



UNIVERSITÀ DEGLI STUDI DI MESSINA

*Dipartimento di Scienze Cognitive, Psicologiche, Pedagogiche e degli Studi
Culturali*

DOTTORATO DI RICERCA IN SCIENZE COGNITIVE

Ciclo XXXV-PON

La via cognitiva alla manutenzione preventiva
nei sistemi di energie rinnovabili

Settore Scientifico Disciplinare di Afferenza M-PSI/01

Coordinatore:

Chiar.ma Prof. Alessandra Falzone

Tutor:

Chiar.mo Prof. Alessio Plebe

Presentata da:

Lucia Guerrisi

Anno Accademico 2021-2022

INDICE

INTRODUZIONE

CAPITOLO 1

La via cognitiva della manutenzione di impianti energetici	p. 9
1.1 Deep Learning e Intelligenza Artificiale	p. 10
1.2 Il Deep Learning e il suo successo applicativo	p. 25
1.3 Come sviluppare modelli Deep Learning	p. 30

CAPITOLO 2

Energia eolica e manutenzione	p. 33
2.1 Manutenzione predittiva	p. 33
2.2 Reti neurali artificiali per manutenzione di turbine eoliche	p. 37
2.3 Prototipo di predizione neurale su dati simulati	p. 40

CAPITOLO 3

Energia solare e manutenzione	p. 51
3.1 La manutenzione predittiva negli impianti fotovoltaici	p. 51
3.1.1 Che cos'è un Inverter?	p. 54
3.1.2 Reti neurali per la manutenzione degli impianti fotovoltaici	p. 55
3.2 Sperimentazione preliminare di previsione guasti in impianti fotovoltaici	p. 57
3.2.1 Collaborazione con Ecoprime	p. 57
3.2.2 Interrogazione meteocontrol.de	p. 61
3.2.3 Interfaccia dei dati	p. 66
3.2.4 Modello	p. 67
3.2.5 Risultati	p. 70

CAPITOLO 4	
Corrosione dei materiali: una fonte di degrado	p. 80
4.1 Modello deep learning per il filtraggio del rumore	p. 82
4.1.1 Collaborazione con UCT di Praga	p. 82
4.1.2 Descrizione del modello	p. 86
4.1.3. Descrizione dei passaggi	p. 91
4.1.4 Da excel a file CSV	p. 91
4.1.5 Da file CSV a file HDF5	p. 91
4.1.6 Addestramento del modello deep ed <i>evaluation</i>	p. 97
4.1.7 Risultati	p. 98
CONCLUSIONI	p. 109
BIBLIOGRAFIA	p. 113

Introduzione

La ricerca affrontata durante questo triennio di dottorato, e descritta in questa tesi, fa parte delle iniziative PON, Azione 1.I Dottorati Innovativi con Caratterizzazione Industriale, e in particolare costituisce l'attuazione della proposta di progetto intitolata "Big Data Analytics for Predictive Maintenance in Renewable Energy Power Plants". Si ritiene utile partire da una giustificazione riguardo l'attinenza di un progetto di questo genere, con la Scuola di Dottorato in Scienze Cognitive, a cui fa capo la ricerca qui descritta, nonché con la personale formazione in psicologia della scrivente. È evidente come l'oggetto finale della ricerca, ovvero la problematica della manutenzione di impianti di energie rinnovabili, si configuri nettamente come inerente all'ingegneria, e pertanto apparirebbe come di competenze di quell'ambito, evidentemente lontano dalle scienze cognitive.

L'appropriatezza di questo progetto all'interno del dottorato in scienze cognitive, apparentemente anomala, deriva dal genere di soluzione che è stato proposto per la problematica della manutenzione. Ovvero, l'adozione di metodi previsionali che derivano dalla cognizione naturale, e da come viene realizzata nel cervello, in particolare da come, in senso ampio, la cognizione è in grado di formulare scenari futuri dello stato dell'ambiente circostante, partendo dallo stato attuale a quelli pregressi. Questa motivazione sulla pertinenza dell'ambito delle scienze cognitive rispetto al progetto sopra citato, inevitabilmente ne circoscrive subito anche le direzioni di indagine. Contrariamente ad un convenzionale ciclo di sviluppo di una ricerca ingegneristica, risulta qui superflua una preliminare ricognizione dello stato dell'arte sulle possibili alternative da adottare, e una eventuale loro comparazione a livello prototipale, giacché è immediatamente la sola direzione dell'emulazione cognitiva a costituire la direttiva da percorrere, senza interesse alle alternative tradizionali. Si tratta, tuttavia, di una limitazione ragionevole, considerando che da

oltre un decennio l'imitazione della cognizione ha messo sul campo uno degli strumenti informatici più potenti e di più ampio spettro mai prima disponibile, che va sotto il nome di Deep Learning. Esiste dunque una ragionevole confidenza su un ridotto rischio di ricerca nel percorrere una direzione univoca, segnata dalla famiglia di tecniche che fanno capo al Deep Learning.

Il primo capitolo della tesi è dedicato proprio al Deep Learning, delineandone anzitutto la sua storica derivazione dalla cognizione, anzi dalla “microstruttura della cognizione”, citando il titolo del celebre volume dei primi ideatori delle reti neurali artificiali, David Rumelhart e James McClelland; contestualizzando la sua collocazione all'interno dell'Intelligenza Artificiale; ripercorrendo la sua recente transizione alla forma “deep” e il conseguente inatteso successo in una moltitudine di ambiti applicativi. Si tratta necessariamente di una estrema sintesi, che dà maggiore respiro ad alcune tipologie di reti neurali artificiali, la cui imitazione di funzioni cognitive risulterà particolarmente proficua nelle applicazioni qui di interesse. L'ultima sezione del capitolo ha carattere tecnico, ed illustra i modi possibili con cui oggi il concetto di rete neurale artificiale si concretizza in implementazioni, grazie ad uno dei possibili sistemi di sviluppo specifici per Deep Learning oggi disponibili. Questa sezione oltre al suo scopo illustrativo sulla metodologia concreta messa in campo nel progetto, sintetizza anche il percorso formativo attraversato dalla scrivente, per poter acquisire la capacità di realizzare in software i vari modelli neurali ideati per gli scopi della ricerca.

Mentre lo strumento utilizzato nella ricerca è sostanzialmente uno solo, la modellistica tramite reti neurali artificiali, come appena detto, il caso applicativo si è invece andato via via articolandosi nel corso del dottorato, e di conseguenza anche il genere di modello, pur se sempre nella categoria generale del Deep Learning, si è via via diversificato in varie sue istanze.

L'ambito applicativo affrontato per primo riguarda i sistemi di produzione energetica rinnovabile con l'utilizzo di turbine eoliche, della cui manutenzione si occupa il partner industriale) inizialmente partecipante al progetto. Conseguentemente, una parte della ricerca ha riguardato la necessaria presa di conoscenza sul quadro dei malfunzionamenti a cui vanno soggetti questi tipi di impianti, e sul genere di informazioni sensoriali disponibili, su cui poter costruire modelli previsionali. È l'argomento del secondo capitolo, in cui viene preliminarmente illustrata la modalità di produzione energetica di tipo eolica e le sue tipiche problematiche di manutenzione; viene riportata una ricognizione in letteratura sull'impiego del Deep Learning in questo ambito; descritto infine il modello prototipale sviluppato su una simulazione realistica di turbine eoliche. Fin da subito si sono evidenziati due elementi, tra di loro connessi: la scarsa e ancora immatura penetrazione delle reti neurali artificiali nel settore, e l'oggettiva difficoltà alla loro adozione derivante da una ridotta disponibilità di dati storici. Il Deep Learning funziona molto bene dovunque esista una casistica di esempi ampia e generalizzata, diventa via via più problematico da impiegare tanto più i dati sono pochi e scarsamente rappresentativi. La disponibilità dei dati è stato anche il motivo di cambiamento di partner industriale nel progetto, e conseguente cambiamento di tipologia di impianto energetico preso a caso di studio. In questo caso il Deep Learning non c'entra nulla, vi sono state problematiche relative alla pubblicità dei dati, che hanno costretto a lavorare solamente su dati simulati, fondamentali per una fase prototipale, ma penalizzanti nel voler indagare in modo rigoroso ed oggettivo sulla applicabilità dei modelli Deep Learning in ambito di manutenzione preventiva.

