso precedente ma con sequenze di durata 1 giorno.

La dimensione dei mini-batch è di 17 elementi con dei fold ognuno da 34 elementi, per un totale di 136 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza del 97% con 1,26% di deviazione standard. L’errore di classificazione è approssimativamente del 3%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑31: Confusion matrix per predizioni con anticipo di 1 giorno e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 33.1 ± 1.0 | 0.9 ± 1.0 |
| 1 | 3.6 ± 1.5 | 98.4 ± 1.5 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑31 evidenzia che sulla totalità del dataset mediamente 3,6 elementi vengono classificati erroneamente come patologici e 0,9 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi sani.

#### Previsione a 3 giorni con sequenze lunghe 3 giorni

In questo caso manteniamo gli stessi parametri ma impostiamo sequenze lunghe 3 giorni e intervallo di previsione uguale a 3 giorni.

La dimensione dei mini-batch è di 15 elementi con dei fold ognuno da 45 elementi, per un totale di 180 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 90% con 1,56% di deviazione standard. L’errore di classificazione è approssimativamente del 10%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑32: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 36.7 ± 2.5 | 8.3 ± 2.5 |
| 1 | 9.6 ± 3.1 | 125.4 ± 3.1 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑32 evidenzia come sulla totalità del dataset mediamente 9,6 elementi vengono classificati erroneamente come patologici e 8,3 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi sani.

#### Previsione a 3 giorni con sequenze lunghe 1 giorno

In questo caso manteniamo gli stessi parametri ma impostiamo sequenze lunghe 1 giorno.

La dimensione dei mini-batch è di 10 elementi con dei fold ognuno da 31 elementi, per un totale di 124 sequenze nel dataset. Da notare che il numero di sequenze è minore rispetto al caso precedente.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 90% con 1,87% di deviazione standard. L’errore di classificazione è approssimativamente del 10%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑33: Confusion matrix per predizioni con anticpo di 3 giorni e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 26.6 ± 1.8 | 4.4 ± 1.8 |
| 1 | 8.2 ± 1.9 | 84.8 ± 1.9 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑33 evidenzia che sulla totalità del dataset mediamente 8,2 elementi vengono classificati erroneamente come patologici e 4,4 elementi vengono classificati erroneamente come sani. Inoltre, le deviazioni standard per la classificazione degli elementi sani e patologici sono approssimativamente uguali.

#### Previsione a 7 giorni con sequenze lunghe 3 giorni

In questo caso impostiamo l’intervallo di previsione uguale a 7 giorni e sequenze lunghe 3 giorni.

La dimensione dei mini-batch è di 15 elementi con dei fold ognuno da 45 elementi, per un totale di 180 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 81% con 1,76% di deviazione standard. L’errore di classificazione è approssimativamente del 19%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑34: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 30.0 ± 4.0 | 15.0 ± 4.0 |
| 1 | 18.6 ± 4.3 | 116.4 ± 4.3 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑40 evidenzia che sulla totalità del dataset mediamente 18,6 elementi vengono classificati erroneamente come patologici e 15,0 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi sani.

#### Previsione a 7 giorni con sequenze lunghe 1 giorno

In questo caso manteniamo gli stessi parametri del paragrafo precedente impostando però la lunghezza delle sequenze di 1 giorno.

La dimensione dei mini-batch è di 10 elementi con dei fold ognuno da 30 elementi, per un totale di 120 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 81% con 1,80% di deviazione standard. L’errore di classificazione è approssimativamente del 19%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑35: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 18.2 ± 3.6 | 11.8 ± 3.6 |
| 1 | 11.5 ± 3.3 | 78.5 ± 3.3 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑35 evidenzia che sulla totalità del dataset mediamente 11,5 elementi vengono classificati erroneamente come patologici e 11,8 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi patologici.

### Tensione della cella minima, potenza del pannello, SOC, irraggiamento e bilancio delle correnti della batteria

#### Previsione ad 1 giorno con sequenze lunghe 3 giorni

Poniamoci nel caso in cui le sequenze durino 3 giorni, siano sfalsate di 1 giorno, e ci sia una proporzione di sequenze tale che per una sequenza patologica ce ne siano 3 sane.

La dimensione dei mini-batch è di 12 elementi con dei fold ognuno da 47 elementi, per un totale di 188 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 93% con % di deviazione standard. L’errore di classificazione è approssimativamente del 7%.

Per quanto riguarda la matrice di confusione:

Tabella 6‑36: Confusion matrix per predizioni con anticipo di 1 giorno e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 41.7 ± 1.8 | 5.3 ± 1.8 |
| 1 | 7.6 ± 1.6 | 133.4 ± 1.6 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑36 evidenzia che sulla totalità del dataset mediamente 7,6 elementi vengono classificati erroneamente come patologici e 5,3 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard per la classificazione degli elementi patologici e sani è approssimativamente uguale.

#### Previsione ad 1 giorno con sequenze lunghe 1 giorno

Poniamoci nelle stesse condizioni del caso precedente ma con sequenze di durata 1 giorno.

La dimensione dei mini-batch è di 17 elementi con dei fold ognuno da 34 elementi, per un totale di 136 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza del 98% con 1,47% di deviazione standard. L’errore di classificazione è approssimativamente del 2%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑37: Confusion matrix per predizioni con anticipo di 1 giorno e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 33.2 ± 1.3 | 0.8 ± 1.3 |
| 1 | 2.2 ± 1.0 | 99.8 ± 1.0 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑37 evidenzia che sulla totalità del dataset mediamente 2,2 elementi vengono classificati erroneamente come patologici e 0,8 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi patologici.

#### Previsione a 3 giorni con sequenze lunghe 3 giorni

In questo caso manteniamo gli stessi parametri ma impostiamo sequenze lunghe 3 giorni e intervallo di previsione uguale a 3 giorni.

La dimensione dei mini-batch è di 15 elementi con dei fold ognuno da 45 elementi, per un totale di 180 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 90% con 1,45% di deviazione standard. L’errore di classificazione è approssimativamente del 10%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑38: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 37.3 ± 4.0 | 7.7 ± 4.0 |
| 1 | 11.2 ± 2.7 | 123.8 ± 2.7 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑38 evidenzia che sulla totalità del dataset mediamente 11,2 elementi vengono classificati erroneamente come patologici e 7,7 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi patologici.

#### Previsione a 3 giorni con sequenze lunghe 1 giorno

In questo caso manteniamo gli stessi parametri ma impostiamo sequenze lunghe 1 giorno.

La dimensione dei mini-batch è di 10 elementi con dei fold ognuno da 31 elementi, per un totale di 124 sequenze nel dataset. Da notare che il numero di sequenze è minore rispetto al caso precedente.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 88% con 2,72% di deviazione standard. L’errore di classificazione è approssimativamente del 12%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑39: Confusion matrix per predizioni con anticpo di 3 giorni e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 28.0 ± 1.3 | 3.0 ± 1.3 |
| 1 | 12.4 ± 3.3 | 80.7 ± 3.3 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑39 evidenzia che sulla totalità del dataset mediamente 12,4 elementi vengono classificati erroneamente come patologici e 3,0 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard per la classificazione degli elementi sani è maggiore.

#### Previsione a 7 giorni con sequenze lunghe 3 giorni

In questo caso impostiamo l’intervallo di previsione uguale a 7 giorni e sequenze lunghe 3 giorni.

La dimensione dei mini-batch è di 15 elementi con dei fold ognuno da 45 elementi, per un totale di 180 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 85% con 1,90% di deviazione standard. L’errore di classificazione è approssimativamente del 15%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑40: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 32.5 ± 2.1 | 12.5 ± 2.1 |
| 1 | 14.7 ± 3.7 | 120.3 ± 3.7 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑40 evidenzia che sulla totalità del dataset mediamente 14,7 elementi vengono classificati erroneamente come patologici e 12,5 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi sani.

#### Previsione a 7 giorni con sequenze lunghe 1 giorno

In questo caso manteniamo gli stessi parametri del paragrafo precedente impostando però la lunghezza delle sequenze di 1 giorno.

La dimensione dei mini-batch è di 10 elementi con dei fold ognuno da 30 elementi, per un totale di 120 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 82% con 2,46% di deviazione standard. L’errore di classificazione è approssimativamente del 18%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑41: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 17.9 ± 2.6 | 12.1 ± 2.6 |
| 1 | 9.7 ± 2.3 | 80.3 ± 2.3 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑41 evidenzia che sulla totalità del dataset mediamente 9,7 elementi vengono classificati erroneamente come patologici e 12,1 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi patologici.

### Confronto delle accuratezze in base alla durata delle sequenze

A seguire le accuratezze ottenute con la combinazione di tensione della minima e potenza del pannello:

Tabella 6‑42: Tabella delle accuratezze ottenute con la tensione della cella minima e la potenza del pannello

|  |  |  |
| --- | --- | --- |
|  | Sequenze da 3 giorni | Sequenze da 1 giorno |
| Previsione ad 1 giorno | 93.54% ± 0.78% | 94.57% ± 2.13% |
| Previsione a 3 giorni | 90.22% ±1.26% | 90.47% ±1.27% |
| Previsione a 7 giorni | 81.68% ±1.58% | 84.11% ±1.26% |

A seguire un diagramma a barre che rappresenta le accuratezze in Tabella 6‑42:

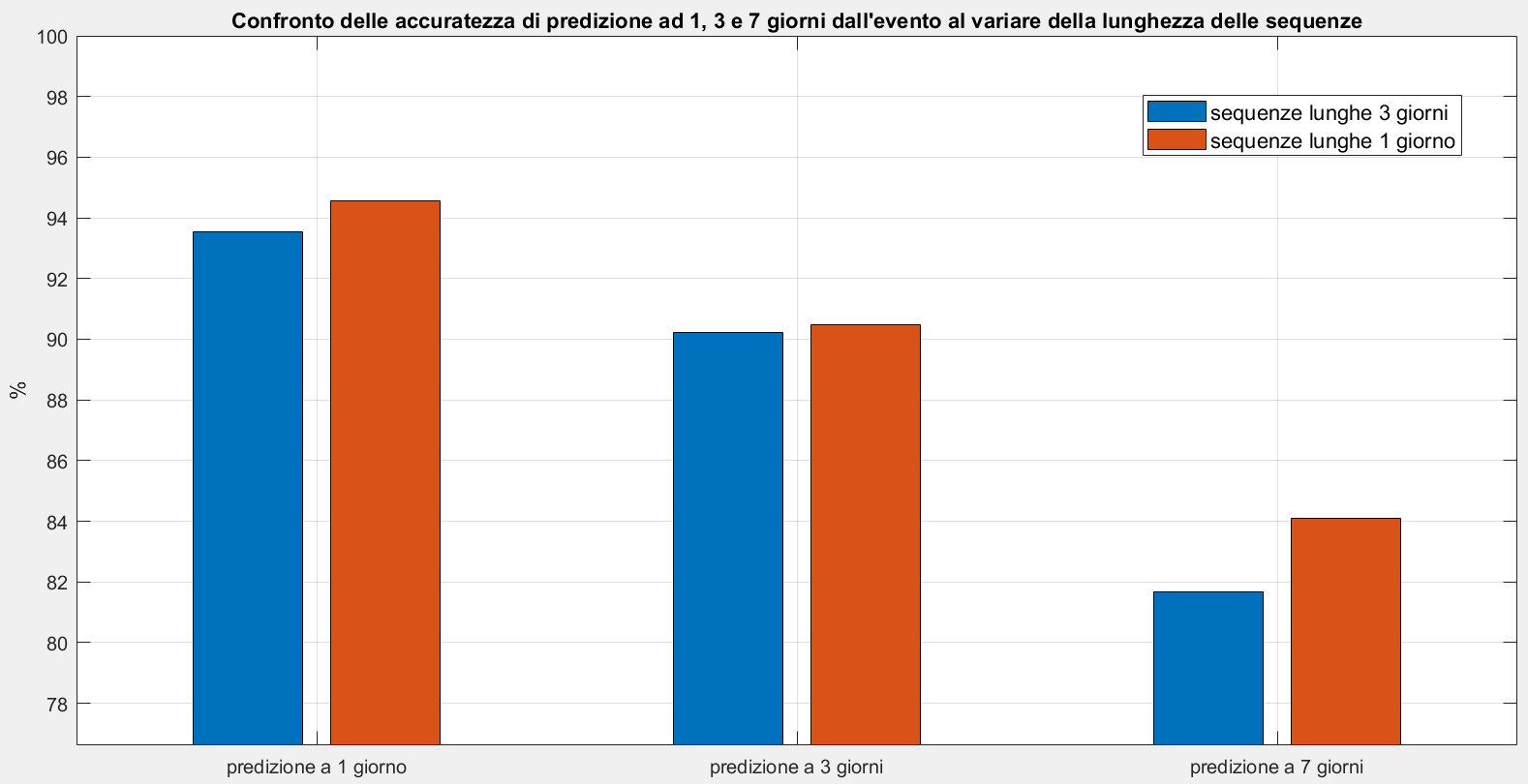


Figura 6‑6:Confronto delle accuratezze ad 1, 3 e 7 giorni dall'evento al variare della lunghezza delle sequenze. Variabili usate: tensione della cella minima e potenza del pannello

In Figura 6‑6, osserviamo come l’accuratezza decresca a mano a mano che il numero di giorni dall’evento aumenta. L’accuratezza migliore in assoluto è uguale al 94,57% e viene ottenuta nel caso di previsione ad 1 giorno con sequenze lunghe 1 giorno. Il caso peggiore invece corrisponde a sequenze lunghe 3 giorni a distanza di 7 giorni dall’evento ed è 81,68%. Nel caso di previsione ad 1 giorno dell’evento, l’accuratezza per sequenze lunghe 1 giorno è maggiore di quelle lunghe 3 giorni di 2,43%. A 3 giorni dall’evento le accuratezze nel caso di sequenze lunghe 3 giorni sono approssimativamente uguali a quelle lunghe 1 giorno. A 7 giorni dall’evento le sequenze lunghe 1 giorno producono risultati migliori del 2,43% rispetto a quelle lunghe 3 giorni.