Si è pertanto provveduto all'individuazione di un nuovo partner industriale, la cui collaborazione è stata autorizzata dal Ministero dell'Università e Ricerca Scientifica nel giugno 2021. La tipologia principale di impianti energetici gestiti da questa azienda ricade nel fotovoltaico, divenuto quindi il nuovo obiettivo applicativo, a cui è qui

dedicato il terzo capitolo. La sua articolazione non è dissimile dal secondo, iniziando con una breve introduzione a questa forma di produzione di energia rinnovabile, e alle sue principali fonti di possibili malfunzionamenti; proseguendo con una ricognizione di letteratura sull'applicazione di modelli neurali artificiali in questo ambito; illustrando il modello prototipale ideato su una prima semplice simulazione di impianto e suoi malfunzionamenti. In questo caso però si è potuti andare ben oltre, progettando un modello atto a funzionare con dati reali degli impianti, come descritto nell'ultima sezione, che comprende diversi inevitabili dettagli tecnici sull'interfacciamento tra il modello neurale e il sistema di collezione dei dati sensoriali degli impianti. Il caso degli impianti fotovoltaici ha ribadito il limite critico di applicabilità del Deep Learning alla predizione di malfunzionamenti, derivante dalla scarsa disponibilità di dati. Stavolta la disponibilità del partner aziendale è stata piena, mettendo a disposizione l'intero loro archivio costituito da quasi 20,000 interventi di manutenzione. È però presto emerso che la maggior parte degli interventi derivano da una miriade di fattori eterogenei, ben pochi dei quali riconducibili a degradi di componenti critici che abbiano qualche riflesso nei dati sensoriali misurati. Una selezione mirata della tipologia di componente più frequentemente indiziata di malfunzionamento, l'inverter, ha permesso di restringere in modo oculato il campo, al prezzo però di non avere più di qualche decina di campioni. Si sono tuttavia ottenuti dei risultati sperimentali che, pur con molta cautela, sembrano indicare il metodo come promettente.

Trasversalmente agli obiettivi costituiti dalle due tipologie di impianti energetici sopra descritti, esiste un altro fattore di possibili malfunzionamenti, che è stato oggetto applicativo di studio in questa ricerca. Per qualunque installazione tecnologica in cielo aperto, una delle possibili fonti di degrado è costituita dalla corrosione, soprattutto a carico degli elementi metallici, dovuta all'ossigeno nell'atmosfera. Questo vale naturalmente anche per le turbine eoliche e gli impianti

fotovoltaici. Si tratta quindi di un terzo ambito, non più costituito da una tipologia di impianto, ma da un fattore generico di guasti ad impianti. In questo caso non si tratta di una scelta maturata nel corso del progetto, ma di una specifica all'interno della proposta di progetto PON, da cui deriva questa ricerca di dottorato. Se la competenza dettagliata sui sistemi di produzione di energia era del tutto ingegneristica, e lontana dall'ambito delle scienze cognitive, lo è ancor di più quella sui fenomeni corrosivi, che riguarda la chimica e la scienza dei materiali. Questa carenza non è stata lesiva della ricerca, in quanto la proposta di progetto PON includeva un rapporto di collaborazione con un partner accademico, depositario della più elevata competenza a livello europeo sui fenomeni corrosivi. Si tratta del Department of Metallic Construction Materials at Technopark Kralupy presso la University of Chemistry and Technology (UCT) Prague, e il riferimento scientifico è proprio il direttore, Tomas Prosek, autorità internazionale sulla corrosione nei metalli. L'argomento corrosione, e la sperimentazione di una modellistica tramite Deep Learning come contributo ad un caso di studio, costituiscono il quarto capitolo. La collaborazione con la UCT Prague è stata fruttifera e anche reciproca, così come per la scrivente e il dottorato a cui afferisce la corrosione era una fetta di scienza sconosciuta, altrettanto per loro erano sconosciuti i metodi basati su reti neurali artificiali e Deep Learning in particolare, ed incuriositi sul loro utilizzo. Lo stesso Prosek ha suggerito come caso il loro attuale sviluppo di un sensore di corrosione innovativo, in grado di offrire diversi vantaggi ma ancora afflitto da problemi di rumore nella misura. Si è pertanto ideato e sperimentato un modello neurale artificiale che fungesse da correttore della misura grezza del sensore, fornendone una più vicina alla reale entità di corrosione. Pur trattandosi di una problematica diversa dalle due descritte precedentemente, anche qui la criticità principale nell'uso del Deep Learning è stata analoga: la limitatezza dei campioni storici a disposizione. Trattandosi di un sensore nuovo, ed essendo particolarmente laborioso effettuare simulazioni controllate di corrosione, le serie di

dati sperimentali a disposizioni erano limitate, e riguardavano una varietà ampia di leghe metalliche, ciascuna con comportamenti diversi. Anche in questo caso, comunque, il modello ha dimostrato una leggera superiorità nel correggere le misure del sensore, rispetto al metodo matematico che UCT Prague aveva utilizzato finora.

Capitolo 1

La via cognitiva

Il presente capitolo intende delineare gli ambiti disciplinari che sottendono il progetto di ricerca proposto. Inizialmente si propone la metodologia del *Deep Learning*, possibile soluzione alla problematica da affrontare nel lavoro di ricerca che si pone come obiettivo la manutenzione predittiva di impianti coinvolti nella produzione di energie rinnovabili.

Ed è proprio questa metodologia a costituire quella che è stata qui indicata come "via cognitiva" per affrontare la problematica della manutenzione degli impianti di energie rinnovabili, e il filo conduttore risiede nella capacità di predizione, principio cardine del funzionamento della cognizione di ogni organismo vivente, e elemento chiave per venire incontro alle esigenze di pianificazione della manutenzione. La storia da cui proviene il deep learning, qui velocemente sintetizzata, è l'impresa intellettuale di utilizzare quel che si è venuti a conoscere sulla cognizione, e sul suo realizzarsi nel cervello, come strada privilegiata per formulare modelli computazionali in grado di risolvere una moltitudine di problemi complessi. In 1.1 si presenterà la progressiva rinascita dell'intelligenza artificiale e si descriveranno i primi modelli neurali, dal Perceptron di Rosenblatt fino ai nuovi modelli deep. In 1.2 si mostrerà il successo applicativo del Deep Learning e i diversi campi d'utilizzo. Verrà anticipato come certi campi di utilizzo, pur essendo così vicini alla cognizione umana quanto lontani dalla manutenzione di impianti energetici, possano risultare percorribili, grazie alla condivisione di certe problematiche di fondo, e dalla loro soluzione ispirata cognitivamente. Infine, in 1.3 si descriveranno gli strumenti opportuni utili a

sviluppare un modello neurale e le piattaforme create per facilitare lo sviluppo di modelli Deep Learning, in particolare Tensorflow.

1.1 Deep Learning e Intelligenza Artificiale

I propose to consider the question, "Can machines think?"
This should begin with definitions of the meaning of the terms "machine" and "think". The definitions might be framed so as to reflect so far as possible the normal use of the words, but this attitude is dangerous.

(Alan M. Turing, Computing Machinery and Intelligence.
«Mind» 49, 1950: p. 433)

Il presente paragrafo descrive la rivoluzione che il *Deep Learning* ha portato all'Intelligenza Artificiale (IA), ed è proprio a questa nuova famiglia di algoritmi che si intende affidare la problematica della manutenzione predittiva nell'ambito dell'energia rinnovabile.

Tale problematica, come verrà chiarito nella sezione 1.2, è prettamente ingegneristica, e in questa prospettiva l'approccio abituale avrebbe previsto una sistematica rassegna di tutte le metodologie impiegabili, seguita da una motivata selezione di un loro sottoinsieme in modo da effettuare una comparazione empirica. Questo prevedeva comunque un'ampia conoscenza ingegneristica degli apparati delle strutture coinvolte nella produzione di energia rinnovabile.

In questa tesi, invece, la metodologia ricercata e prevista è una sola.

È opportuno chiarire i motivi di questa scelta. Il lavoro qui descritto è un progetto di dottorato in scienze cognitive, contesto in cui non è tanto di rilievo la problematica ingegneristica di per sé, quanto la sua suscettibilità a essere risolta mediante una metodologia — il *Deep Learning* — che ha le sue radici proprio nella ricerca cognitiva empirica.

La seconda motivazione è che il *Deep Learning* rappresenta pienamente un'alternativa a metodi tradizionali, e di ciò si parlerà più diffusamente più avanti. Qui si intende descrivere il *Deep Learning* e le sue origini, perciò occorre innanzitutto

staccare la corrente di questa nuova metodologia da una prima generazione di IA e dai differenti obiettivi, prima prettamente ancorati a strutture concettuali predeterminate poi a fini ingegneristici che sfruttano le reti neurali artificiali.

Possiamo definire l'intelligenza artificiale come una scienza che studia lo sviluppo di sistemi Hardware e Software dotati di specifiche capacità tipicamente umane, è dunque un campo di ricerca che studia sistemi di progettazioni che mirano ad assegnare alle macchine capacità tipiche dell'essere umano grazie ad algoritmi, tecniche computazionali.

Il concetto di Intelligenza artificiale muove da due distinte teorie:

- IA forte, secondo cui una macchina può replicare l'intelligenza umana.
- IA debole, secondo cui una macchina può svolgere una o più funzioni umane complesse.

Il tentativo di ridurre il ragionamento umano a una macchina calcolatrice risale a Hobbes (1588-1679); nel 1651 con il *Leviatano*¹, paragona il mostro marino del libro di Giobbe alla forma e al potere dello stato ecclesiastico e civile, simbolo del potere assoluto e autoritario dello stato.

Il tentativo da parte degli uomini di costruire ingegnosi strumenti di calcolo risale ai tempi più antichi; Leibniz nel 1671-1674 costruì una macchina in grado di eseguire le quattro operazioni di base dell'aritmetica. Un'altra evoluzione significativa è stata offerta da Blaise Pascal nel 1645 con la Pascalina, la prima addizionatrice meccanica.

Il passaggio dalle macchine calcolatrici a quelle "programmabili" fu fatto da Charles Babbage (1791-1871) che gettò le basi per la nascita dell'informatica.

Progettando nel 1822 un'apparecchiatura meccanica chiamata *Difference Engine* (macchina differenziale), in grado di calcolare funzioni polinomiali fino al sesto ordine.

¹ Hobbes, T. (1651). *Leviathan*. London. English trans. by Edwin Curley, 1994, Indianapolis: Hackett.

Il termine “differenziale” deriva dal metodo delle differenze divise per l'interpolazione dei coefficienti polinomiali.

Nel 1935 Alan Turing (1912-1954) propone un modello ideale di calcolatore universale, offrendo una classe di dispositivi computazionali, macchine calcolatrici astratte definite a prescindere dall'hardware utilizzato. Turing aveva concepito in maniera matematica l'idea di un sistema che fosse in grado di fare quello che oggi fa un calcolatore. L'idea era del tutto astratta ma ben strutturata e rimane ancora oggi il modello universale per qualunque computer.

Il primo articolo di Turing risale a quando aveva 24 anni, ed è intitolato *On Computable Numbers, with an Application to the Entscheidungsproblem* (1936)², ed proprio l' *Entscheidungsproblem*³ che Turing con il modello astratto vuole risolvere fornendo un procedimento reale che andasse incontro al problema della “decidibilità”, dunque avere un procedimento unico che sia in grado di decidere se qualunque espressione matematicamente ben formata sia vera o falsa. Quindi essere in grado di poter eseguire qualunque tipo di ragionamento matematico.

La Macchina di Turing manipola i dati contenuti su un nastro potenzialmente infinito, secondo un insieme di regole.

Il modello Macchina di Turing non solo ha consentito la nascita e lo sviluppo dell'informatica, continua ancora oggi a costituire il modello matematico astratto di riferimento per qualunque computer, ad esclusione dei nuovi e ancora pionieristici computer quantistici. Mentre l'articolo del 1936 rimane estremamente tecnico senza avanzare nessuna pretesa su possibile intelligenza di futuri sistemi di calcolo, il tema viene proposto in modo eclatante in quell'articolo il cui passaggio saliente viene riportato in epigrafe di questa

² Turing, A. (1936). *On computable numbers, with an application to the Entscheidungsproblem*. *Proceedings of the London Mathematical Society*, 42: 230– 265.

³ Problema posto da David Hilbert nel 1928.

sezione. L'articolo propone un criterio per discriminare se un computer si possa considerare intelligente oppure no, divenuto celebre come test di Turing, in cui un interlocutore umano conversa in modo remoto col computer, che ottiene il marchio di intelligenza qualora il suo conversare sia indistinguibile da quello di un altro essere umano. Turing non si limita a proporre questo genere di verifica, si sbilancia anche ad ipotizzare che in un futuro non troppo lontano arriveranno computer in grado di superarla, e prende in esame una lunga rassegna di possibili obiezioni che potrebbero essere poste all'idea di una macchina intelligente, fornendo sue argomentazioni in difesa. Per esempio, una delle obiezioni prese in esame da Turing era quella appena circolata mentre stava lavorando al suo articolo.

Geoffrey Jefferson in un suo articolo "The mind of mechanical man" pubblicato nel 1949 sul *British Medical Journal* scrive: "Fino a quando una macchina non potrà scrivere un sonetto o comporre un concerto in base a pensieri ed emozioni provate, e non per la giustapposizione casuale di simboli, non potremo essere d'accordo sul fatto che una macchina eguagli il cervello - cioè, che non solo scriva ma sappia di aver scritto. Nessun meccanismo potrebbe sentire (e non semplicemente segnalare artificialmente, che sarebbe un facile trucco) piacere per i suoi successi, dolore quando una sua valvola fonde, arrossire per l'adulazione, sentirsi depresso per i propri errori, essere attratto dal sesso, arrabbiarsi o abbattersi quando non può ottenere quel che desidera."⁴

La risposta di Turing consiste nell'osservare che l'argomento di Jefferson, portato avanti in modo rigoroso, non può che condurre ad una posizione solipsistica, non avendo modo di accertare sensazioni private di chiunque se non se stessi. Un argomento che anticipa anche una futura e celebre obiezione, da parte di Searle, che esamineremo tra poco.

⁴ Jefferson, G. (1949). The mind of mechanical man. *British Medical Journal*, 1(4616), 1105.

Una sostanziale accettazione della macchina di Turing, ma una costellazione di confutazioni del test di Turing, che prelude all'IA forte, hanno accompagnato la crescita dell'informatica e di conseguenza la gestione e la fruizione delle informazioni. L'attuale centralità dell'informatica deve essere ricondotta al passaggio dal modello analogico al modello digitale nella codifica dei dati, unito alle possibilità offerte dallo sviluppo della tecnologia, la cui evoluzione è stata caratterizzata da tre percorsi: l'estensione del digitale dai soli linguaggi naturali e formali al mondo delle immagini e dei suoni; lo sviluppo delle rappresentazioni avvenuto grazie al miglioramento dei monitor e delle interfacce grafiche; l'integrazione in un unico network dei domini di informazioni sviluppati in precedenza. L'ultimo percorso, caratterizzato dal passaggio dal personal computer a internet, è frutto della crescita e della ricomposizione del dominio digitale in un network sempre più ampio, che consente di sviluppare una versione digitale del sapere umano.⁵

La via aperta da Turing sull'autodidattica dei computers⁶ non è stata subito percorsa e ampliata.

Solo nel 1956, sei anni dopo la Turinghiana frase «I propose to consider the question "Can machines think?"»⁷, John McCarthy, Marvin Minsky, Nathaniel Rochester e Claude Shannon, durante il *Dartmouth Summer Research Project*, segnano l'inizio del traballante progresso iniziale dell'IA che sfocerà, solo dopo il 2000, in strumenti in grado di apprendere spontaneamente dai dati:

"We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or

⁵ Cfr. Enciclopedia filosofica, volume VI, p. 6579, Bompiani.