A seguire le accuratezze ottenute con la combinazione di tensione della minima, potenza del pannello, SOC e irraggiamento:

Tabella 6‑43: Tabella delle accuratezze ottenute con la tensione della cella minima, la potenza del pannello, SOC e irraggiamento

|  |  |  |
| --- | --- | --- |
|  | Sequenze da 3 giorni | Sequenze da 1 giorno |
| Previsione ad 1 giorno | 94.26% ±2.14% | 96.69% ± 1.26% |
| Previsione a 3 giorni | 90.06% ±1.56% | 89.84% ±1.87% |
| Previsione a 7 giorni | 81.33% ±1.76% | 80.58%±1.80% |

A seguire un diagramma a barre che rappresenta le accuratezze in Tabella 6‑43:

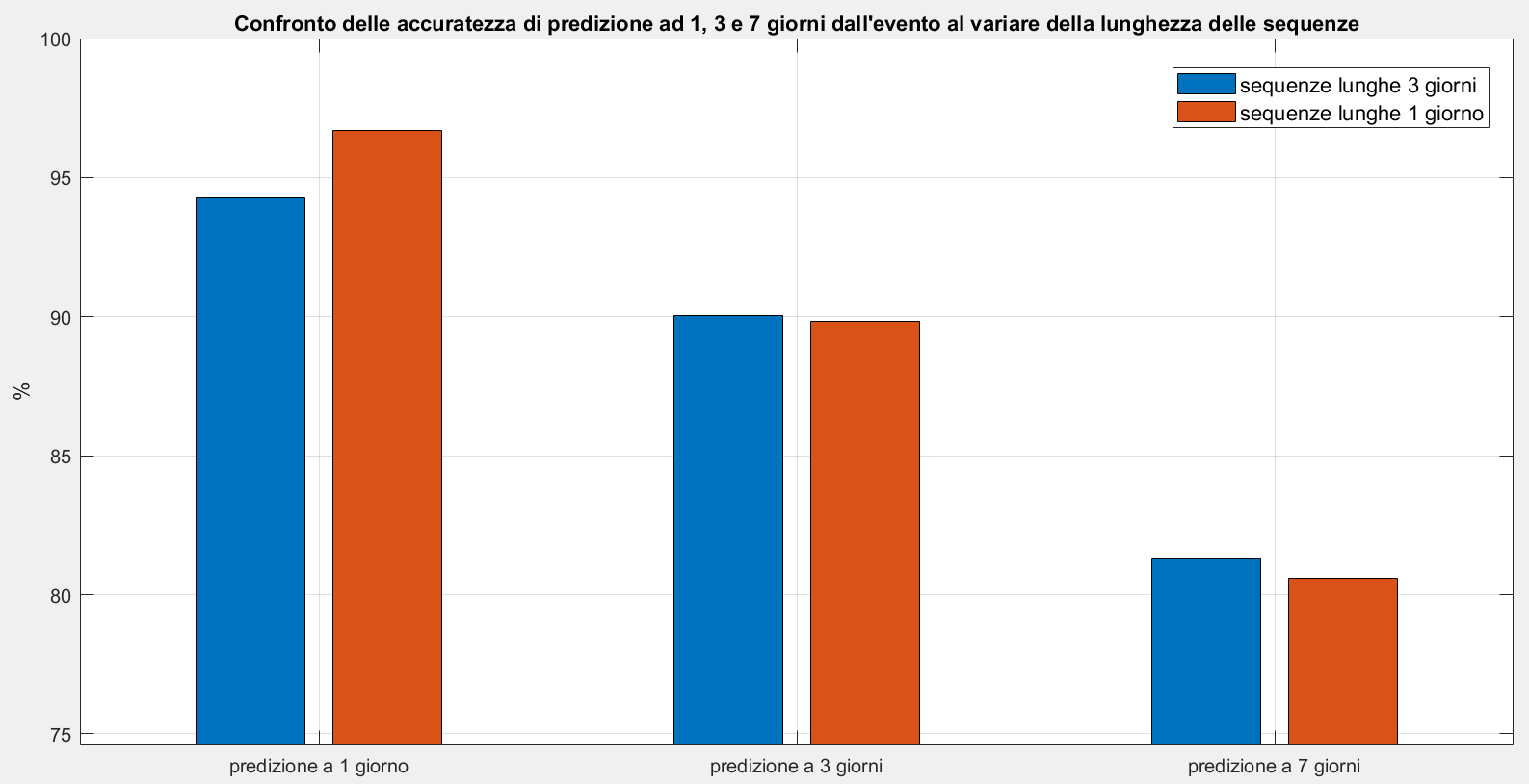


Figura 6‑7:Confronto delle accuratezze ad 1, 3 e 7 giorni dall'evento al variare della lunghezza delle sequenze. Variabili usate: tensione della cella minima, potenza del pannello, SOC e irraggiamento

In Figura 6‑7, osserviamo come l’accuratezza decresca a mano a mano che il numero di giorni dall’evento aumenta. L’accuratezza migliore in assoluto è uguale al 96,69% e viene ottenuta nel caso di previsione ad 1 giorno con sequenze lunghe 1 giorno. Il caso peggiore invece corrisponde a sequenze lunghe 1 giorno a distanza di 7 giorni dall’evento ed è 80,58%. Nel caso di previsione ad 1 giorno dall’evento, l’accuratezza per sequenze lunghe 1 giorno è maggiore di quella per sequenze lunghe 3 giorni del 2,43%. Sia a 3 giorni che a 7 giorni dall’evento le accuratezze per le sequenze lunghe 3 giorni sono approssimativamente uguali a quelle per le sequenze lunghe 1 giorno.

A seguire le accuratezze ottenute con la combinazione di tensione della minima, potenza del pannello, SOC, irraggiamento e bilancio delle correnti della batteria:

Tabella 6‑44: Tabella delle accuratezze ottenute con la tensione della cella minima, la potenza del pannello, SOC, irraggiamento e bilancio delle correnti della batteria

|  |  |  |
| --- | --- | --- |
|  | Sequenze da 3 giorni | Sequenze da 1 giorno |
| Previsione ad 1 giorno | 93.18% ± 1.41% | 97.79% ± 1.47% |
| Previsione a 3 giorni | 89.50% ± 1.45% | 87.58% ± 2.72% |
| Previsione a 7 giorni | 84.89% ± 1.90% | 81.83% ± 2.48% |

A seguire un diagramma a barre che rappresenta le accuratezze in Tabella 6‑44:

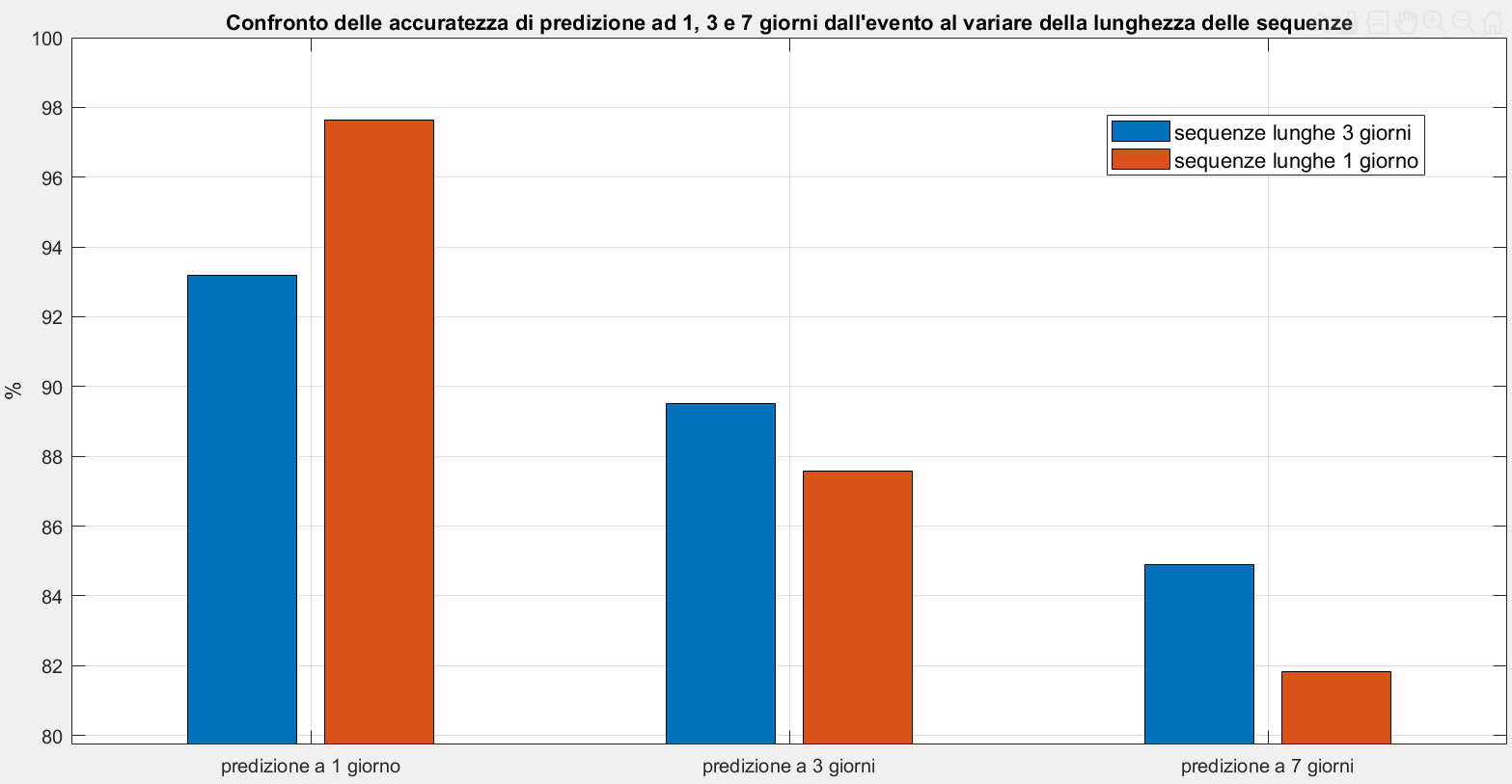


Figura 6‑8:Confronto delle accuratezze ad 1, 3 e 7 giorni dall'evento al variare della lunghezza delle sequenze. Variabili usate: tensione della cella minima, potenza del pannello, SOC, irraggiamento e bilancio delle correnti della batteria

In Figura 6‑8, osserviamo come l’accuratezza decresca a mano a mano che il numero di giorni dall’evento aumenta. L’accuratezza migliore in assoluto è uguale al 97,65% e viene ottenuta nel caso di previsione ad 1 giorno con sequenze lunghe 1 giorno. Il caso peggiore invece corrisponde a sequenze lunghe 1 giorno a distanza di 7 giorni dall’evento ed è 83,33%. Nel caso di previsione ad 1 giorno dell’evento l’accuratezza per sequenze lunghe 1 giorno è maggiore di quelle lunghe 3 giorni del 4,61%. Sia a 3 che a 7 giorni dall’evento, le accuratezze nel caso di sequenze lunghe 3 giorni sono maggiori di quelle ad 1 giorno rispettivamente del 1,92% e del 3,06%.

Globalmente, in Figura 6‑6, Figura 6‑7 e Figura 6‑8, osserviamo che a distanza di 1 giorno dall’evento le sequenze della durata di 1 giorno hanno un’accuratezza migliore rispetto a quelle di una durata di 3 giorni. Ciò non è vero quando ci poniamo ad una distanza di 3 o 7 giorni dall’evento. In questi casi sequenze della durata di 3 giorni possono avere risultati migliori in base alla combinazione di variabili che scegliamo di dare in ingresso alla rete neurale.

### Confronto delle accuratezze in base alla combinazione di grandezze

Andiamo ora a confrontare l’andamento delle accuratezze in base alla combinazione di dati per determinare quale dia risultati migliori. In Figura 6‑9 e in Figura 6‑10 vediamo dei diagrammi a barre. Le barre blu rappresentano le accuratezze ottenute quando in ingresso alla rete diamo la tensione della cella minima e la potenza del pannello. Le barre arancioni corrispondono alle accuratezze considerando la tensione della cella minima, la potenza del pannello, il SOC e l’irraggiamento. Infine, le barre gialle corrispondono alle accuratezze quando in ingresso alla rete diamo la tensione della cella minima, la potenza del pannello, il SOC, l’irraggiamento e il bilancio delle correnti delle batterie. Sulle ascisse troviamo l’anticipo, in giorni, con il quale si vuole predire l’evento.

In Tabella 6‑45 riportiamo le accuratezze al variare dell’intervallo di previsione. Le intestazioni delle colonne sono il colore delle barre in Figura 6‑9:

Tabella 6‑45: Accuratezze al variare dell'intervallo di previsione con varie combinazioni di variabili e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
|  | Blu | Arancione | Giallo |
| Previsione ad 1 giorno | 94.57% ± 2.13% | 96.69% ± 1.26% | 97.79% ± 1.47% |
| Previsione a 3 giorni | 90.47% ±1.27% | 89.84% ±1.87% | 87.58% ± 2.72% |
| Previsione a 7 giorni | 84.11% ±1.26% | 80.58%±1.80% | 81.83% ± 2.48% |

Grafichiamo le accuratezze relative alle sequenze lunghe 1 giorno:

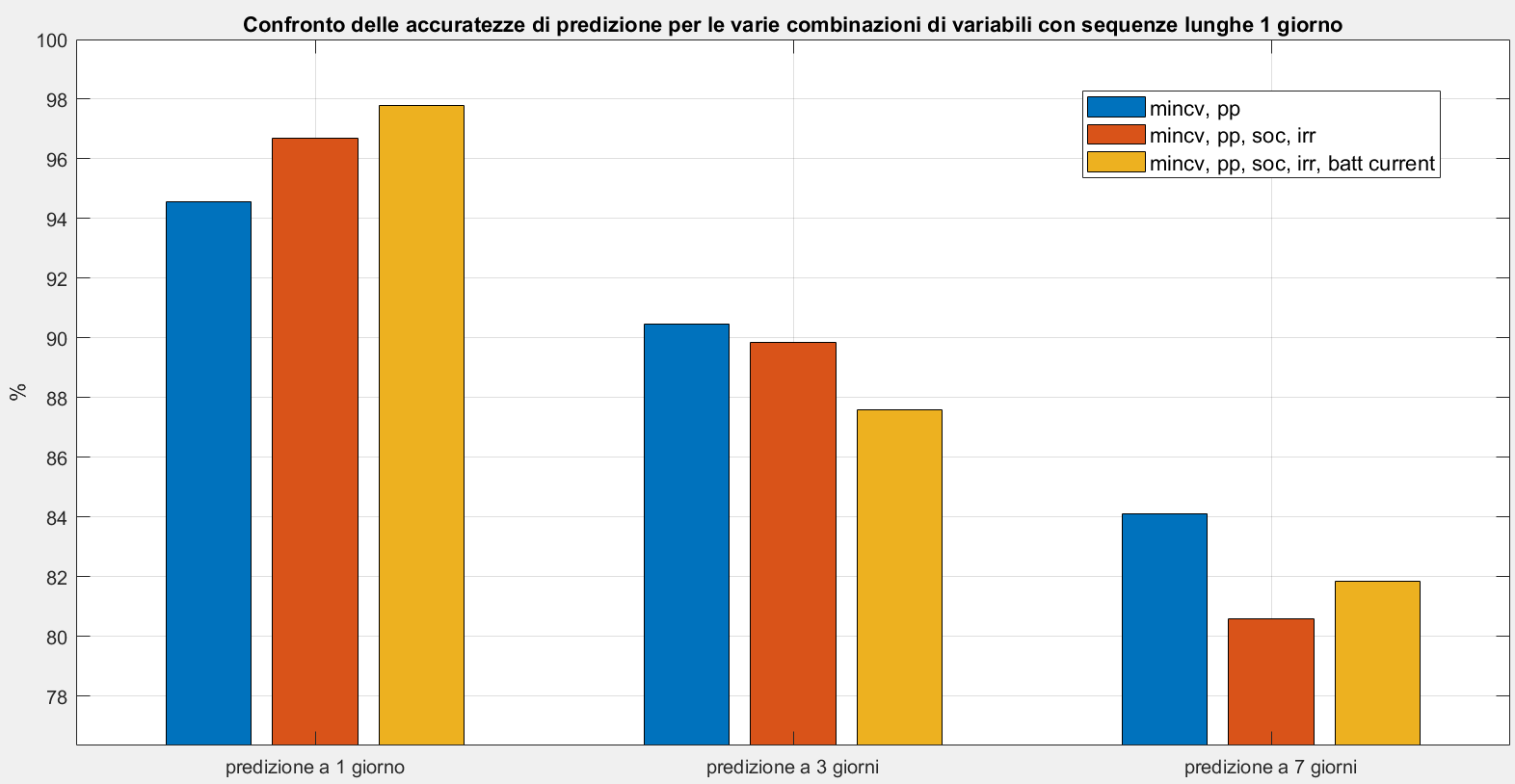


Figura 6‑9: Confronto delle accuratezze di previsione per le varie combinazioni di variabili con sequenze lunghe 1 giorno

Le accuratezze in Figura 6‑9si riferiscono al caso in cui le sequenze abbiano una durata di 24 ore. Constatiamo che, con un intervallo di previsione di 1 giorno i risultati migliori di accuratezza sono rappresentati dalla barra gialla con un’accuratezza del 97,65%. Per un intervallo di previsione d 3 giorni l’accuratezza migliore è rappresentata dalla barra blu ed è del 91,33%. Per predizioni a 7 giorni l’accuratezza migliore è data dalla barra blu e vale 84,11%.

In Tabella 6‑46 riportiamo le accuratezze al variare dell’intervallo di previsione. Le intestazioni delle colonne sono il colore delle barre in Figura 6‑5 :

Tabella 6‑46: Accuratezze al variare dell'intervallo di previsione con varie combinazioni di variabili e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
|  | Blu | Arancione | Giallo |
| Previsione ad 1 giorno | 93.54% ± 0.78% | 94.26% ±2.14% | 93.18% ± 1.41% |
| Previsione a 3 giorni | 90.22% ±1.26% | 90.06% ±1.56% | 89.50% ± 1.45% |
| Previsione a 7 giorni | 81.68% ±1.58% | 81.33% ±1.76% | 84.89% ± 1.90% |

Grafichiamo le accuratezze relative alle sequenze lunghe 1 giorno:

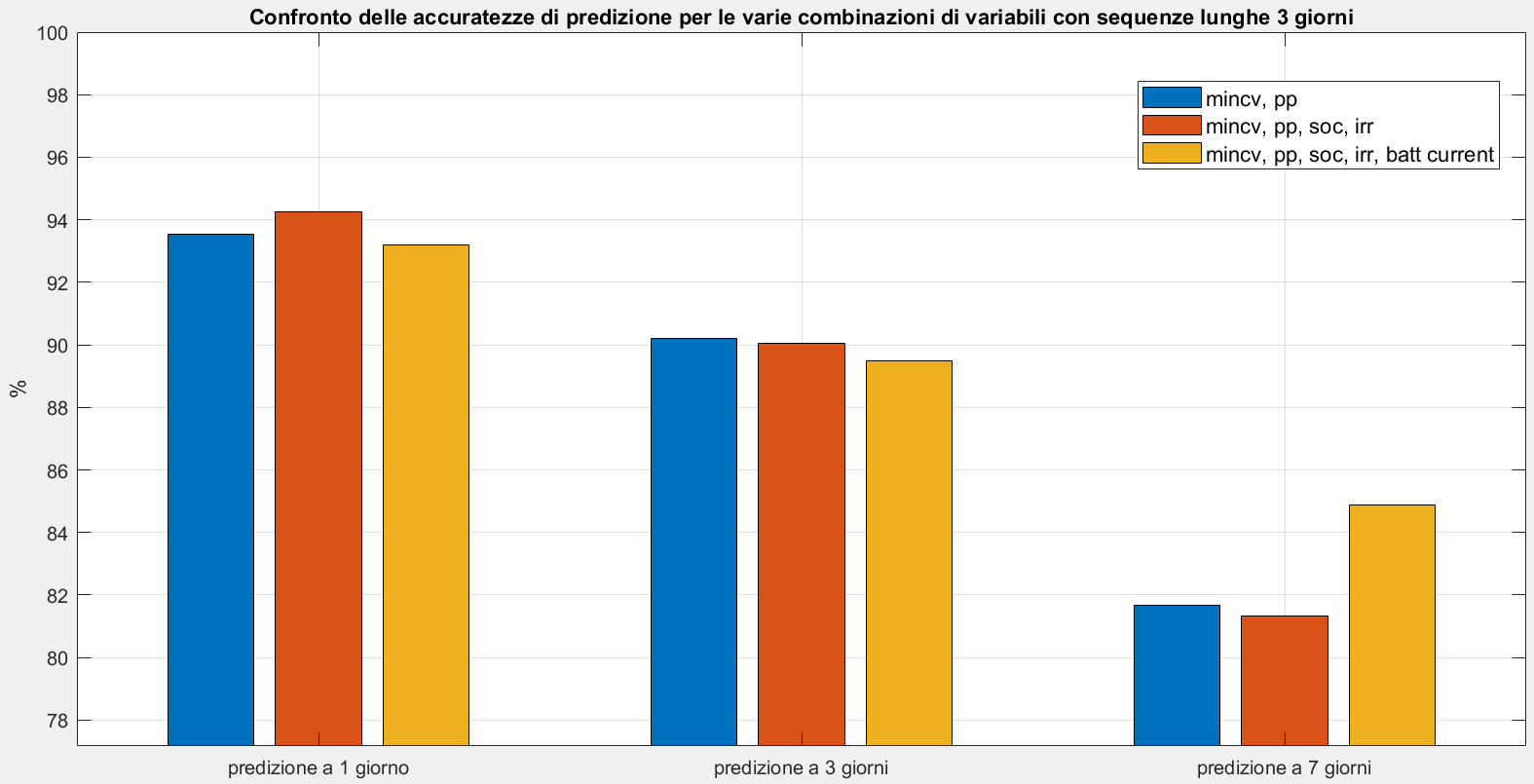


Figura 6‑10: Confronto delle accuratezze di previsione per le varie combinazioni di variabili con sequenze lunghe 3 giorni

Le accuratezze in Figura 6‑10 si riferiscono al caso in cui le sequenze abbiano una durata di 72 ore. Osserviamo che, con un intervallo di previsione di 1 giorno, i risultati migliori di accuratezza sono rappresentati dalla barra arancione con un’accuratezza del 94,26%. Per un intervallo di previsione di 3 giorni, l’accuratezza migliore è rappresentata dalla barra blu ed è del 90,22%. Per predizioni a 7 giorni l’accuratezza migliore è data dalla barra gialla e vale 84,89%. E da notare, che sia nel caso di previsione a 3 giorni che nel caso di previsione a 7 giorni, i risultati forniti dalle combinazioni di variabili rappresentate dalla barra blu e dalla barra arancione sono pressocché uguali.

Facendo riferimento a Figura 6‑9 e a Figura 6‑10 constatiamo che, per un dato intervallo di previsione, non emerge una combinazione di variabili che dia in assoluto i migliori risultati in termini di accuratezza, in quanto i risultati dipendono anche dalla lunghezza in giorni delle sequenze. Globalmente possiamo osservare, per le predizioni ad 1 giorno, delle accuratezze maggiori nel caso di sequenze lunghe 1 giorno rispetto a quelle lunghe 3 giorni. Indipendentemente dalla durata delle sequenze, le accuratezze in questo intervallo di previsione rimangono sempre superiori al 93% fino ad un massimo del 98% approssimativamente. Nell’intervallo di previsione di 3 giorni, le accuratezze vanno da un minimo del 87% fino ad un massimo del 91%. Nell’intervallo di previsione di 7 giorni, le accuratezze vanno da un minimo del 80% fino ad un massimo del 85%.

## Seconda rete alternativa

In questo paragrafo andremmo a vedere i risultati ottenuti con una rete più complessa di quella usata nel paragrafo 6.1. Facendo questo tentativo, vogliamo osservare il comportamento della rete, e valutare se ha maggiore capacità di estrarre correlazioni tra le misurazioni date in ingresso. L’obbiettivo è di avere un modello più robusto e con una maggiore affidabilità. Le prove verranno fatte con le stesse combinazioni di variabili usate nel paragrafo 6.1 e i risultati verranno presentati nello stesso modo. I parametri della rete rimarranno quelli impostati in Tabella 5‑1 e Tabella 5‑2.

Andremmo a commentare i risultati ottenuti con una rete composta da 1 layer BiLSTM seguito da 3 layers fully connected. I primi due layers fully connnected sono dotati di funzioni di attivazione ReLU, mentre il terzo è dotato di una funzione di attivazione softmax. L’utilizzzo delle funzioni di attivazione ReLU per i primi due layers è motivato dal fatto che vogliamo evitare che durante la discesa del gradiente si verifichi il fenomeno della scomparsa del gradiente (17). Essenso la funzione di attivazione ReLU lineare la correzione dei pesi è fatta in modo che essi non vadano a zero esponenzialmente. Il numero dei neuroni di uscita andrà decrescendo dal primo al terzo layer fully connected. Nel nostro caso, abbiamo 19 neuroni in uscita dal primo layer, 10 in uscita dal secondo layer e 2 in uscita dal terzo. L’ultimo layer della rete è un classification layer (vedi appendice in Figura 8‑21 per il codice).

### Tensione della cella minima e potenza del pannello

#### Previsione ad 1 giorno con sequenze lunghe 3 giorni

Poniamoci nel caso in cui le sequenze durino 3 giorni, siano sfalsate di 1 giorno, e ci sia una proporzione di sequenze tale che per una sequenza patologica ce ne siano 3 sane. Vogliamo fare una previsione 1 giorno prima dell’evento. La dimensione dei mini-batch è di 12 elementi con dei fold ognuno da 48 elementi, per un totale di 192 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza del 95% con 0,56% di deviazione standard. L’errore di classificazione è approssimativamente del 5%.

Per quanto riguarda la confusion matrix, le righe rappresentano le classi e sulle colonne troviamo le classi assegnate alle sequenze dal modello predittivo. La prima riga rappresenta la classe patologica e la seconda la classe sana:

Tabella 6‑47: Confusion Matrix per predizioni con anticpo di 1 giorno e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 44.7 ± 1.3 | 3.3 ± 1.3 |
| 1 | 7.3 ± 1.0 | 136.7 ± 1.0 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑47 evidenzia come sulla totalità del dataset mediamente 7,3 elementi vengono classificati erroneamente come patologici e 3,3 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi patologici.

#### Previsione ad 1 giorno con sequenze lunghe 1 giorno

Poniamoci nelle stesse condizioni del caso precedente ma con sequenze di durata 1 giorno. La dimensione dei mini-batch è di 7 elementi con dei fold ognuno da 35 elementi, per un totale di 140 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza del 95% con 3,22% di deviazione standard. L’errore di classificazione è approssimativamente del 5%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑48: Confusion matrix per predizioni con anticipo di 1 giorno e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 33.3 ± 2.9 | 1.7 ± 2.9 |
| 1 | 5.9 ± 3.8 | 99.1 ± 3.8 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑48 evidenzia che sulla totalità del dataset mediamente 5,9 elementi vengono classificati erroneamente come patologici e 1,7 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi sani.

#### Previsione a 3 giorni con sequenze lunghe 3 giorni

In questo caso manteniamo gli stessi parametri ma impostiamo sequenze lunghe 3 giorni e intervallo di previsione uguale a 3 giorni.

La dimensione dei mini-batch è di 23 elementi con dei fold ognuno da 46 elementi, per un totale di 184 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 92% con 1,52% di deviazione standard. L’errore di classificazione è approssimativamente del 8%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑49: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 41.8 ± 2.3 | 4.2 ± 2.3 |
| 1 | 11.4 ± 2.8 | 126.6 ± 2.8 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑49 evidenzia che sulla totalità del dataset mediamente 11,4 elementi vengono classificati erroneamente come patologici e 4,2 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi sani.

#### Previsione a 3 giorni con sequenze lunghe 1 giorno

In questo caso manteniamo gli stessi parametri ma impostiamo sequenze lunghe 1 giorno.

La dimensione dei mini-batch è di 16 elementi con dei fold ognuno da 32 elementi, per un totale di 128 sequenze nel dataset. Da notare che il numero di sequenze è minore rispetto al caso precedente.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 91% con 2,05% di deviazione standard. L’errore di classificazione è approssimativamente del 9%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑50: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 26.1 ± 2.0 | 5.9 ± 2.0 |
| 1 | 6.1 ± 2.6 | 89.9 ± 2.6 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑50 evidenzia che sulla totalità del dataset mediamente 6,1 elementi vengono classificati erroneamente come patologici e 5,9 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi sani.

#### Previsione a 7 giorni con sequenze lunghe 3 giorni

In questo caso manteniamo gli stessi parametri e impostiamo l’intervallo di previsione uguale a 7 giorni,

La dimensione dei mini-batch è di 23 elementi con dei fold ognuno da 46 elementi, per un totale di 184 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 81% con 2,40% di deviazione standard. L’errore di classificazione è approssimativamente del 19%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑51: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 28.2 ± 4.5 | 17.8 ± 4.5 |
| 1 | 17.2 ± 5.1 | 120.8 ± 5.1 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑51 evidenzia che sulla totalità del dataset mediamente 17,2 elementi vengono classificati erroneamente come patologici e 17,8 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi sani.

#### Previsione a 7 giorni con sequenze lunghe 1 giorno

In questo caso manteniamo gli stessi parametri del paragrafo precedente impostando però la lunghezza delle sequenze di 1 giorno.

La dimensione dei mini-batch è di 10 elementi con dei fold ognuno da 31 elementi, per un totale di 124 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 83% con 3,92% di deviazione standard. L’errore di classificazione è approssimativamente del 17%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑52: Confusion matrixper predizioni con anticipo di 7 giorni e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 17.0 ± 5.5 | 14.0 ± 5.5 |
| 1 | 7.1 ± 1.2 | 85.9 ± 1.2 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑52 evidenzia che sulla totalità del dataset mediamente 7,1 elementi vengono classificati erroneamente come patologici e 14,0 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi patologici.

### Tensione della cella minima, potenza del pannello, SOC e irraggiamento

Usiamo la combinazione di tensione della cella minima, potenza del pannello, lo stato di carica della batteria (SOC) e l’irraggiamento del pannello.

#### Previsione ad 1 giorno con sequenze lunghe 3 giorni

Poniamoci nel caso in cui le sequenze durino 3 giorni, siano sfalsate di 1 giorno, e ci sia una proporzione di sequenze tale che per una sequenza patologica ce ne siano 3 sane.

La dimensione dei mini-batch è di 12 elementi con dei fold ognuno da 47 elementi, per un totale di 188 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 94% con 1,18% di deviazione standard. L’errore di classificazione è approssimativamente del 6%.

Per quanto riguarda la confusion matrix, le righe rappresentano le classi e sulle colonne troviamo le classi assegnate alle sequenze dal modello predittivo. La prima riga rappresenta la classe patologica e la seconda la classe sana:

Tabella 6‑53: Confusion matrix per predizioni con anticipo di 1 giorno e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 41.7 ± 3.0 | 5.3 ± 3.0 |
| 1 | 5.7 ± 1.4 | 135.3 ± 1.4 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑53 evidenzia che sulla totalità del dataset mediamente 5,7 elementi vengono classificati erroneamente come patologici e 5,3 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi sani.

#### Previsione ad 1 giorno con sequenze lunghe 1 giorno

Poniamoci nelle stesse condizioni del caso precedente ma con sequenze di durata 1 giorno.

La dimensione dei mini-batch è di 17 elementi con dei fold ognuno da 34 elementi, per un totale di 136 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza del 97% con 1,67% di deviazione standard. L’errore di classificazione è approssimativamente del 3%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑54: Confusion matrix per predizioni con anticpo di 1 giorno e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 33.0 ± 1.6 | 1.0 ± 1.6 |
| 1 | 2.5 ± 2.0 | 99.5 ± 2.0 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑54 evidenzia che sulla totalità del dataset mediamente 2,5 elementi vengono classificati erroneamente come patologici e 1,0 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi sani.

#### Previsione a 3 giorni con sequenze lunghe 3 giorni

In questo caso manteniamo gli stessi parametri ma impostiamo sequenze lunghe 3 giorni e intervallo di previsione uguale a 3 giorni.

La dimensione dei mini-batch è di 15 elementi con dei fold ognuno da 45 elementi, per un totale di 180 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 91% con 1,98% di deviazione standard. L’errore di classificazione è approssimativamente del 9%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑55: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 37.9 ± 3.1 | 7.1 ± 3.1 |
| 1 | 9.9 ± 3.1 | 125.1 ± 3.1 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑55 evidenzia che sulla totalità del dataset mediamente 9,9 elementi vengono classificati erroneamente come patologici e 7,1 elementi vengono classificati erroneamente come sani. Inoltre, le deviazioni standard per la classificazione degli elementi patologici e per la classificazione degli elementi sani sono uguali.

#### Previsione a 3 giorni con sequenze lunghe 1 giorno

In questo caso manteniamo gli stessi parametri ma impostiamo sequenze lunghe 1 giorno.