⁶ Turing, A., 1948. *Intelligent machinery*. Tech. rep., National Physical Laboratory, London, raccolto in Ince, D. C. (ed.) *Collected Works of A. M. Turing: Mechanical Intelligence*, Edinburgh University Press, 1969.

⁷A. Turing, *Computing Machinery and Intelligence*, 1950, *Mind* (vol.49, pp. 433-460).

any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it”⁸.

Dopo il seminario si succedettero molte riflessioni filosofiche relative alla IA, tra queste nel 1957, il General Problem Solver (GPS)⁹, di A. Newell e H. Simon¹⁰. Nello sviluppo dell'IA ha avuto un ruolo centrale la figura di Chomsky sin dagli anni '60, anche se l'intento dello scienziato era diretto alle scienze cognitive. Chomsky¹¹, usò i primi strumenti matematici sviluppati per il linguaggio.

Nel 1964, dieci anni dopo i primi tentativi di “superare” il test di Turing Joseph Weizenbaum (1966) propone “*Eliza*”, il programma più diffuso nella storia dell'IA¹². *Eliza* impersonifica uno psicoterapeuta rogersiano. La scelta dell'orientamento di Carl Rogers (1942) si adatta perfettamente, perché il professionista dovrebbe sempre apparire comprensivo con il paziente, facendo solo domande blande, chiedendo spesso al paziente di chiarire ciò che ha appena detto. Pertanto, il terapeuta non deve impegnarsi in alcun dettaglio con il vero problema del paziente, e *Eliza* può felicemente ignorare tutti i riferimenti al mondo reale nelle frasi del paziente. La contemporanea efficacia di *Eliza* nel suggestionare chi conversava con essa, e la sua palese mancanza di comprensione del linguaggio suscitarono non

⁸ McCarthy, J., Minsky, M. L., Rochester, N. and Shannon, C. E. (1955). A proposal for the Dartmouth summer research project on artificial intelligence, August 31, 1955. reprinted in *AI Magazine*, 27: 12–14.

⁹ Newell, A., Shaw, J. C., & Simon, H. A. (1959, June). Report on a general problem solving program. In *IFIP congress* (Vol. 256, p. 64).

¹⁰ Nel GPS il “problema” è trasformare gli oggetti tramite l'applicazione di operatori appropriati finché non c'è differenza tra lo stato attuale e lo stato desiderato.

Questo processo è moderato dall'euristica che può suggerire l'ordine in cui le possibili soluzioni dovrebbero essere esaminate; può escludere classi di passaggi; può fornire uno screening rapido per selezionare i passaggi più promettenti.

¹¹ Chomsky, N. (1956). Three models for the description of languages. *IRE Transaction on Information Theory*, 2: 113–124.

¹² Wallace, R. S. (2008). The anatomy of A.L.I.C.E. In Epstein et al. (2008), pages 181–210.

poche reazioni nel mondo filosofico dell'epoca, ci limitiamo a menzionare la più celebre di esse, l'esperimento mentale ideato da Searle.

La stanza cinese è un esempio di tentativo di confutazione dell' IA forte. È un esperimento mentale ideato da Searle nel suo articolo "Minds, Brains and Programs"¹³ pubblicato nel 1980:

Suppose that I'm locked in a room and given a large batch of Chinese writing. Suppose furthermore [...] that I know no Chinese [...] Now suppose further that [...] I am given a second batch of Chinese script together with a set of rules for correlating the second batch with the first batch. The rules are in English, and I understand these rules [...]. They enable me to correlate one set of formal symbols with another set of formal symbols [...]. Now suppose also that I am given a third batch of Chinese symbols together with some instructions, again in English, that enable me to correlate elements of this third batch with the first two batches, and these rules instruct me how to give back certain Chinese symbols with certain sorts of shapes in response to certain sorts of shapes given me in the third batch. Unknown to me, the people who are giving me all of these symbols call the first batch "a script," they call the second batch a "story." and they call the third batch "questions." Furthermore, they call the symbols I give them back in response to the third batch "answers to the questions." and the set of rules in English that they gave me, they call "the program."¹⁴

Questo esperimento si baserebbe su una metafora, Searle immagina di essere in una stanza che comunica con l'ambiente esterno soltanto attraverso una feritoia. All'esterno dei cinesi passano a Searle domande in cinese, Searle però, non ha nessuna conoscenza del cinese, ha a disposizione un manuale di istruzioni, scritto in inglese, per combinare i diversi simboli tra di loro, per poi scrivere una risposta corretta ai cinesi fuori dalla stanza. I signori all'esterno, capiscono le risposte, ed essendo tali risposte pienamente sensate, sono certi che nella stanza ci sia qualcuno che, come loro, intende perfettamente il cinese. In realtà, Searle riesce solo scegliere

¹³ J.R. Searle. Minds, brains and programs. Behavioral and Brain Sciences, 1980.

¹⁴ John R. Searle, (1980) Minds, brains, and programs. Behavioral and Brain Sciences 3 (3): p. 3.

correttamente i simboli identificandoli in base a criteri puramente formali: “all that 'formal' means here is that I can identify the symbols entirely by their shapes. (...) As far as the Chinese is concerned, I simply behave like a computer; I perform computational operations on formally specified elements”¹⁵. Al di là del caso Eliza, Searle con questo esperimento mentale ha inaugurato un ventennio, l'ultimo del secolo scorso, di intenso dibattito sulla reale possibilità dell'IA forte¹⁶. Ambito per il quale risulta invece cruciale l'insieme di metodologie che si sono andate sviluppando nell'IA, lungo la cui storia si sono ramificate direzioni di ricerca profondamente diverse ed alternative tra loro. Con una notevole semplificazione si possono rintracciare due filoni prevalenti, in qualche modo riconducibili ad una loro filosofia di fondo, razionalista oppure empirista. Fino al momento dell'articolo di Searle, la linea dominante in IA era quella razionalista, fondata sugli strumenti della logica, mirante alla soluzione di problemi mediante l'istituzione di sistemi di regole, e pienamente coerente con la macchina di Turing quale fondamento algoritmico. Curiosamente Turing stesso aveva gettato le basi per il filone alternativo, quello empirista¹⁷.

- **1.1.1 Il perceptron di Rosenblatt**

Si può dedurre che un primo step di intelligenza artificiale ha una base prettamente razionalista. Presto altre forme filosofiche si avvicinarono alla comunità dell'IA, dando vita a due versanti, uno razionalista ben consolidato in AI e uno empirista.

La visione empirista dell'intelligenza artificiale si collega a una tradizione filosofica di lunga data che ha avuto origine con Aristotele, il quale insisteva sul fatto che la

¹⁵ Paul M. e Patricia S. Churchland, Could a Machine think? in «Scientific America» N. 262, January 1990) in *Le Scienze* Vol. 44 N. 259, Milano Marzo 1990.

¹⁶ Non si aggiunge altro in questa sede (per ulteriori dettagli vedi nota 13), essendo il tema centrale per l'IA, ma non per la sua applicazione in un ambito applicativo come quello della manutenzione, qui di interesse

¹⁷ Turing, A., 1948. *Intelligent machinery*. Tech. rep., National Physical Laboratory, London, raccolto in Ince, D. C. (ed.) *Collected Works of A. M. Turing: Mechanical Intelligence*, Edinburgh University Press, 1969.

conoscenza è basata sull'esperienza piuttosto che su idee innate. Il programma empirista era ripreso nel XVII secolo da alcuni filosofi inglesi, primo da John Locke (1690), sostenendo che la mente viene al mondo come un "armadio vuoto", poi da David Hume (1739), il quale sosteneva che i concetti sono il prodotto di cosa noi percepiamo, e la ragione ottiene tutti i suoi materiali dalla percezione. La battaglia tra il razionalismo e l'empirismo ha attraversato la lunga storia della filosofia occidentale a causa di punti di vista radicalmente diversi. Per i razionalisti, la conoscenza è costruita su un'impalcatura di principi innati mentre per gli empiristi tutta la conoscenza deriva dall'esperienza. Il progetto più influente del versante empirista fu il *Perceptron*, un dispositivo elettronico progettato da Frank Rosenblatt^{18,19} al Cornell Aeronautical Laboratory. L'apprendimento nel perceptron si basa il più possibile su come funziona l'apprendimento umano; Rosenblatt è stato in grado di dimostrare matematicamente il suo funzionamento²⁰ e ha inaugurato l'interazione tra AI e scienze cognitive.