La dimensione dei mini-batch è di 10 elementi con dei fold ognuno da 31 elementi, per un totale di 124 sequenze nel dataset. Da notare che il numero di sequenze è minore rispetto al caso precedente.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 90% con 2,02% di deviazione standard. L’errore di classificazione è approssimativamente del 10%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑56: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 26.4 ± 1.6 | 4.6 ± 1.6 |
| 1 | 8.0 ± 1.4 | 85.0 ± 1.4 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑56 evidenzia che sulla totalità del dataset mediamente 8,0 elementi vengono classificati erroneamente come patologici e 4,6 elementi vengono classificati erroneamente come sani. Inoltre, le deviazioni standard per la classificazione degli elementi patologici e per la classificazione degli elementi sani sono pressocché uguali.

#### Previsione a 7 giorni con sequenze lunghe 3 giorni

In questo caso manteniamo gli stessi parametri e impostiamo l’intervallo di previsione uguale a 7 giorni.

La dimensione dei mini-batch è di 15 elementi con dei fold ognuno da 46 elementi, per un totale di 184 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 82% con 1,85% di deviazione standard. L’errore di classificazione è approssimativamente del 18%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑57: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 31.0 ± 2.8 | 14.0 ± 2.8 |
| 1 | 19.2 ± 3.5 | 115.8 ± 3.5 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑57 evidenzia che sulla totalità del dataset mediamente 19,2 elementi vengono classificati erroneamente come patologici e 14,0 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi patologici.

#### Previsione a 7 giorni con sequenze lunghe 1 giorno

In questo caso manteniamo gli stessi parametri del paragrafo precedente impostando però la lunghezza delle sequenze di 1 giorno.

La dimensione dei mini-batch è di 10 elementi con dei fold ognuno da 30 elementi, per un totale di 120 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 80% con 2,30% di deviazione standard. L’errore di classificazione è approssimativamente del 20%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑58: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 16.4 ± 4.6 | 13.6 ± 4.6 |
| 1 | 10.8 ± 4.5 | 79.2 ± 4.5 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑58 evidenzia che sulla totalità del dataset mediamente 10,8 elementi vengono classificati erroneamente come patologici e 13,6 elementi vengono classificati erroneamente come sani. Inoltre, le deviazioni standard per la classificazione degli elementi patologici e per la classificazione degli elementi sani sono pressocché uguali.

### Tensione della cella minima, potenza del pannello, SOC, irraggiamento e bilancio delle correnti della batteria

#### Previsione ad 1 giorno con sequenze lunghe 3 giorni

Poniamoci nel caso in cui le sequenze durino 3 giorni, siano sfalsate di 1 giorno, e ci sia una proporzione di sequenze tale che per una sequenza patologica ce ne siano 3 sane.

La dimensione dei mini-batch è di 12 elementi con dei fold ognuno da 47 elementi, per un totale di 188 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 93% con % di deviazione standard. L’errore di classificazione è approssimativamente del 7%.

Per quanto riguarda la matrice di confusione:

Tabella 6‑59: Confusion matrix per predizioni con anticipo di 1 giorno e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 43.1 ± 2.1 | 3.9 ± 2.1 |
| 1 | 10.0 ± 3.6 | 131.0 ± 3.6 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑59 evidenzia che sulla totalità del dataset mediamente 10,0 elementi vengono classificati erroneamente come patologici e 3,9 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi sani.

#### Previsione ad 1 giorno con sequenze lunghe 1 giorno

Poniamoci nelle stesse condizioni del caso precedente ma con sequenze di durata 1 giorno.

La dimensione dei mini-batch è di 17 elementi con dei fold ognuno da 34 elementi, per un totale di 136 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza del 98% con 0,88% di deviazione standard. L’errore di classificazione è approssimativamente del 2%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑60: Confusion matrix per predizioni con anticipo di 1 giorno e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 33.4 ± 1.1 | 0.6 ± 1.1 |
| 1 | 1.5 ± 0.7 | 100.5 ± 0.7 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑60 evidenzia che sulla totalità del dataset mediamente 1,5 elementi vengono classificati erroneamente come patologici e 0,6 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi patologici.

#### Previsione a 3 giorni con sequenze lunghe 3 giorni

In questo caso manteniamo gli stessi parametri ma impostiamo sequenze lunghe 3 giorni e intervallo di previsione uguale a 3 giorni.

La dimensione dei mini-batch è di 15 elementi con dei fold ognuno da 45 elementi, per un totale di 180 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 90% con 1,53% di deviazione standard. L’errore di classificazione è approssimativamente del 10%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑61: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 38.1 ± 2.4 | 6.9 ± 2.4 |
| 1 | 11.4 ± 2.4 | 123.6 ± 2.4 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑61 evidenzia che sulla totalità del dataset mediamente 11,4 elementi vengono classificati erroneamente come patologici e 6,9 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard per la classificazione degli elementi sani e patologici è uguale.

#### Previsione a 3 giorni con sequenze lunghe 1 giorno

In questo caso manteniamo gli stessi parametri ma impostiamo sequenze lunghe 1 giorno.

La dimensione dei mini-batch è di 10 elementi con dei fold ognuno da 31 elementi, per un totale di 124 sequenze nel dataset. Da notare che il numero di sequenze è minore rispetto al caso precedente.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 88% con 2,32% di deviazione standard. L’errore di classificazione è approssimativamente del 12%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑62: Confusion matrix per predizioni con anticpo di 3 giorni e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 29.0 ± 1.6 | 2.0 ± 1.6 |
| 1 | 13.6 ± 2.3 | 79.4 ± 2.3 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑62 evidenzia che sulla totalità del dataset mediamente 13,6 elementi vengono classificati erroneamente come patologici e 2,0 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard per la classificazione degli elementi sani è maggiore.

#### Previsione a 7 giorni con sequenze lunghe 3 giorni

In questo caso impostiamo l’intervallo di previsione uguale a 7 giorni e sequenze lunghe 3 giorni.

La dimensione dei mini-batch è di 15 elementi con dei fold ognuno da 45 elementi, per un totale di 180 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 85% con 1,67% di deviazione standard. L’errore di classificazione è approssimativamente del 15%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑63: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 32.5 ± 3.9 | 12.5 ± 3.9 |
| 1 | 14.3 ± 3.1 | 120.7 ± 3.1 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑63 evidenzia che sulla totalità del dataset mediamente 14,3 elementi vengono classificati erroneamente come patologici e 12,5 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi patologici.

#### Previsione a 7 giorni con sequenze lunghe 1 giorno

In questo caso manteniamo gli stessi parametri del paragrafo precedente impostando però la lunghezza delle sequenze di 1 giorno.

La dimensione dei mini-batch è di 10 elementi con dei fold ognuno da 30 elementi, per un totale di 120 sequenze nel dataset.

L’accuratezza che otteniamo è la seguente:

Otteniamo un’accuratezza approssimativa del 80% con 2,40% di deviazione standard. L’errore di classificazione è approssimativamente del 20%.

La media e la deviazione standard delle confusion matrix sono le seguenti:

Tabella 6‑64: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
| Etichette di riferimento | 0 | 15.6 ± 3.1 | 14.4 ± 3.1 |
| 1 | 9.1 ± 3.5 | 80.9 ± 3.5 |
|  | | 0 | 1 |
| Etichette predette | |

La confusion matrix in Tabella 6‑64 evidenzia che sulla totalità del dataset mediamente 9,1 elementi vengono classificati erroneamente come patologici e 14,4 elementi vengono classificati erroneamente come sani. Inoltre, la deviazione standard è maggiore per la classificazione degli elementi sani.

### Confronto delle accuratezze in base alla durata delle sequenze

A seguire le accuratezze ottenute con la combinazione di tensione della cella minima e potenza del pannello:

Tabella 6‑65: Tabella delle accuratezze ottenute con la tensione della cella minima e potenza del pannello

|  |  |  |
| --- | --- | --- |
|  | Sequenze da 3 giorni | Sequenze da 1 giorno |
| Previsione ad 1 giorno | 94.48% ±0.56% | 94.57% ±3.22% |
| Previsione a 3 giorni | 91.52% ±1.52% | 90.63% ±2.05% |
| Previsione a 7 giorni | 80.98% ±2.40% | 82.98% ±3.92% |

A seguire un diagramma a barre che rappresenta le accuratezze in Tabella 6‑65:

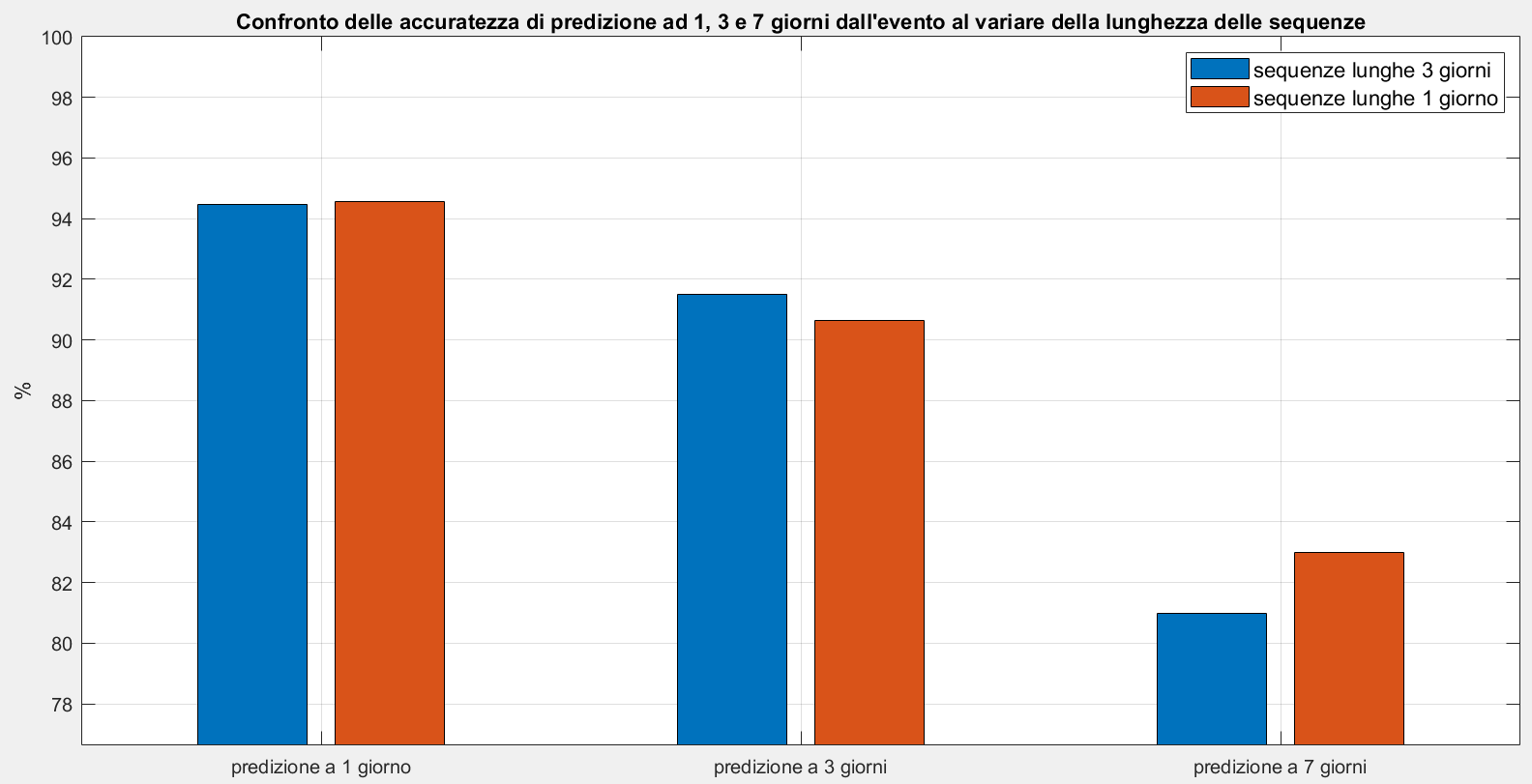


Figura 6‑11: Confronto delle accuratezze ad 1, 3 e 7 giorni dall'evento al variare della lunghezza delle sequenze. Variabili usate: tensione della cella minima e potenza del pannello

In Figura 6‑11, osserviamo come l’accuratezza decresca a mano a mano che il numero di giorni dall’evento aumenta. L’accuratezza migliore in assoluto è uguale a 94,57% e viene ottenuta nel caso di previsione ad 1 giorno con sequenze lunghe 1 giorno. Il caso peggiore invece corrisponde a sequenze lunghe 3 giorni a distanza di 7 giorni dall’evento ed è 80,98%. Nel caso di previsione ad 1 e a 3 giorni dall’evento, l’accuratezza per sequenze lunghe 1 giorno è pressocché identica a quella per sequenze lunghe 3 giorni. A 7 giorni dall’evento l’accuratezza per sequenze lunghe 1 giorno è maggiore di quella per sequenze lunghe 3 giorni del 2%.

A seguire le accuratezze ottenute con la combinazione di tensione della cella minima, potenza del pannello, SOC e irraggiamento:

Tabella 6‑66: Tabella delle accuratezze ottenute con la tensione della cella minima, potenza del pannello, SOC e irraggiamento

|  |  |  |
| --- | --- | --- |
|  | Sequenze da 3 giorni | Sequenze da 1 giorno |
| Previsione ad 1 giorno | 94.15% ±1.18% | 97.43% ±1.67% |
| Previsione a 3 giorni | 90.56% ±1.98% | 89.84% ±2.02% |
| Previsione a 7 giorni | 81.56% ±1.85% | 79.67% ±2.30% |

A seguire un diagramma a barre che rappresenta le accuratezze in Tabella 6‑66:

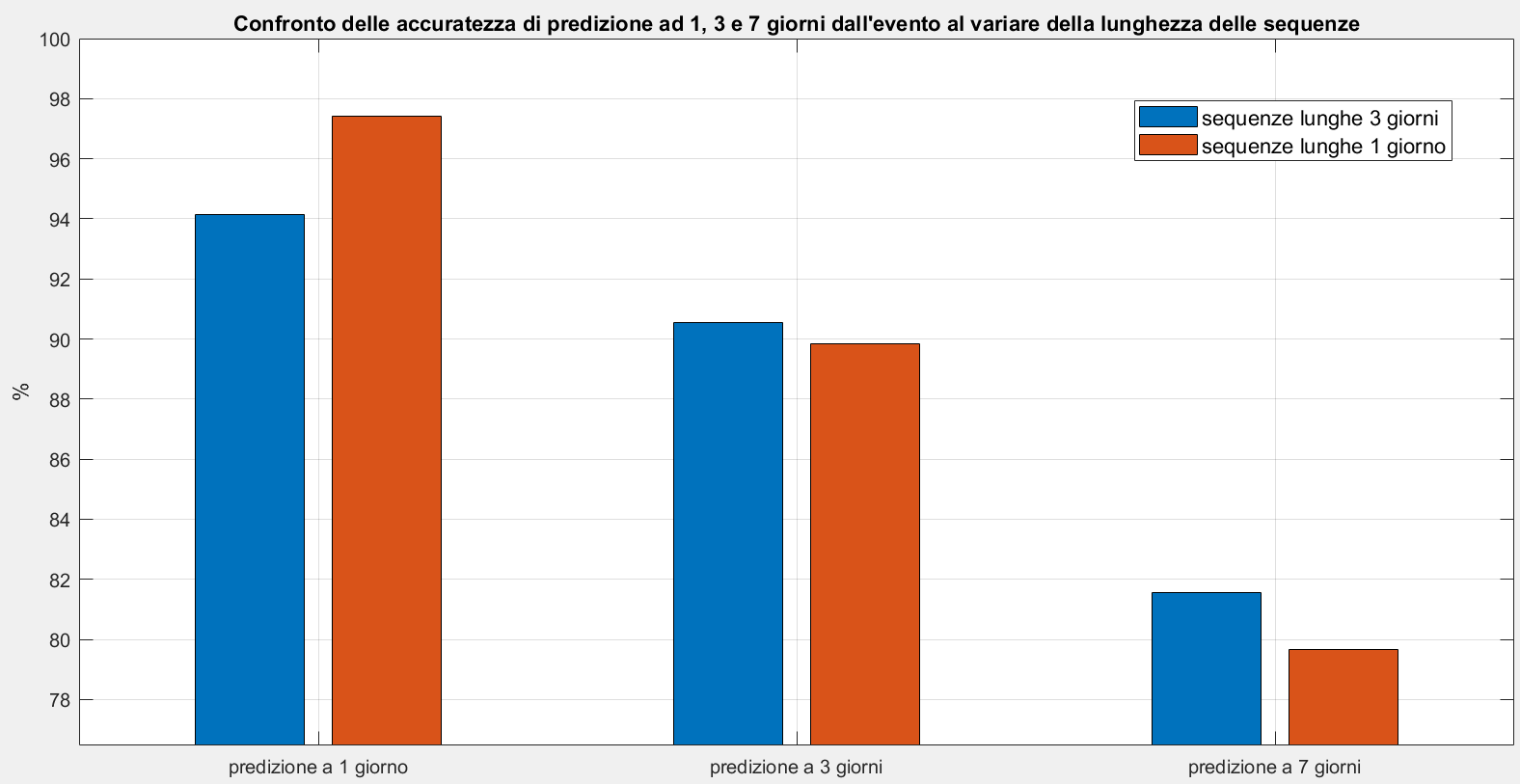


Figura 6‑12: Confronto delle accuratezze ad 1, 3 e 7 giorni dall'evento al variare della lunghezza delle sequenze. Variabili usate: tensione della cella minima, potenza del pannello, SOC e irraggiamento

In Figura 6‑12, osserviamo come l’accuratezza decresca a mano a mano che il numero di giorni dall’evento aumenta. L’accuratezza migliore in assoluto è uguale a 97,43% e viene ottenuta nel caso di previsione ad 1 giorno con sequenze lunghe 1 giorno. Il caso peggiore invece corrisponde a sequenze lunghe 1 giorno a distanza di 7 giorni dall’evento ed è 79,67%. Nel caso di previsione ad 1 giorno dall’evento, l’accuratezza per sequenze lunghe 1 giorno è maggiore di quella per sequenze lunghe 3 giorni del 3,28%. A 3 giorni dall’evento l’accuratezza nel caso di sequenze lunghe 3 giorni è pressocché uguale a quella ad 1 giorno. A 7 giorni dall’evento l’accuratezza per sequenze lunghe 3 giorni è maggiore di quella per sequenze lunghe 1 giorno del 1,89%.

A seguire le accuratezze ottenute con la combinazione di tensione della cella minima, potenza del pannello, SOC, irraggiamento e bilancio delle correnti della batteria:

Tabella 6‑67: Tabella delle accuratezze ottenute con la tensione della cella minima, la potenza del pannello, SOC, irraggiamento e bilancio delle correnti della batteria

|  |  |  |
| --- | --- | --- |
|  | Sequenze da 3 giorni | Sequenze da 1 giorno |
| Previsione ad 1 giorno | 92.61% ±2.00% | 98.46% ±0.88% |
| Previsione a 3 giorni | 89.83% ±1.53% | 87.42% ±2.32% |
| Previsione a 7 giorni | 85.11% ±1.67% | 80.42%±2.40% |

A seguire un diagramma a barre che rappresenta le accuratezze in Tabella 6‑67:

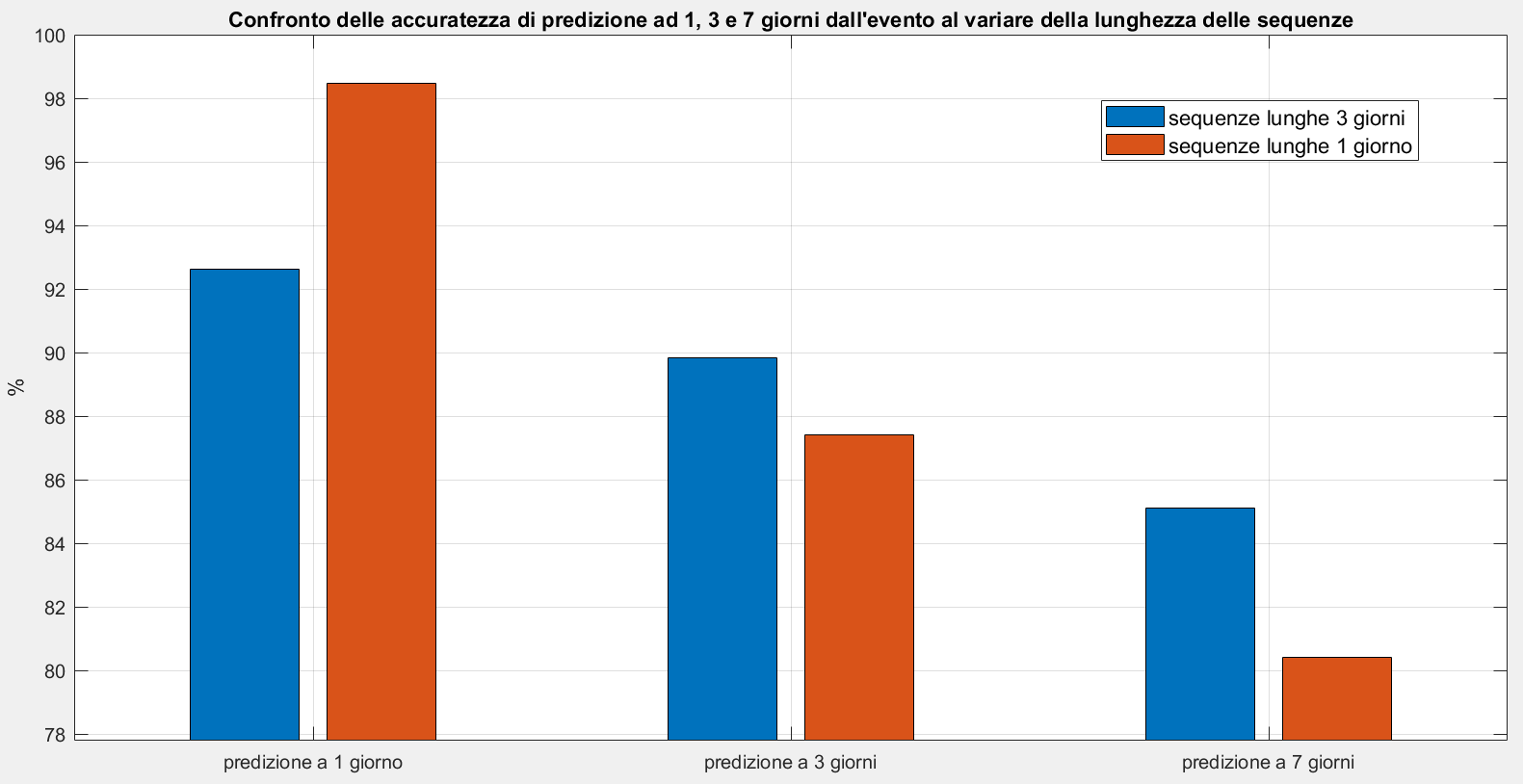


Figura 6‑13: Confronto delle accuratezze ad 1, 3 e 7 giorni dall'evento al variare della lunghezza delle sequenze. Variabili usate: tensione della cella minima, potenza del pannello, SOC, irraggiamento e bilancio delle correnti della batteria

In Figura 6‑13, osserviamo come l’accuratezza decresca a mano a mano che il numero di giorni dall’evento aumenta. L’accuratezza migliore in assoluto è uguale a 98,46% e viene ottenuta nel caso di previsione ad 1 giorno con sequenze lunghe 1 giorno. Il caso peggiore invece corrisponde a sequenze lunghe 1 giorno a distanza di 7 giorni dall’evento ed è 80,42%. Nel caso di previsione ad 1 giorno dall’evento, l’accuratezza per sequenze lunghe 1 giorno è maggiore di quella per sequenze lunghe 3 giorni del 5,85%. Sia a 3 che a 7 giorni dall’evento le accuratezze nel caso di sequenze lunghe 3 giorni sono maggiori di quelle ad 1 giorno rispettivamente del 2,41% e del 4,69%.

### Confronto delle accuratezze in base alla combinazione di grandezze

Andiamo ora a confrontare l’andamento delle accuratezze in base alla combinazione di dati per determinare quale dia risultati migliori. In Figura 6‑14 e in Figura 6‑15 vediamo dei diagrammi a barre. Le barre blu rappresentano le accuratezze ottenute quando in ingresso alla rete diamo la tensione della cella minima e la potenza del pannello. Le barre arancioni corrispondono alle accuratezze considerando la tensione della cella minima, la potenza del pannello, il SOC e l’irraggiamento. Infine, le barre gialle corrispondono alle accuratezze quando in ingresso alla rete diamo la tensione della cella minima, la potenza del pannello, il SOC, l’irraggiamento e il bilancio delle correnti delle batterie. Sulle ascisse troviamo l’anticipo, in giorni, con il quale si vuole predire l’evento.

In Tabella 6‑68 riportiamo le accuratezze al variare dell’intervallo di previsione. Le intestazioni delle colonne sono il colore delle barre in Figura 6‑14:

Tabella 6‑68: Accuratezze al variare dell'intervallo di previsione con varie combinazioni di variabili e sequenze lunghe 1 giorno

|  |  |  |  |
| --- | --- | --- | --- |
|  | Blu | Arancione | Giallo |
| Previsione ad 1 giorno | 94.57% ±3.22% | 97.43% ±1.67% | 98.46% ±0.88% |
| Previsione a 3 giorni | 90.63% ±2.05% | 89.84% ±2.02% | 87.42% ±2.32% |
| Previsione a 7 giorni | 82.98% ±3.92% | 79.67% ±2.30% | 80.42%±2.40% |

Grafichiamo le accuratezze relative alle sequenze lunghe 1 giorno:

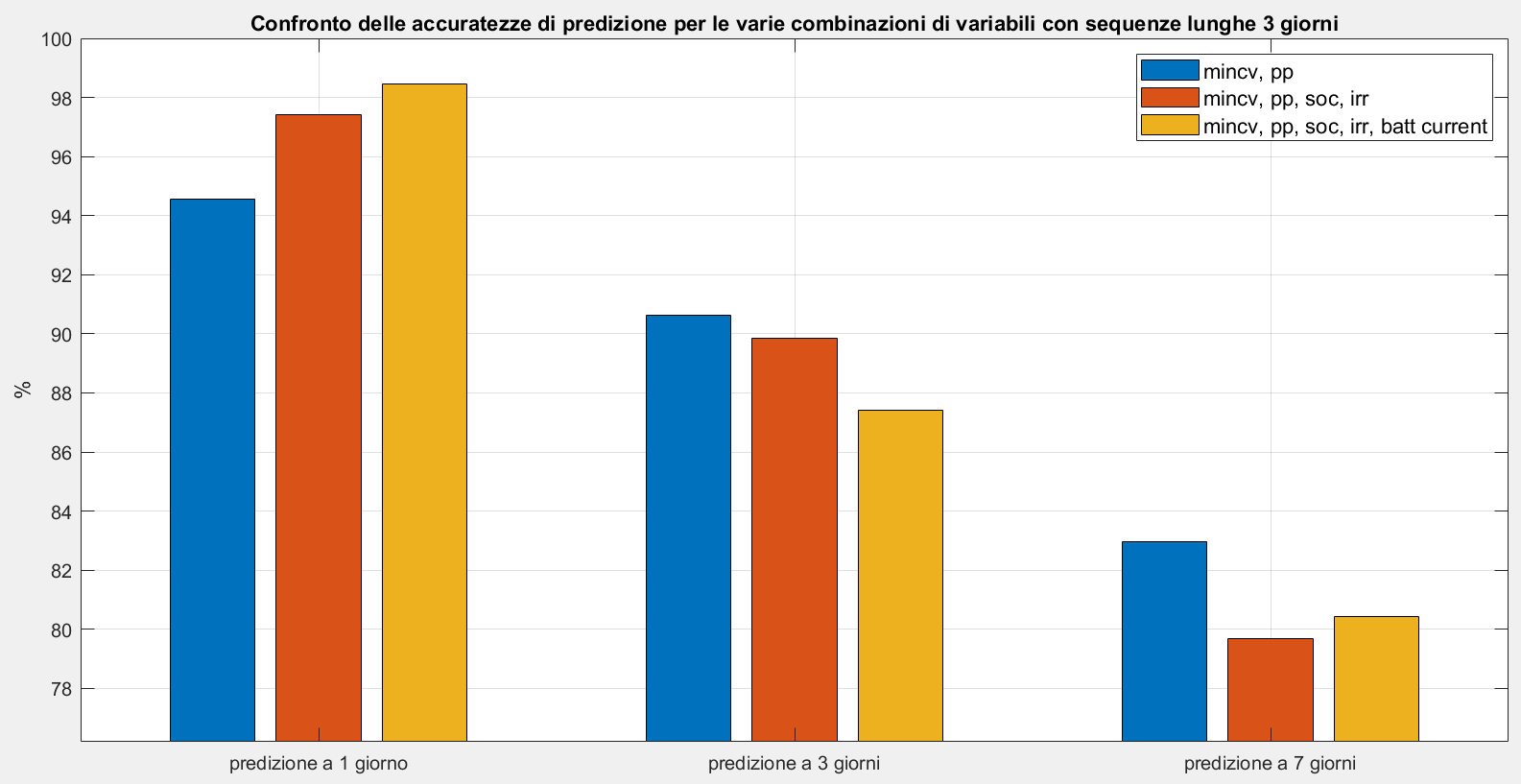


Figura 6‑14: Confronto delle accuratezze di previsione per le varie combinazioni di variabili con sequenze lunghe 1 giorno

Le accuratezze in Figura 6‑14 si riferiscono al caso in cui le sequenze abbiano una durata di 24 ore. Constatiamo che, con un intervallo di previsione di 1 giorno i risultati migliori di accuratezza sono rappresentati dalla barra gialla con un’accuratezza del 98,46%. Per un intervallo di previsione d 3 giorni l’accuratezza migliore è rappresentata dalla barra blu ed è del 90,63%. Per predizioni a 7 giorni l’accuratezza migliore è data dalla barra blu e vale 82,98%.