Inizialmente i citati esponenti dell' IA, tra cui Minsky, tendevano ad ancorarsi alla neonata scienza cognitiva²¹, subito dopo il *Perceptron*, la comunità razionalista restituisce al progetto di Rosenblatt una serie di critiche e attacchi (Minsky e Papert²²) che si ridimensioneranno negli anni '80:

"The perceptron has shown itself worthy of study despite (and even because of!) its severe limitations. It has many features to attract attention: its linearity; its intriguing learning theorem; its clear paradigmatic simplicity as a kind of parallel computation. There is no reason to suppose that

¹⁸ Rosenblatt, F. (1959). Two theorems of statistical separability in the perceptron. In *Mechanisation of thought processes: Proceedings of symposium No. 10*, pages 419–472, London. H. M. Stationery Office.

¹⁹ Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organisation in the brain. *Psychological Review*, 65: 386–408.

²⁰ Per una disamina approfondita del Perceptron cfr. Plebe, A., Perconti, P. *The Future of the Artificial Mind*. Taylor&Francis Group, LLC 2022

²¹ Novaes, C. D., 2012. *Formal languages in logic: a philosophical and cognitive analysis*. Cambridge University Press, Cambridge (UK).

²² Minsky, M. and Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge (MA).

any of these virtues carry over to the many-layered version. Nevertheless, we consider it to be an important research problem to elucidate (or reject) our intuitive judgment that the extension is sterile²³.

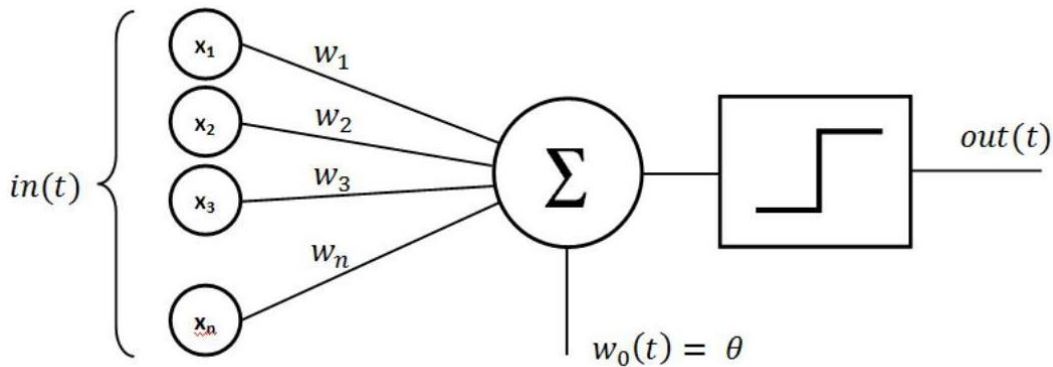


Figure 1.1 Perceptrone di Rosenblatt

Il Perceptron implementava un modo di apprendere che indubbiamente coglieva aspetti dell'apprendimento umano, nell'amplificare le connessioni che conducevano ad un risposta corretta nello strato da lui denominato associativo e lo strato di neuroni in uscita. Tuttavia la sua formulazione consentiva la modifica delle connessioni di un unico strato. Troppo poco per poter realizzare funzioni complesse, lasciando pertanto spazio ai detrattori come Minsky e Papert di dimostrarne limitazioni. Rosenblatt inventò il termine "backpropagation", come soluzione, auspicando una formulazione matematica che fosse in grado di "propagare" all'indietro l'informazione sull'errore prodotto dalla rete neurale nel suo strato di uscita, in modo da poter modificare opportunamente non solo lo strato immediatamente precedente, ma anche tutti gli altri. Venti anni dopo furono altri a scoprire formulazioni in grado di dare forma a quella parola.

Il modello neurale artificiale per il quale si applica la backpropagation introdotta da Rosenblatt, chiamato "feedforward", è costituito da unità semplici organizzate in strati distinti, con collegamenti unidirezionali tra ogni strato e quello successivo.

²³ *Ibidem*, p. 231.

Rumelhart e McClelland quando progettano il *Parallel Distributed Processing* (PDP)²⁴, aprono la strada al prevaricamento del versante razionalista ortodosso delle scienze cognitive da parte di quello empirista²⁵. Infatti il paradigma del PDP prescinde dalla classica struttura della mente e l'apprendimento su cui si fonda è graduale, instabile e si rinnova con l'esperienza.

Come accennato sopra, dopo un lungo periodo in cui l'IA ha subito una lunga discesa, solo dopo diversi anni e grazie all'esigenza di inserire più strati (layers) all'interno di una rete neurale, si è passati a proporre una versione più avanzata dell'IA dove primeggia uno degli autori del PDP project, Geoffrey Hinton che utilizzò per primo il nuovo termine "deep". In cosa consiste la profondità?

La backpropagation - a cui lo stesso Hinton aveva contribuito negli anni '80 - in linea di principio permette l'apprendimento in reti con un numero arbitrario di strati, ma nella pratica era emerso che reti con più di tre strati erano instabili durante l'addestramento, ed erano pertanto evitate. La rete neurale artificiale classica dal 1988 al 2010 aveva sempre tre strati, ed oggi viene denominata "shallow", poco profonda.

Hinton e Salakhutdinov (2006), sono i primi a proporre addestramenti con modelli di quattro o cinque strati nascosti. Questo risultato è stato ottenuto rispolverando la Restricted Boltzmann Machines (RBM). Mentre la backpropagation funziona in modalità supervisionata, RBM comprende i dati senza supervisione. Il trucco di Hinton consisteva nel prendere due livelli adiacenti in una rete feedforward e addestrarli come Restricted Boltzmann Machines. La procedura è avviata con l'input e il primo livello nascosto, in modo che sia possibile utilizzare l'input del set di dati per addestrare il modello non supervisionato della Boltzmann Machine.

²⁴ Rumelhart, D. E., McClelland, J. L., 1986a. *On learning the past tenses of English verbs*. In: *Rumelhart and McClelland* (1986b), pp. 216-271.

²⁵ Per una chiara descrizione delle posizioni razionaliste ed empiriste nell'ambito dell'IA vedi Perconti, Pietro & Plebe, Alessio (2020). *Deep Learning and cognitive science*. *Cognition* 203:104365.

Questo modello viene utilizzato per generare un nuovo dataset, semplicemente elaborando tutti gli input.

Questo nuovo dataset viene utilizzato per addestrare gli altri livelli. Questa procedura è una sorta di pre-addestramento che dà una forma iniziale a tutte le connessioni della rete, da perfezionare ulteriormente mediante la normale retropropagazione utilizzando sia gli input che gli output noti del dataset.

La novità – oltre che nell’attenta analisi di una grande mole di informazioni, Big data²⁶ – risiede nella capacità della sequenza di precise istruzioni, a differenza dei primi algoritmi²⁷, di apprendere dai dati stessi adattandosi alle nuove ‘nozioni’ accolte e reinterpretate, apprendendo dai propri errori. L’imitazione del cervello umano è il principio fondamentale degli algoritmi di IA ma, come si scoprirà in seguito con il *Deep Learning*, un algoritmo ingegnoso può dare il massimo dei benefici anche al di là dell’imitazione della natura umana, sfruttando sapientemente le reti neurali artificiali.