In Tabella 6‑69 riportiamo le accuratezze al variare dell’intervallo di previsione. Le intestazioni delle colonne sono il colore delle barre in Figura 6‑15:

Tabella 6‑69: Accuratezze al variare dell'intervallo di previsione con varie combinazioni di variabili e sequenze lunghe 3 giorni

|  |  |  |  |
| --- | --- | --- | --- |
|  | Blu | Arancione | Giallo |
| Previsione ad 1 giorno | 94.48% ±0.56% | 94.15% ±1.18% | 92.61% ±2.00% |
| Previsione a 3 giorni | 91.52% ±1.52% | 90.56% ±1.98% | 89.83% ±1.53% |
| Previsione a 7 giorni | 80.98% ±2.40% | 81.56% ±1.85% | 85.11% ±1.67% |

Grafichiamo le accuratezze relative alle sequenze lunghe 3 giorni:

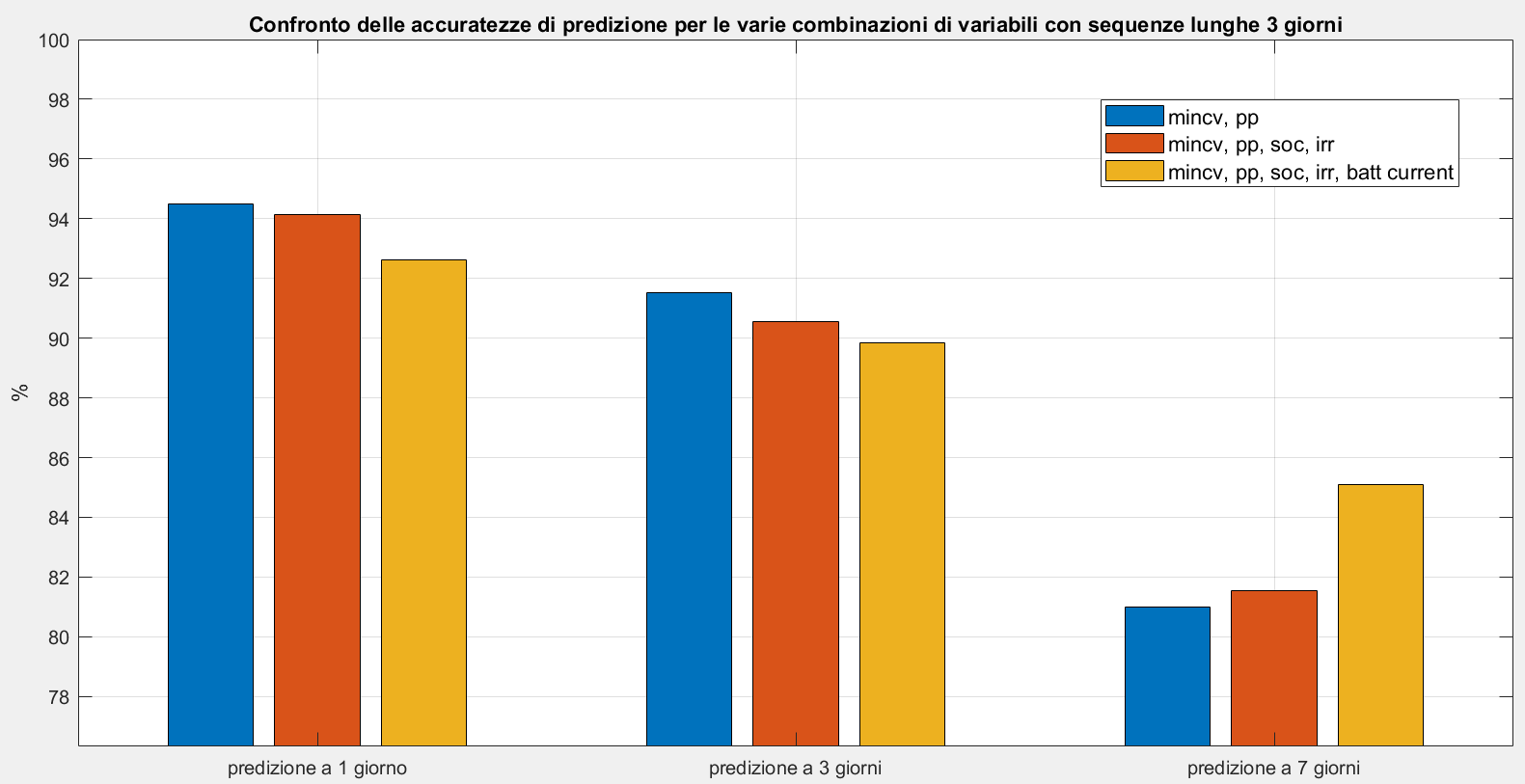


Figura 6‑15: Confronto delle accuratezze di previsione per le varie combinazioni di variabili con sequenze lunghe 3 giorni

Le accuratezze in Figura 6‑15 si riferiscono al caso in cui le sequenze abbiano una durata di 72 ore. Osserviamo che, con un intervallo di previsione di 1 giorno, i risultati migliori di accuratezza sono rappresentati dalla barra blu con un’accuratezza del 94,48%. Per un intervallo di previsione di 3 giorni, l’accuratezza migliore è rappresentata dalla barra blu ed è del 91,52%. Per predizioni a 7 giorni, l’accuratezza migliore è data dalla barra gialla e vale 85,11%.

Facendo riferimento a Figura 6‑14 e a Figura 6‑15 constatiamo che, per un dato intervallo di previsione, non emerge una combinazione di variabili che dia in assoluto i migliori risultati in termini di accuratezza, in quanto i risultati dipendono anche dalla lunghezza in giorni delle sequenze. Globalmente possiamo osservare, per le predizioni ad 1 giorno, delle accuratezze maggiori nel caso di sequenze lunghe 1 giorno rispetto a quelle lunghe 3 giorni. Indipendentemente dalla durata delle sequenze, le accuratezze in questo intervallo di previsione rimangono sempre superiori al 92% fino ad un massimo del 98,5% approssimativamente. Nell’intervallo di previsione di 3 giorni, le accuratezze vanno da un minimo del 87% fino ad un massimo del 92%. Nell’intervallo di previsione di 7 giorni, le accuratezze vanno da un minimo del 80% fino ad un massimo del 85%.

## Confronto delle prestazioni delle reti neurali

A seguire, andiamo a evidenziare quali scelte portino alle accuratezze migliori, per la rete neurale usata in **Errore. L'origine riferimento non è stata trovata.**, in base all’intervallo di previsione:

Tabella 6‑70: Presentazione delle migliori accuratezze per ogni intervallo di previsione in base alla lunghezza delle sequenze e alla combinazione di variabili

|  |  |  |  |
| --- | --- | --- | --- |
|  | Combinazione di variabili | Lunghezza delle sequenze in giorni | Accuratezza |
| Previsione ad 1 giorno | tensione della cella minima, potenza del pannello, SOC, irraggiamento, bilancio delle correnti della batteria | 1 | 97.65% ± 0.83% |
| Previsione a 3 giorni | tensione della cella minima, potenza del pannello | 1 | 91.33% ± 1.44% |
| Previsione a 7 giorni | tensione della cella minima, potenza del pannello, SOC, irraggiamento, bilancio delle correnti della batteria | 3 | 85.33% ± 1.02% |

In Tabella 6‑70, vediamo che a distanza di 1 giorno dall’evento l’accuratezza massima è approssimativamente del 98% e viene ottenuta combinando tutte le variabili e prendendo sequenze della durata di 1 giorno. A distanza di 3 giorni dall’evento, l’accuratezza massima è approssimativamente del 90,5% e viene ottenuta con sequenze della durata di 1 giorno, e dando in ingresso la tensione della cella minima e la potenza del pannello solare. A distanza di 7 giorni dall’evento, l’accuratezza massima è approssimativamente del 85% e viene ottenuta combinando tutte le variabili e prendendo sequenze della durata di 3 giorni.

A seguire, andiamo a evidenziare quali scelte portino alle accuratezze migliori, per la rete neurale usata in **Errore. L'origine riferimento non è stata trovata.**, in base all’intervallo di previsione:

Tabella 6‑71: Presentazione delle migliori accuratezze per ogni intervallo di previsione in base alla lunghezza delle sequenze e alla combinazione di variabili

|  |  |  |  |
| --- | --- | --- | --- |
|  | Combinazione di variabili | Lunghezza delle sequenze in giorni | Accuratezza |
| Previsione ad 1 giorno | tensione della cella minima, potenza del pannello, SOC, irraggiamento, bilancio delle correnti della batteria | 1 | 97.79% ± 1.47% |
| Previsione a 3 giorni | tensione della cella minima, potenza del pannello | 1 | 90.47% ±1.27% |
| Previsione a 7 giorni | tensione della cella minima, potenza del pannello, SOC, irraggiamento, bilancio delle correnti della batteria | 3 | 84.89% ± 1.90% |

In Tabella 6‑71, vediamo che a distanza di 1 giorno dall’evento l’accuratezza massima è approssimativamente del 98% e viene ottenuta combinando tutte le variabili e prendendo sequenze della durata di 1 giorno. A distanza di 3 giorni dall’evento, l’accuratezza massima è approssimativamente del 90,5% e viene ottenuta dando in ingresso la tensione della cella minima e la potenza del pannello solare, e prendendo sequenze della durata di 1 giorno. A distanza di 7 giorni dall’evento, l’accuratezza massima è approssimativamente del 85% e viene ottenuta combinando tutte le variabili e prendendo sequenze della durata di 3 giorni.

A seguire, andiamo a evidenziare quali scelte portino alle accuratezze migliori, per la rete neurale usata in **Errore. L'origine riferimento non è stata trovata.**, in base all’intervallo di previsione:

Tabella 6‑72: Presentazione delle migliori accuratezze per ogni intervallo di previsione in base alla lunghezza delle sequenze e alla combinazione di variabili

|  |  |  |  |
| --- | --- | --- | --- |
|  | Combinazione di variabili | Lunghezza delle sequenze in giorni | Accuratezza |
| Previsione ad 1 giorno | tensione della cella minima, potenza del pannello, SOC, irraggiamento, bilancio delle correnti della batteria | 1 | 98.46% ±0.88% |
| Previsione a 3 giorni | tensione della cella minima, potenza del pannello | 3 | 91.52% ±1.52% |
| Previsione a 7 giorni | tensione della cella minima, potenza del pannello, SOC, irraggiamento, bilancio delle correnti della batteria | 3 | 85.11% ±1.67% |

In Tabella 6‑72, vediamo che a distanza di 1 giorno dall’evento l’accuratezza massima è approssimativamente del 98,5% e viene ottenuta combinando tutte le variabili e prendendo sequenze della durata di 1 giorno. A distanza di 3 giorni dall’evento, l’accuratezza massima è approssimativamente del 91,5% e viene ottenuta dando in ingresso la tensione della cella minima e la potenza del pannello solare, e prendendo sequenze della durata di 3 giorni. A distanza di 7 giorni dall’evento, l’accuratezza massima è approssimativamente del 85% e viene ottenuta combinando tutte le variabili e prendendo sequenze della durata di 3 giorni.

A seguire vediamo la rappresentazione dei dati in Tabella 6‑70, Tabella 6‑71 e Tabella 6‑72:

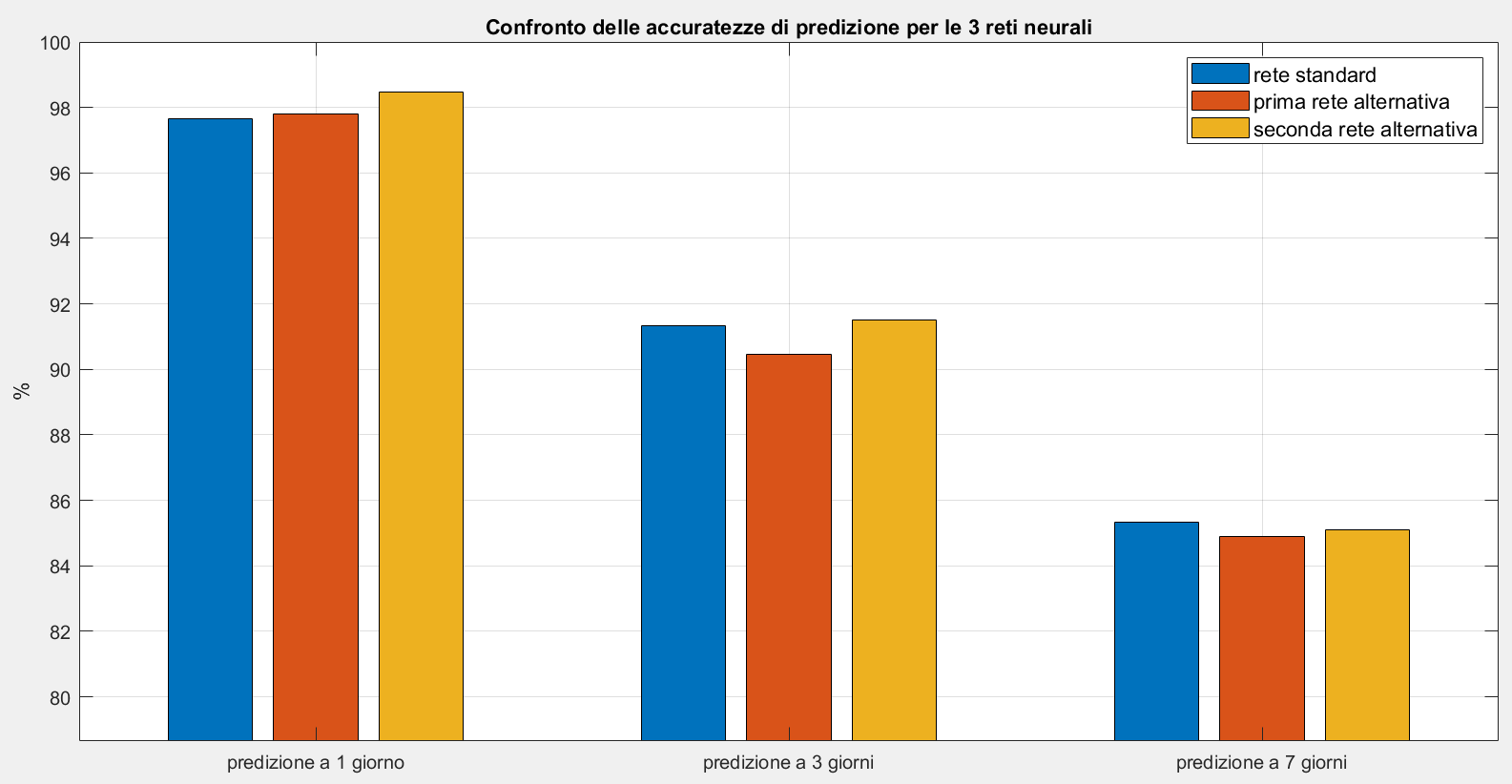


Figura 6‑16: Confronto delle accuratezze per le 3 reti neurali usate al variare dell'intervallo di previsione

In Figura 6‑16, le barre blu rappresentano la rete neurale, presentata in 6, e a cui ci riferiamo come rete standard. Essa è composta da 1 layer BiLSTM seguito da 1 layer fully connected. Le barre arancioni fanno riferimento alla rete neurale, usata in 6.2, e a cui ci riferiamo come prima rete alternativa. Essa è composta da 2 layers BiLSTM seguiti da un layer fully connected. Le barre gialle rappresentano la seconda rete alternativa, presentata in 6.3, e a cui ci riferiamo come seconda rete alternativa. Essa è composta da 1 layer BiLSTM e 3 layers fully connected in cascata.

Nel caso di previsione con 1 giorno d’anticipo rispetto all’evento, la prima rete alternativa ottiene l’accuratezza maggiore e vale 98,5%. Nel caso di previsione con 3 giorni d’anticipo rispetto all’evento, la prima rete alternativa ottiene l’accuratezza maggiore e vale 91,5%. Nel caso di previsione con 7 giorni d’anticipo rispetto all’evento, la rete standard ottiene l’accuratezza maggiore e vale 85,3%. In questo stesso caso la seconda rete alternativa ottiene un’accuratezza del 85,1%, un valore pressocché uguale a quello della rete standard. Ne deduciamo dunque che globalmente la seconda rete alternativa ha risultati di accuratezza maggiori ed è dunque la più affidabile.