L’apprendimento automatico, attraverso una serie di calcoli matematici che vengono eseguiti con hardware evoluti, prevede tre modalità diverse²⁸: l’apprendimento supervisionato (*supervised learning*) in cui l’algoritmo apprende restituendo output a ogni input tramite esempi forniti dal programmatore. Dall’insieme dei dati mostrati, la macchina risponde allo stimolo un determinato

²⁶ L’espressione *Big Data* riassume l’enorme quantità di informazioni (Volume), l’analisi tempestiva (Velocità) e le differenze trasversali (Varietà) dei dati raccolti. Le tre V dei Big data a cui sono state aggiunte la Veridicità e il Valore fino ad arrivare a 42 caratteristiche per considerare i dati come Big Data. Cfr. Shafer, T. *The 42 V’s of Big Data and Data Science*. Available online: <https://www.elderresearch.com/blog/42-v-of-big-data>; Martinez-Mosquera, D.; Navarrete, R.; Lujan-Mora, S. *Modeling and Management Big Data in Databases — A Systematic Literature Review*. *Sustainability* 2020, 12, 634; AM Paraschiv, M Danubianu, *BIG DATA - Present Opportunities and Challenges*. *Brain. Broad Research in Artificial Intelligence & Neuroscience*. 2019 Special Issue, Vol. 10, p.15-21; Terzi, D.S.; Arslan, B.; Sagioglu, S. Smart Grid Security Evaluation with a Big Data Use Case. In *Proceedings of the 2018 IEEE 12th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG 2018)*.

²⁷ Per un ragionevole tributo a Ada Lovelace per il primo algoritmo nella storia dell’informatica cfr. Patricia Fara (2020) *Ada Lovelace: the making of a computer scientist*, *Women's History Review*, 29:1, 168-170.

²⁸ Love, B.C. *Comparing supervised and unsupervised category learning*. *Psychonomic Bulletin & Review* 9, 829–835 (2002).

input, scegliendo l'output corretto per il raggiungimento dell'obiettivo stabilito; non supervisionato (*unsupervised learning*) in cui l'algoritmo stesso trova una logica di classificazione e associazione. La macchina è in grado di scegliere la soluzione migliore senza prima avere visto le varie possibilità di output, imparando dai propri errori senza far ricorso a una logica matematica impostata; con rinforzo (*reinforced learning*), in cui l'algoritmo apprende e cerca soluzioni adatte ai propri errori.

È con il terzo tipo di apprendimento che si lascia libertà all'algoritmo di imparare dai propri errori rinforzandosi. Stesso principio utilizzato, per sconfiggere il campione del mondo di *Go*, dal potente software della DeepMind fondata da Demis Hassabis²⁹.

Se negli anni cinquanta con il *Perceptron* di Frank Rosenblatt³⁰, il più semplice modello di rete neurali, si tentava di organizzare e memorizzare le informazioni nel cervello analogamente ai neuroni, le nuove reti neurali emulano il funzionamento del cervello umano attraverso più livelli tra l'input e l'output interconnessi da nodi "sinaptici". Il termine *rete* appunto designa una composizione incrociata di più funzioni dove la prima rappresenta il primo *layer* e l'ultima il layer di *output*. Più livelli interni, quindi nascosti, ci sono nella rete più la rete neurale è "profonda".

Ogni output di un layer profondo è utilizzato come input per il layer successivo. In ogni neurone viene eseguita un'operazione semplice utilizzando come dati di input, gli output provenienti dai neuroni del layer precedente.

La topologia (il numero e tipo dei neuroni, i collegamenti della rete *feed forward*) è organizzata infatti a livelli: ogni neurone di un livello riceve input solo dai neuroni del livello precedente e propaga gli output solamente verso i neuroni dei livelli

²⁹ Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., et al. (2016). *Mastering the game of Go with deep neural networks and tree search*. *Nature*, 529, 484–489.

³⁰ Rosenblatt, F. (1958). *The perceptron: A probabilistic model for information storage and organization in the brain*. *Psychological Review*, 65(6), 386 -408. <https://doi.org/10.1037/h0042519>; cfr. Rosenblatt, F. (1962). *Principles of neurodynamics: Perceptron and the theory of brain mechanisms*. Washington (DC): Spartan.

successivi, interconnessi da livelli nascosti. La feed-forward neural network è una rete in cui tutte le connessioni vanno da un vertice ad un altro di numero maggiore. In pratica, i vertici sono strutturati in strati, con connessioni solamente verso strati più alti. Ogni neurone ha quindi la funzione di propagare il segnale attraverso la rete che risponde sollecitata dagli stessi input. La presenza di più strati nascosti tra il livello di input e il livello di output è ciò che differenzia il *Deep Learning* e la generazione intermedia di reti neurali, di cui si parlerà più avanti.

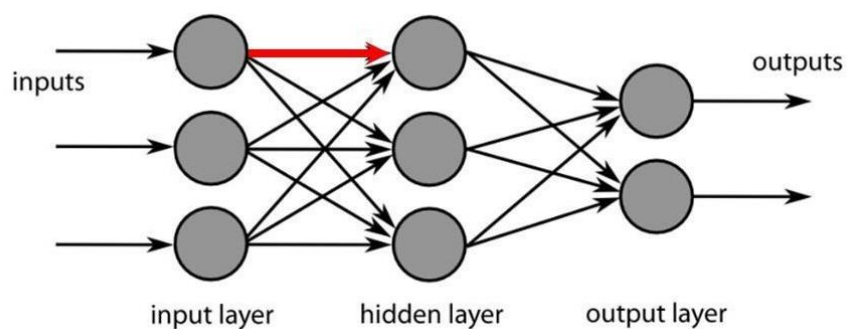


Figure1.2 Rete Feedforward

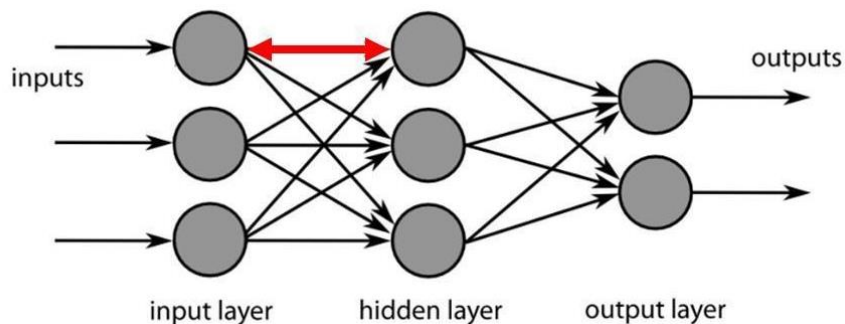


Figure 1.3 Rete Recurrent

Il cervello umano elabora le informazioni appunto in modo parallelo e le distribuisce in tutti i vari nodi neurali non necessariamente a un motore centrale. Per questo motivo reti neurali del tipo PDP rappresentano i precursori empirici del *Deep Learning*. Se da un lato il gruppo di scienziati coinvolti nel PDP ha forgiato un'alleanza ancora più forte con le scienze cognitive rispetto a Rosenblatt, dall'altro la connessione con le neuroscienze è andata sempre più a scemare. Anche se

inizialmente la maggior parte dei protagonisti del Gruppo PDP erano appassionati estimatori delle neuroscienze³¹. Ciò che più ha avvicinato le nuove reti neurali artificiali alle neuroscienze sono stati gli studi che hanno comparato le reti neurali artificiali al visual system³² e che hanno suscitato diverse reazioni³³. Alla luce di una ricognizione dell'inattingibile trasparenza degli algoritmi del *Deep Learning*, che hanno visto una rapida e inaspettata ascesa dovuta sia al controllo pieno dei dati analizzati con precisione sia alla succitata connessione con lo sviluppo delle *neuroscienze*,³⁴ la capacità predittiva dell'algoritmo sarà fonte del ragguglio energetico per questo tipo di manutenzione di impianti di energia rinnovabile di cui discuteremo nei successivi capitoli.

1.2 Il Deep Learning e il suo successo applicativo

Il *Deep Learning* è comunemente considerata una tecnica di machine learning³⁵, ma si distanzia ampiamente da questo ambito nel suo insieme, che ha una

³¹ Rogers, T. T. and McClelland, J. L. (2014). Parallel distributed processing at 25: Further explorations in the microstructure of cognition. *Cognitive Science*, 38: 1024–1077.

³² Clarke, A., Devereux, B. J., Randall, B. and Tyler, L. K. (2015). Predicting the time course of individual objects with MEG. *Cerebral Cortex*, 25: 3602–3612. Cichy, R. M., Khosla, A., Pantazis, D., Torralba, A. and Oliva, A. (2016). Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. *Scientific Reports*, 6: 27755.

³³ Gauthier, I. and Tarr, M. J. (2016). Visual object recognition: Do we (finally) know more now than we did? *Annual Review of Vision Science*, 2: 16.1–16.20. Olshausen, B. A. (2014). Perception as an inference problem. In Gazzaniga, M. S., editor, *The Cognitive Neurosciences*, pages 295–304. MIT Press, Cambridge (MA). Fifth edition.

³⁴ Per una disamina sull'interconnessione tra *deep neural network* e *neuroscienze* vedi Plebe, A., & Grasso, *The Unbearable Shallow Understanding of Deep Learning*, In *Minds and Machine* 29:515-553; Plebe, A., & Grasso, G. (2016). *The brain in silicon: History, and skepticism*. In F. Gadducci & M. Tavosanis (Eds.), *History and philosophy of computing* (pp. 273–286). Berlin: Springer.

³⁵ Per una rassegna sui trent'anni di storia del Machine Learning cfr. J. Wang, C. Jiang, H. Zhang, Y. Ren, K. Chen and L. Hanzo, *Thirty Years of Machine Learning: The Road to Pareto-Optimal Wireless Networks*, in *IEEE Communications Surveys & Tutorials*, doi: 10.1109/COMST.2020.2965856.

tradizione distinta dall'IA, interna alla statistica, e nella sua intersezione con l'IA maggiormente legata al filone razionalista citato precedentemente.

Caratterizzata appunto da “profondi” e nascosti livelli di reti neurali, la metodologia del *Deep Learning* restituisce piena dignità all’apprendimento empirico rendendo necessaria una rielaborazione della storia dell’ IA e delle reti neurali. Tutto quello che arriva in input viene memorizzato nei ‘pesi’ di ogni strato che parametrizzano i dati trasformati al fine di creare, grazie a questo tipo di apprendimento, un set di valori per i pesi totali di tutti gli strati della rete. Una rete neurale *deep* può contenere decine di milioni di parametri. I parametri principali di una rete sono precisamente i suoi pesi, ve ne sono anche altri di minor importanza su cui non ci si sofferma.

La funzione di misurare e controllare i dati output è svolta dalla *loss function* della rete³⁶ ovvero una funzione che misura l’errore di approssimazione. Per riassumere una rete *deep* è composta da: dati in input, *loss function* (che osserva i cambiamenti dovuti all’apprendimento e misurando quanto i risultati siano congrui con l’obiettivo del modello di rete) e *Optimizer* (che determina come prosegue il processo di apprendimento grazie ai valori offerti dalla *loss function*). In particolare la *loss function* è una per ogni output per poi servirsi della media di tutte le *loss* e ricavarne un solo valore. La *loss function* durante l’addestramento della rete deve tendere sempre al minimo per dimostrare il successo del modello.

³⁶ Chollet F., *Deep Learning with python*, Manning publications Co, Shelter Island 2018, pp. 10-11.

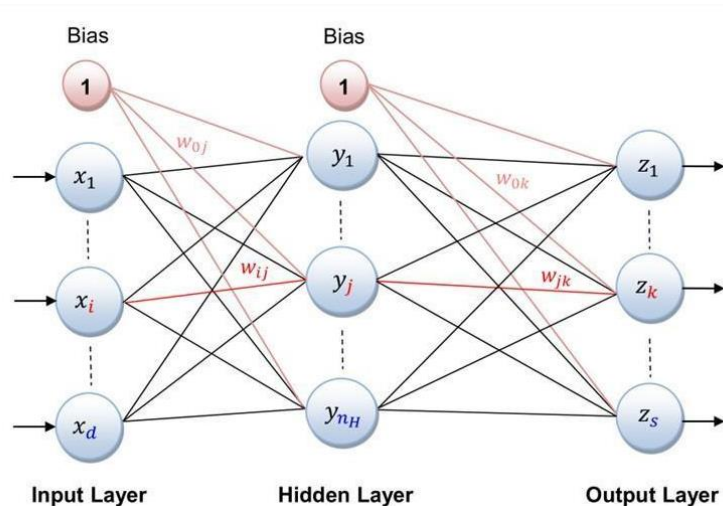


Figure 1.4 Rete con strato nascosto (multistrato)

Con i modelli *deep* è l'algoritmo stesso a scegliere e definire i dati da utilizzare e da associare³⁷. Dal 2012, quando Hinton³⁸ e colleghi convalidano la nuova tipologia di apprendimento automatico profondo³⁹, si apre una nuova possibilità di previsione e calcolo di una mole di dati inanalizzabili dagli umani⁴⁰ ma esemplificati e riprodotti da una "deep mind".

Come descritto sopra, Hinton riesce a inserire due strati adiacenti in una rete feedforward e addestrarli come la *Restricted Boltzmann Machines* (RBM)⁴¹. Mentre la backpropagation risponde a un apprendimento supervisionato⁴², la RBM apprende

³⁷ Schmidhuber, J. (2015). *Deep Learning in neural networks: An overview*. *Neural Networks*, 61, 85–117.

³⁸ Ilya Sutskever, James Martens, George Dahl, Geoffrey Hinton ; *Proceedings of the 30th International Conference on Machine Learning*, PMLR 28(3):1139-1147, 2013.

³⁹ Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). *ImageNet classification with deep convolutional neural networks*. In *Advances in neural information processing systems* (pp. 1090–1098).

⁴⁰ Cfr. Vespignani, A., *L'algoritmo e l'oracolo*, il saggiatore, Milano 2019, pp. 71 e sg.

⁴¹ Hinton cerca un modello alternativo alla backpropagation, e insieme a Sejnowski lo chiameranno Restricted Boltzmann Machine (RBM).

⁴² Partendo dal layer di input calcola il risultato, passando l'informazione avanti lungo la rete, la funzione di attivazione dei nodi di ciascun layer, le derivate della loss function rispetto all'output dell'ultimo layer utilizzando il metodo *Gradient Descent*.

senza supervisione. Il procedimento inizia con un livello di input e un primo strato nascosto utilizzando gli inputs del dataset per l'addestramento non supervisionato della BM e in seguito questo nuovo dataset viene utilizzato per l'addestramento dei successivi strati. Un esempio di modello è l' "autoencoder".

L'autoencoder è un modello neurale insolito, in quanto il suo output non è altro che l'input stesso, ma nel tentare di ricostruire il suo input il modello ne costruisce una rappresentazione compatta. Organizzato in *encoder* che riceve input e riduce il numero di neuroni da portare avanti nel flusso di dati e da un *decoder* che viceversa prende come input l'ultimo strato dell'encoder e lo passa a strati di neuroni via via più numerosi. La parte dell'autoencoder più preziosa è il *code layer* lo strato più interno.

Negli ultimi anni gli algoritmi di *Deep Learning* sono risultati utili in diversi campi della letteratura, in bioinformatica⁴³, medicina⁴⁴, visione computerizzata⁴⁵, guida automatica⁴⁶, robotica⁴⁷, genomica⁴⁸, finanza⁴⁹, e in altri ambiti applicativi della produzione industriale, in particolare nel settore energetico che vedremo più avanti⁵⁰.

⁴³ Min S, Lee B, Yoon S. *Deep Learning in bioinformatics*. Brief, Bioinform. 2017;18(5):851-869..

⁴⁴ Litjens G, Kooi T, Bejnordi BE, et al. *A survey on Deep Learning in medical image analysis*. Med Image Anal J. 2017;42:60-88. <https://doi.org/10.1016/j.media.2017.07.005>.

⁴⁵ Krizhevsky A, Sutskever I, Hinton GE. *ImageNet classification with deep convolutional neural networks*. Adv Neural Inf Process Syst. 2012;1:1-9.

⁴⁶ NVIDIA. *Deep Learning for self-driving cars*. <https://www.nvidia.com/en-us/deep-learning-ai/industries/automotive/>. Accessed May 10, 2018.

⁴⁷ Gashler MS. *Deep Learning in robotics: a review of recent research*. arXiv:1707.07217. 1-41.