# Conclusioni

Negli ultimi anni, l'interesse per la stima delle condizioni delle apparecchiature in ambito industriale è fortemente aumentato. Grazie all’avvento dell’internet of things, è cresciuta esponenzialmente la disponibilità di misurazioni, in grado di descrivere accuratamente lo stato di salute dei sistemi. Questa mole di dati, ha reso possibile fare manutenzione predittiva sui componenti dei dispositivi, per abbattere i costi relativi ai guasti, e azzerarne/ridurne le conseguenze. In questa tesi, abbiamo creato un sistema di manutenzione predittiva, per dei sistemi di monitoraggio alimentati da pannelli solari, la cui criticità risiede nell’ubicazione, che li rende difficilmente accessibili, nel caso sia necessaria una manutenzione da parte di un operatore. A tal fine, abbiamo sfruttato misurazioni di campo prelevate nel corso di un anno, riguardanti il pacco batterie, il pannello solare e la centralina meteo, per fare manutenzione predittiva. La rete neurale implementata, è in grado di analizzare questi dati, e di riconoscerne andamenti anomali, che possano portare ad uno spegnimento del dispositivo. Nello specifico, abbiamo addestrato dei modelli, per predire un guasto 1, 3 e 7 giorni prima dell’evento di guasto. Chiaramente, all’aumentare della distanza dall’evento di guasto, diminuisce l’accuratezza del modello. Abbiamo distinto, i casi in cui le sequenze siano della durata di 1 giorno, da quelli in cui le sequenze siano della durata di 3 giorni. Inoltre, abbiamo variato anche le combinazioni di variabili, che compongono le sequenze. In particolare, siamo partiti dando in ingresso, solamente la tensione della cella minima e la potenza del pannello, per poi aggiungere l’irraggiamento, lo stato di carica (SOC), ed il bilancio delle correnti nella batteria. Per quest’applicazione, le reti neurali implementate, fanno tutte uso di layers BiLSTM, di modo da evitare, il problema della scomparsa del gradiente, e quindi sfruttare le correlazioni tra campioni distanti fra loro, visto che usiamo sequenze con più di 1000 campioni. Nello specifico, siamo partiti da una rete costituita un layer BiLSTM e un layer fully connected, per poi provarne una con 2 layers BILSTM e 1 fully connected, ed infine, provarne una con 1 layer BiLSTM e 3 fully connected. Tutte queste variazioni, sono state fatte per capire, quale di queste prove ci dia l’affidabilità maggiore.

In Tabella 7‑1 ricapitoliamo mostra le accuratezze migliori ottenute:

Tabella 7‑1: Ricapitolativo delle migliori accuratezze ottenute al variare dell'intervallo di previsione in base alla composizione della rete neurale, la combinazione di variabili e la lunghezza delle sequenze

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Composizione della rete neurale | Combinazione di variabili | Lunghezza delle sequenze in giorni | Accuratezza |
| Previsione ad 1 giorno dall’evento | 1 BiLSTM + 3 fully connected | tensione della cella minima, potenza del pannello, SOC, irraggiamento, bilancio delle correnti della batteria | 1 | 98.46% ±0.88% |
| Previsione a 3 giorni dall’evento | 1 BiLSTM + 3 fully connected | tensione della cella minima, potenza del pannello | 3 | 91.52% ±1.52% |
| Previsione a 7 giorni dall’evento | 1 BiLSTM + 1 fully connected | tensione della cella minima, potenza del pannello, SOC, irraggiamento, bilancio delle correnti della batteria | 3 | 85.33% ± 1.02% |

Come vediamo in Tabella 7‑1, l’accuratezza migliore è del 98,5%, nel caso di previsione 1 giorno prima dell’evento. Nel caso di previsione a 3 giorni, l’accuratezza è del 91,5%, mentre nel caso di previsione a 7 giorni, l’accuratezza è del 85%.

In base ai risultati ottenuti, concludiamo che sia possibile, fare un‘analisi predittiva dei livelli di carica della batteria, per permettere un corretto funzionamento degli apparati, e minimizzare gli spegnimenti. Potrebbe essere una prospettiva concreta, quella di integrare questo strumento di analisi su uno SCADA, che generi dei messaggi di avvertimento per l’utente, di modo da poter programmare degli interventi di manutenzione, oppure effettuare delle azioni da remoto.

In futuro, potrebbe essere interessante raccogliere ulteriori dati dai sensori, di modo da poter allenare dei modelli, non solo in grado di prevedere uno spegnimento, ma anche di risalire alla causa dello scarso irraggiamento. In particolare, in zone maggiormente inquinate, si può verificare la riduzione di efficienza dei pannelli solari, causata dalla sedimentazione di polveri sottili sugli stessi. Sarebbe interessante, confrontare le ore di irraggiamento del pannello, con quelle previste da un modello del profilo dell’orizzonte, al fine di rilevare eventuali differenze rispetto al modello, riconducibili a degli oggetti che intralcino la corretta irraggiamento del pannello (vegetazione, neve o altro). Inoltre, confrontando la potenza effettivamente prodotta con l’irraggiamento rilevato dalla stazione meteo sarebbe possibile rilevare anomalie di installazione quale l’inclinazione, l’angolo di azimut, o difetti hardware nei componenti. Futuri approfondimenti, volti ad aumentare l’affidabilità dei modelli, potrebbero riguardare l’utilizzo di dataset nativamente sincroni grazie ai quali, sarebbe possibile ridurre l’incertezza introdotta dall’interpolazione dei dati.

# Appendice

Immagine che contiene testo

Descrizione generata automaticamente

Figura 8‑1: Creazione di una struttura con celle contenenti ognuna i dati relativi ad una sola variabile

Immagine che contiene testo

Descrizione generata automaticamente

Figura 8‑2: Interpolazione dei campioni in diagnostica

Immagine che contiene testo

Descrizione generata automaticamente

Figura 8‑3: Sovrascrittura dei valori invariati

Immagine che contiene testo

Descrizione generata automaticamente

Figura 8‑4: Funzione di sovra campionamento

Immagine che contiene testo, persona, screenshot

Descrizione generata automaticamente

Figura 8‑5: allineamento delle sequenze temporali

Immagine che contiene testo

Descrizione generata automaticamente

Figura 8‑6: Struttura dati finale

Immagine che contiene testo

Descrizione generata automaticamente

Figura 8‑7: Definizione dei parametri come variabili globali

Immagine che contiene testo

Descrizione generata automaticamente

Figura 8‑8: Main

Immagine che contiene testo

Descrizione generata automaticamente

Figura 8‑9: funzione per scandire i giorni

Immagine che contiene testo

Descrizione generata automaticamente

Figura 8‑10: Creazione delle sequenze da 3 giorni

Immagine che contiene testo

Descrizione generata automaticamente

Figura 8‑11: Suddivisione delle sequenze tra sane e patologiche

Immagine che contiene testo

Descrizione generata automaticamente

Figura 8‑12: Estrazione delle sequenze predittive

Immagine che contiene testo

Descrizione generata automaticamente

Figura 8‑13: Estrazione casuale delle sequenze sane

Immagine che contiene testo, persona, screenshot

Descrizione generata automaticamente

Figura 8‑14: Funzione di normalizzazione

Immagine che contiene testo

Descrizione generata automaticamente

Figura 8‑15: Etichettatura del dataset

Immagine che contiene testo

Descrizione generata automaticamente

Figura 8‑16: Creazione del dataset

Immagine che contiene testo

Descrizione generata automaticamente

Figura 8‑17: Partizione statica

Immagine che contiene testo

Descrizione generata automaticamente

Figura 8‑18: K-Folding

Immagine che contiene testo

Descrizione generata automaticamente

Figura 8‑19: Layers della rete neurale

Immagine che contiene testo

Descrizione generata automaticamente

Figura 8‑20: Layers della prima rete neurale alternativa

Immagine che contiene testo

Descrizione generata automaticamente

Figura 8‑21: Layers della seconda rete neurale alternativa

# Indice delle figure

[Figura 2‑1: Neurone artificiale 10](#_Toc104296839)

[Figura 2‑2: Rete feedforward 11](#_Toc104296840)

[Figura 2‑3: Recurrent Neural Network 12](#_Toc104296841)

[Figura 2‑4: Schema RNN srotolato 13](#_Toc104296842)

[Figura 2‑5: Rete RNN bidirezionale 15](#_Toc104296843)

[Figura 2‑6: Previsione delle parole in una frase in base al resto della frase 15](#_Toc104296844)

[Figura 2‑7: Cella LSTM 16](#_Toc104296845)

[Figura 3‑1: Diversi tipi di manutenzione 19](#_Toc104296846)

[Figura 3‑2: Schema di manutenzione predittiva 20](#_Toc104296847)

[Figura 3‑3: Costi dell'implementazione di un sistema di manutenzione predittiva 22](#_Toc104296848)

[Figura 3‑4: Schema a blocchi di un dispositivo di tipo A 26](#_Toc104296849)

[Figura 3‑5: Implementazione hardware di un dispositivo di tipo B 28](#_Toc104296850)

[Figura 3‑6: Connessione bidirezionale tra i dispositivi in campo e il databse 31](#_Toc104296851)

[Figura 4‑1: Interpolazione 36](#_Toc104296852)

[Figura 4‑2: Correzione del valore nullo in diagnostica 37](#_Toc104296853)

[Figura 4‑3: Traslazione del dato 38](#_Toc104296854)

[Figura 4‑4: Problema della non trascrizione del dato di notte per la corretta rappresentazione della potenza del pannello 39](#_Toc104296855)

[Figura 4‑5: Sovra campionamento al minuto 40](#_Toc104296856)

[Figura 4‑6: Differenza di andamenti tra dato sovra campionato e non, là dove si verifica una mancata ricezione del dato 43](#_Toc104296857)

[Figura 4‑7: Paragone tra le sequenze estratte e la totalità del dato sincronizzato 44](#_Toc104296858)

[Figura 4‑8: Lasso temporale non valido ai fini del dataset 45](#_Toc104296859)

[Figura 4‑9: Sequenza patologicac di riferimento 47](#_Toc104296860)

[Figura 4‑10: Sequenza predittiva 47](#_Toc104296861)

[Figura 4‑11: Selezione della sequenza predittiva con predzione a 7giorni e sequenza di riferimento sana 48](#_Toc104296862)

[Figura 4‑12:Selezione della sequenza predittiva con predzione a 7giorni e sequenza di riferimento patologica 48](#_Toc104296863)

[Figura 4‑13: Disposizione tra sequenze di riferimento e predittive a 7 giorni per ogni dispositivo 49](#_Toc104296864)

[Figura 4‑14: Dispostivo 13008 50](#_Toc104296865)

[Figura 4‑15: Dispositivo 1021 50](#_Toc104296866)

[Figura 4‑16: Dispositivo 1025 51](#_Toc104296867)

[Figura 4‑17: Dispositivo 1059 51](#_Toc104296868)

[Figura 4‑18: Dispositivo 16399 52](#_Toc104296869)

[Figura 4‑19: Periodo in cui l’andamento della tensione della cella minima diventa patologico 53](#_Toc104296870)

[Figura 4‑20: Prima sequenza 54](#_Toc104296871)

[Figura 4‑21: Seconda sequenza 54](#_Toc104296872)

[Figura 4‑22: Terza sequenza 54](#_Toc104296873)

[Figura 4‑23: Sfalsamento delle sequenze 55](#_Toc104296874)

[Figura 4‑24: Andamento della tensione della cella minima normalizzata per il dispositvo 13008 56](#_Toc104296875)

[Figura 4‑25: Sequenze patologiche della tensione della cella minima 58](#_Toc104296876)

[Figura 4‑26: Sequenze sane della tensione della cella minima 59](#_Toc104296877)

[Figura 4‑27: Sequenze patologiche della potenza del pannello 60](#_Toc104296878)

[Figura 4‑28: Sequenze sane della potenza del pannello 61](#_Toc104296879)

[Figura 4‑29: Sequenze patologiche del SOC 62](#_Toc104296880)

[Figura 4‑30: Sequenze sane del SOC 62](#_Toc104296881)

[Figura 4‑31: Rappresentazione delle sequenze consecutive lunghe 1 giorno 65](#_Toc104296882)

[Figura 5‑1: Lunghezza delle sequenze uniforme 69](#_Toc104296883)

[Figura 5‑2: Andamento tipico dell'accuratezza e della funzione di perdita con le suddette opzioni della rete 70](#_Toc104296884)

[Figura 5‑3: Confusion chart 70](#_Toc104296885)

[Figura 6‑1: Confronto delle accuratezze ad 1, 3 e 7 giorni dall'evento al variare della lunghezza delle sequenze. Variabili usate: tensione della cella minima e potenza del pannello 92](#_Toc104296886)

[Figura 6‑2: Confronto delle accuratezze ad 1, 3 e 7 giorni dall'evento al variare della lunghezza delle sequenze. Variabili usate: tensione della cella minima, potenza del pannello, SOC e irraggiamento 94](#_Toc104296887)

[Figura 6‑3: Confronto delle accuratezze ad 1, 3 e 7 giorni dall'evento al variare della lunghezza delle sequenze. Variabili usate: tensione della cella minima, potenza del pannello, SOC, irraggiamento e bilancio delle correnti della batteria 95](#_Toc104296888)

[Figura 6‑4: Confronto delle accuratezze di previsione per le varie combinazioni di variabili con sequenze lunghe 1 giorno 97](#_Toc104296889)

[Figura 6‑5: Confronto delle accuratezze di previsione per le varie combinazioni di variabili con sequenze lunghe 3 giorni 98](#_Toc104296890)

[Figura 6‑6:Confronto delle accuratezze ad 1, 3 e 7 giorni dall'evento al variare della lunghezza delle sequenze. Variabili usate: tensione della cella minima e potenza del pannello 119](#_Toc104296891)

[Figura 6‑7:Confronto delle accuratezze ad 1, 3 e 7 giorni dall'evento al variare della lunghezza delle sequenze. Variabili usate: tensione della cella minima, potenza del pannello, SOC e irraggiamento 120](#_Toc104296892)

[Figura 6‑8:Confronto delle accuratezze ad 1, 3 e 7 giorni dall'evento al variare della lunghezza delle sequenze. Variabili usate: tensione della cella minima, potenza del pannello, SOC, irraggiamento e bilancio delle correnti della batteria 122](#_Toc104296893)

[Figura 6‑9: Confronto delle accuratezze di previsione per le varie combinazioni di variabili con sequenze lunghe 1 giorno 124](#_Toc104296894)

[Figura 6‑10: Confronto delle accuratezze di previsione per le varie combinazioni di variabili con sequenze lunghe 3 giorni 125](#_Toc104296895)

[Figura 6‑11: Confronto delle accuratezze ad 1, 3 e 7 giorni dall'evento al variare della lunghezza delle sequenze. Variabili usate: tensione della cella minima e potenza del pannello 145](#_Toc104296896)

[Figura 6‑12: Confronto delle accuratezze ad 1, 3 e 7 giorni dall'evento al variare della lunghezza delle sequenze. Variabili usate: tensione della cella minima, potenza del pannello, SOC e irraggiamento 146](#_Toc104296897)

[Figura 6‑13: Confronto delle accuratezze ad 1, 3 e 7 giorni dall'evento al variare della lunghezza delle sequenze. Variabili usate: tensione della cella minima, potenza del pannello, SOC, irraggiamento e bilancio delle correnti della batteria 148](#_Toc104296898)

[Figura 6‑14: Confronto delle accuratezze di previsione per le varie combinazioni di variabili con sequenze lunghe 1 giorno 150](#_Toc104296899)

[Figura 6‑15: Confronto delle accuratezze di previsione per le varie combinazioni di variabili con sequenze lunghe 3 giorni 151](#_Toc104296900)

[Figura 6‑16: Confronto delle accuratezze per le 3 reti neurali usate al variare dell'intervallo di previsione 155](#_Toc104296901)

[Figura 8‑1: Creazione di una struttura con celle contenenti ognuna i dati relativi ad una sola variabile 159](#_Toc104296902)

[Figura 8‑2: Interpolazione dei campioni in diagnostica 159](#_Toc104296903)

[Figura 8‑3: Sovrascrittura dei valori invariati 159](#_Toc104296904)

[Figura 8‑4: Funzione di sovra campionamento 160](#_Toc104296905)

[Figura 8‑5: allineamento delle sequenze temporali 160](#_Toc104296906)

[Figura 8‑6: Struttura dati finale 160](#_Toc104296907)

[Figura 8‑7: Definizione dei parametri come variabili globali 161](#_Toc104296908)

[Figura 8‑8: Main 161](#_Toc104296909)

[Figura 8‑9: funzione per scandire i giorni 162](#_Toc104296910)

[Figura 8‑10: Creazione delle sequenze da 3 giorni 162](#_Toc104296911)

[Figura 8‑11: Suddivisione delle sequenze tra sane e patologiche 163](#_Toc104296912)

[Figura 8‑12: Estrazione delle sequenze predittive 163](#_Toc104296913)

[Figura 8‑13: Estrazione casuale delle sequenze sane 163](#_Toc104296914)

[Figura 8‑14: Funzione di normalizzazione 164](#_Toc104296915)

[Figura 8‑15: Etichettatura del dataset 164](#_Toc104296916)

[Figura 8‑16: Creazione del dataset 165](#_Toc104296917)

[Figura 8‑17: Partizione statica 165](#_Toc104296918)

[Figura 8‑18: K-Folding 166](#_Toc104296919)

[Figura 8‑19: Layers della rete neurale 166](#_Toc104296920)

[Figura 8‑20: Layers della prima rete neurale alternativa 167](#_Toc104296921)

[Figura 8‑21: Layers della seconda rete neurale alternativa 167](#_Toc104296922)

# Indice delle tabelle

[Tabella 3‑1: Caratteristiche hardware degli apparati 29](#_Toc104296923)

[Tabella 5‑1: Parametri della rete 67](#_Toc104296924)

[Tabella 5‑2: Training options 68](#_Toc104296925)

[Tabella 6‑1: Confusion Matrix per predizioni con anticpo di 1 giorno e sequenze lunghe 3 giorni 74](#_Toc104296926)

[Tabella 6‑2: Confusion matrix per predizioni con anticipo di 1 giorno e sequenze lunghe 1 giorno 75](#_Toc104296927)

[Tabella 6‑3: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 3 giorni 76](#_Toc104296928)

[Tabella 6‑4: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 1 giorno 77](#_Toc104296929)

[Tabella 6‑5: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 3 giorni 78](#_Toc104296930)

[Tabella 6‑6: Confusion matrixper predizioni con anticipo di 7 giorni e sequenze lunghe 1 giorno 79](#_Toc104296931)

[Tabella 6‑7: Confusion matrix per predizioni con anticipo di 1 giorno e sequenze lunghe 3 giorni 80](#_Toc104296932)

[Tabella 6‑8: Confusion matrix per predizioni con anticpo di 1 giorno e sequenze lunghe 1 giorno 81](#_Toc104296933)

[Tabella 6‑9: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 3 giorni 82](#_Toc104296934)

[Tabella 6‑10: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 1 giorno 83](#_Toc104296935)

[Tabella 6‑11: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 3 giorni 84](#_Toc104296936)

[Tabella 6‑12: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 1 giorno 85](#_Toc104296937)

[Tabella 6‑13: Confusion matrix per predizioni con anticipo di 1 giorno e sequenze lunghe 3 giorni 86](#_Toc104296938)

[Tabella 6‑14: Confusion matrix per predizioni con anticipo di 1 giorno e sequenze lunghe 1 giorno 87](#_Toc104296939)

[Tabella 6‑15: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 3 giorni 88](#_Toc104296940)

[Tabella 6‑16: Confusion matrix per predizioni con anticpo di 3 giorni e sequenze lunghe 1 giorno 89](#_Toc104296941)

[Tabella 6‑17: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 3 giorni 90](#_Toc104296942)

[Tabella 6‑18: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 1 giorno 91](#_Toc104296943)

[Tabella 6‑19: Tabella delle accuratezze ottenute con la tensione della cella minima e la potenza del pannello 92](#_Toc104296944)

[Tabella 6‑20: Tabella delle accuratezze ottenute con la tensione della cella minima, la potenza del pannello, il SOC e l’irraggiamento 93](#_Toc104296945)

[Tabella 6‑21: Tabella delle accuratezze ottenute con la tensione della cella minima, la potenza del pannello, il SOC, l’irraggiamento e la corrente della batteria 95](#_Toc104296946)

[Tabella 6‑22: Accuratezze al variare dell'intervallo di previsione con varie combinazioni di variabili e sequenze lunghe 1 giorno 96](#_Toc104296947)

[Tabella 6‑23: Accuratezze al variare dell'intervallo di previsione con varie combinazioni di variabili e sequenze lunghe 1 giorno 98](#_Toc104296948)

[Tabella 6‑24: Confusion Matrix per predizioni con anticpo di 1 giorno e sequenze lunghe 3 giorni 101](#_Toc104296949)

[Tabella 6‑25: Confusion matrix per predizioni con anticipo di 1 giorno e sequenze lunghe 1 giorno 102](#_Toc104296950)

[Tabella 6‑26: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 3 giorni 103](#_Toc104296951)

[Tabella 6‑27: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 1 giorno 104](#_Toc104296952)

[Tabella 6‑28: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 3 giorni 105](#_Toc104296953)

[Tabella 6‑29: Confusion matrixper predizioni con anticipo di 7 giorni e sequenze lunghe 1 giorno 106](#_Toc104296954)

[Tabella 6‑30: Confusion matrix per predizioni con anticipo di 1 giorno e sequenze lunghe 3 giorni 107](#_Toc104296955)

[Tabella 6‑31: Confusion matrix per predizioni con anticipo di 1 giorno e sequenze lunghe 1 giorno 108](#_Toc104296956)

[Tabella 6‑32: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 3 giorni 109](#_Toc104296957)

[Tabella 6‑33: Confusion matrix per predizioni con anticpo di 3 giorni e sequenze lunghe 1 giorno 110](#_Toc104296958)

[Tabella 6‑34: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 3 giorni 111](#_Toc104296959)

[Tabella 6‑35: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 1 giorno 112](#_Toc104296960)

[Tabella 6‑36: Confusion matrix per predizioni con anticipo di 1 giorno e sequenze lunghe 3 giorni 113](#_Toc104296961)

[Tabella 6‑37: Confusion matrix per predizioni con anticipo di 1 giorno e sequenze lunghe 1 giorno 114](#_Toc104296962)

[Tabella 6‑38: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 3 giorni 115](#_Toc104296963)

[Tabella 6‑39: Confusion matrix per predizioni con anticpo di 3 giorni e sequenze lunghe 1 giorno 116](#_Toc104296964)

[Tabella 6‑40: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 3 giorni 117](#_Toc104296965)

[Tabella 6‑41: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 1 giorno 118](#_Toc104296966)

[Tabella 6‑42: Tabella delle accuratezze ottenute con la tensione della cella minima e la potenza del pannello 119](#_Toc104296967)

[Tabella 6‑43: Tabella delle accuratezze ottenute con la tensione della cella minima, la potenza del pannello, SOC e irraggiamento 120](#_Toc104296968)

[Tabella 6‑44: Tabella delle accuratezze ottenute con la tensione della cella minima, la potenza del pannello, SOC, irraggiamento e bilancio delle correnti della batteria 121](#_Toc104296969)

[Tabella 6‑45: Accuratezze al variare dell'intervallo di previsione con varie combinazioni di variabili e sequenze lunghe 1 giorno 123](#_Toc104296970)

[Tabella 6‑46: Accuratezze al variare dell'intervallo di previsione con varie combinazioni di variabili e sequenze lunghe 3 giorni 124](#_Toc104296971)

[Tabella 6‑47: Confusion Matrix per predizioni con anticpo di 1 giorno e sequenze lunghe 3 giorni 127](#_Toc104296972)

[Tabella 6‑48: Confusion matrix per predizioni con anticipo di 1 giorno e sequenze lunghe 1 giorno 128](#_Toc104296973)

[Tabella 6‑49: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 3 giorni 129](#_Toc104296974)

[Tabella 6‑50: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 1 giorno 130](#_Toc104296975)

[Tabella 6‑51: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 3 giorni 131](#_Toc104296976)

[Tabella 6‑52: Confusion matrixper predizioni con anticipo di 7 giorni e sequenze lunghe 1 giorno 132](#_Toc104296977)

[Tabella 6‑53: Confusion matrix per predizioni con anticipo di 1 giorno e sequenze lunghe 3 giorni 133](#_Toc104296978)

[Tabella 6‑54: Confusion matrix per predizioni con anticpo di 1 giorno e sequenze lunghe 1 giorno 134](#_Toc104296979)

[Tabella 6‑55: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 3 giorni 135](#_Toc104296980)

[Tabella 6‑56: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 1 giorno 136](#_Toc104296981)

[Tabella 6‑57: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 3 giorni 137](#_Toc104296982)

[Tabella 6‑58: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 1 giorno 138](#_Toc104296983)

[Tabella 6‑59: Confusion matrix per predizioni con anticipo di 1 giorno e sequenze lunghe 3 giorni 139](#_Toc104296984)

[Tabella 6‑60: Confusion matrix per predizioni con anticipo di 1 giorno e sequenze lunghe 1 giorno 140](#_Toc104296985)

[Tabella 6‑61: Confusion matrix per predizioni con anticipo di 3 giorni e sequenze lunghe 3 giorni 141](#_Toc104296986)

[Tabella 6‑62: Confusion matrix per predizioni con anticpo di 3 giorni e sequenze lunghe 1 giorno 142](#_Toc104296987)

[Tabella 6‑63: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 3 giorni 143](#_Toc104296988)

[Tabella 6‑64: Confusion matrix per predizioni con anticipo di 7 giorni e sequenze lunghe 1 giorno 144](#_Toc104296989)

[Tabella 6‑65: Tabella delle accuratezze ottenute con la tensione della cella minima e potenza del pannello 145](#_Toc104296990)

[Tabella 6‑66: Tabella delle accuratezze ottenute con la tensione della cella minima, potenza del pannello, SOC e irraggiamento 146](#_Toc104296991)

[Tabella 6‑67: Tabella delle accuratezze ottenute con la tensione della cella minima, la potenza del pannello, SOC, irraggiamento e bilancio delle correnti della batteria 148](#_Toc104296992)

[Tabella 6‑68: Accuratezze al variare dell'intervallo di previsione con varie combinazioni di variabili e sequenze lunghe 1 giorno 149](#_Toc104296993)

[Tabella 6‑69: Accuratezze al variare dell'intervallo di previsione con varie combinazioni di variabili e sequenze lunghe 3 giorni 150](#_Toc104296994)

[Tabella 6‑70: Presentazione delle migliori accuratezze per ogni intervallo di previsione in base alla lunghezza delle sequenze e alla combinazione di variabili 152](#_Toc104296995)

[Tabella 6‑71: Presentazione delle migliori accuratezze per ogni intervallo di previsione in base alla lunghezza delle sequenze e alla combinazione di variabili 153](#_Toc104296996)

[Tabella 6‑72: Presentazione delle migliori accuratezze per ogni intervallo di previsione in base alla lunghezza delle sequenze e alla combinazione di variabili 154](#_Toc104296997)

[Tabella 7‑1: Ricapitolativo delle migliori accuratezze ottenute al variare dell'intervallo di previsione in base alla composizione della rete neurale, la combinazione di variabili e la lunghezza delle sequenze 157](#_Toc104296998)

# Bibliografia

1. Carvalho, Soaresa, Vita, Francisco, Bastoc, Alcaláb (2019). “A systematic literature review of machine learning methods applied to predictive maintenance “.
2. Mele (2021). “Sistemi SCADA“, Università degli studi di Napoli Federico II, in <https://www.docenti.unina.it/webdocenti-be/allegati/materiale-didattico/34275901> (visitato il 12 maggio 2022).
3. Vita, Bruneo (2019). “On the use of LSTM networks for Predictive Maintenance in Smart Industries”.
4. Texas Instruments, “ Using Deep Learning for Predictive Maintenance “, in <https://training.ti.com/sites/default/files/docs/using-deep-learning-for-predictive-maintenance-slides.pdf> (visitato il 20 aprile 2022).
5. Zhang, Yang, Wang (2019). “Data-Driven Methods for Predictive Maintenance of Industrial Equipment : A Survey.”.
6. Susto, Schirru, Pampuri, McLoone, Beghi (2015). ”Machine Learning for Predictive Maintenance : A Multiple Classifier Approach”.
7. Tsui, Chen, Zhou, Hai, Wang (2014). “Prognostics and Health Management : A Review on Data Driven Approaches”.
8. Finch (2009). “Toyota sudden acceleration: a case study of the national highway traffic safety administration-recalls for change” Loyola Consumer Law Review, vol. 22, p. 472.
9. <https://smartme.io/blog/conosciamo-il-protocollo-di-comunicazione-lora/> (visitato il 20 aprile 2022).

1. <https://www.vodafone.it/portal/Privati/Supporto/Glossario/lte> (visitato il 20 aprile 2022).
2. Marques (2014). “Battery Management System (BMS) for Lithium-Ion Batteries”.
3. Parisi (2007). “Reti neurali e vita artificiale“, Enciclopedia della Scienza e della Tecnica, in <https://www.treccani.it/enciclopedia/reti-neurali-e-vita-artificiale_%28Enciclopedia-della-Scienza-e-della-Tecnica%29/> (visitato il 20 aprile 2022).
4. Panella. “Soft Computing & Computational Intelligence”, Circuiti e Algoritmi per il Calcolo Distribuito, La Sapienza.
5. Panella, “Deep Learning and Deep Neural Networks”, Circuiti e Algoritmi per il Calcolo Distribuito, La Sapienza.
6. Pascanu, Mikolov, Bengio (2013). “On the difficulty of training recurrent neural networks”.
7. Saxena, “Introduction to Softmax for Neural Network“, in <https://www.analyticsvidhya.com/blog/2021/04/introduction-to-softmax-for-neural-network/>
8. Brownlee, “Rectified linear unit“, in <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/>
9. Elsheikh, Yacout, Ouali (2018). “Bidirectional handshaking LSTM for remaining useful life prediction”.
10. Phi (2018). “Illustrated Guide to LSTM’s and GRU’s: A step by step explanation”, in <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21> (visitato il 28 aprile 2022).
11. Zhang, Lipton, Li, Smola (2021). “Bidirectional neural network“. “Dive into Deep Learning “, E-Book in <https://d2l.ai/chapter_recurrent-modern/bi-rnn.html>
12. UNE-EN 13306 :2018. Estratto in <https://www.certifico.com/marcatura-ce/documenti-marcatura-ce/79-documenti-riservati-marcatura-ce/8102-manutenzione-terminologia-uni-en-13306-2018> (visitato il 15 aprile 2022).
13. Araby, Elhameed, Magdy, Said, Abdelaal, Allah, Darweesh, Fahim, Mostafa (2019). “Smart IoT Monitoring System for Agriculture with Predictive Analysis”
14. Gerum, Altay, Gürsoy (2019). “Data-driven predictive maintenance scheduling policies for railways”.
15. Srikanth Namuduri et al (2020). ”Review - Deep Learning Methods for Sensor Based Predictive Maintenance and Future Perspectives for Electrochemical Sensors”.
16. Yuan, Wu, Lin (2016). “Fault diagnosis and remaining useful life estimation of aero engine using LSTM network”
17. Singh, “Fully Connected Layer : The brute force layer of a Machine Learning model”, OpenGenus Foundation, in <https://iq.opengenus.org/fully-connected-layer/> (visitato il 3 maggio 2022)
18. Documentazione Matlab, <https://it.mathworks.com/help/deeplearning/ref/classificationlayer.html;jsessionid=e487a09d04fbd162c6c53b72658a> (visitato il 4 maggio 2022)
19. Brounlee (2020). “A Gentle Introduction to k-fold Cross-Validation”, in <https://machinelearningmastery.com/k-fold-cross-validation/> (visitato il 4 maggio 2022).
20. Kingma, Ba (2015). “Adam : A method for stochastic optimization“.