⁴⁸ Yue T, Wang H., *Deep Learning for genomics: a concise overview*. Handbook of Deep Learning Applications; 2018. <https://doi.org/10.1021/acs.molpharmaceut.5b00982>.

⁴⁹ H. Buehler, L. Gonon, J. Teichmann e B. Wood. "Deep hedging". In: Quantitative Finance 19.8 (2019), pp. 1271-1291. Christian Bayer e Benjamin Stemper. Deep calibration of rough stochastic volatility models. 2018. arXiv: 1810.03399 [q-fin.PR. Henry Stone. Calibrating rough volatility models: a convolutional neural network approach. 2019. arXiv: 1812.05315 [q-fin.CP].

⁵⁰ Nguyen VN, Jenssen R, Roverso D., *Electrical power and energy systems automatic autonomous vision-based power line inspection: a review of current status and the potential role of Deep Learning*. Electr Power Energy Syst. 2018;99:107-120. <https://doi.org/10.1016/j.ijepes.2017.12.016>.

Helbing G, Ritter M., *Deep Learning for fault detection in wind turbines*. Renew Sustain Energy Rev. 2018;98:189-198. <https://doi.org/10.1016/j.rser.2018.09.012>.

Il progressivo sviluppo del DL ha contribuito al miglioramento delle prestazioni nei campi della visione artificiale e linguaggio artificiale, avvicinandosi progressivamente alle prestazioni dell'uomo. In particolare sembra ci siano diversi elementi in comune tra i modelli di reti neurali *deep* e la struttura cerebrale della vista umana. Si pensi all'architettura neurale specializzata per la visione e proposta inizialmente da Fukushima⁵¹ nel 1979 per il riconoscimento dei caratteri scritti a mano giapponese, *Neocognitron* che è stata trasformata da Hinton insieme a Krizhevsky⁵² nel 2012 in un misto di reti neurali convoluzionali *Neocognitron* e *backpropagation*.

Le reti neurali artificiali, rinnovate con le architetture *deep*, hanno presto trovato applicazione anche nel campo del linguaggio naturale, ma senza i rapidi e vistosi successi dimostrati nella visione. Vi sono diversi motivi per cui il linguaggio si presta in modo più immediato ad essere trattato con strumenti di IA simbolici piuttosto che con le reti neurali. Anzitutto il linguaggio è realizzato con parole, che sono di loro natura simboliche, mentre le reti neurali non possono che processare valori numerici. I modi con cui si transitava dalle parole a vettori numerici ideati durante il periodo del PDP project avevano permesso di esplorare fin da subito ambiti linguistici, ma si trattava di stratagemmi poco efficienti. Una soluzione definitiva a questo problema è stata fornita da Mikolov, con l'introduzione della tecnica di *word embedding*, che trasforma le parole in vettori di attività vettoriale⁵³. Ma questo è solo uno degli aspetti problematici parlando di linguaggio. Quello sicuramente più arduo è il dipanarsi del linguaggio nel tempo, una parola dopo l'altra. Ogni parola mantiene delle relazioni con quelle che l'hanno preceduta, e con quelle a venire. Abbiamo visto precedentemente l'introduzione delle reti ricorsive di Elman nel panorama del PDP

⁵¹ Fukushima, K. (1980). *Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position*. *Biological Cybernetics*, 36:193–202.

⁵² Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). *ImageNet classification with deep convolutional neural networks*. In *Advances in Neural Information Processing Systems*, pages 1090–1098.

⁵³ Tomas Mikolov et al. 2013 *Distributed representations of words and phrases and their compositionality*, NIPS, p. 3111–3119.

project, ed erano una primissima e basilare maniera per dotare le reti neurali artificiali di una forma di memoria. Ma la loro abilità nel catturare la dipendenza delle parole si limitava a forme sintattiche semplici, composte tipicamente da soggetto, verbo e uno o due complementi. Nel linguaggio reale non sono insolite frasi con decine di parole, composte in varie strutture sintattiche concatenate.

Un avanzamento significativo è avvenuto già prima dell'avvento del deep learning, con l'invenzione della struttura denominata LSTM (Long Short-Term Memory)⁵⁴. Questa rete deve il suo acronimo al tentativo di avvicinare la capacità della cognizione umana di tenere presente elementi anche lontani nel tempo, perché cruciali al contesto presente, ma dimenticare informazioni anche molto recenti, se di poco rilievo. Da notare come questa prerogativa sia essenziale per comprendere il linguaggio, ma costituisca anche una strategia cognitiva ben più generale. La memoria è una risorsa preziosa ma limitata, e la sua gestione è un continuo selezionare cosa buttare via, e ogni qual volta vi siano esperienze da conservare, lo diventano sempre in relazione con altre tracce di memoria. La LSTM tenta di imitare questo comportamento con una cella composta da un neurone ricorsivo, in grado di mantenere il proprio contenuto numerico e aggiornarlo nel tempo, con in aggiunta tre gate, che possono essere aperti o chiusi, o anche parzialmente aperti. Con riferimento alla figura 1.5, l'input gate è quello che si può aprire per consentire al neurone di venire aggiornato con il segnale di ingresso attuale. Il forget gate ha l'effetto di annullare il precedente contenuto del neurone, che riflette i segnali precedenti, infine l'output gate controlla il mettere a disposizione o meno il suo attuale contenuto da parte del neurone ad altri che sono collegati oltre la cella di memoria. I tre gate funzionano alla stessa stregua dei neuroni, nel senso che apprendono quando aprirli e chiudersi in base al contesto dei segnali, che possono

⁵⁴ Gers FA, Schraudolph NN, Schmidhuber J (2002) Learning precise timing with LSTM recurrent networks. J Mach Learn Res 3:115– 143.

essere sia gli stessi del neurone che controllano, che quelli di tutti gli altri neuroni che concorrono ad uno strato LSTM.

I primi successi del deep learning nell'ambito del linguaggio furono raggiunti in forza dell'uso combinato del word embedding e di LSTM, organizzate su molti strati, e lo portarono ad essere presto competitivo rispetto ai sistemi simbolici, in ambiti che spaziavano dalla traduzione da una lingua all'altra ai risponditori automatici. Nonostante il rapido progresso, non vi era modo di avvicinare le capacità linguistiche di un essere umano. Per questo fu necessaria una ulteriore e ancor più radicale rivoluzione, che arriva nel 2017, con l'invenzione dell'architettura chiamata Transformer⁵⁵. Si tratta di un metodo più sofisticato di LSTM per stabilire relazioni tra parole all'interno di una frase, o anche diverse frasi, derivate interamente dall'esperienza su consistenti corpora di linguaggio scritto.

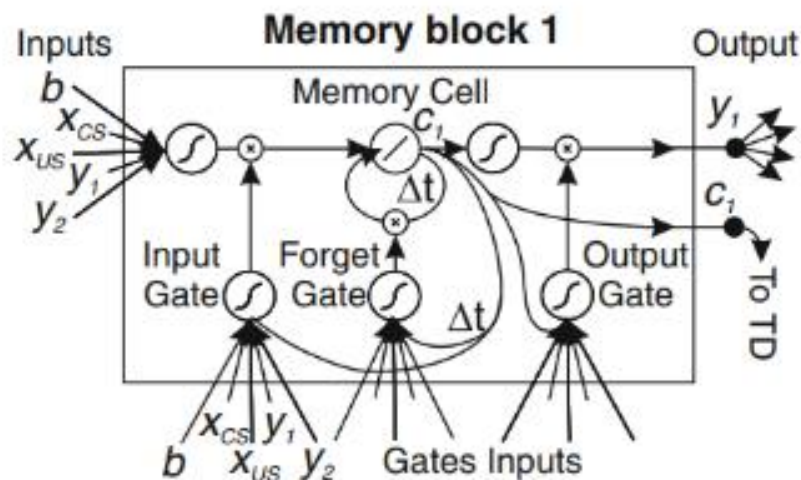


Figura 1.5

Nel giro di pochi anni il deep learning, sottoforma di modelli Trasformer, ha fatto del processamento del linguaggio naturale il suo ambito di maggior successo, fino a

⁵⁵ ⁵⁵ Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems, pages 6000–6010.

centrare in pieno l'agognato obiettivo delineato nel test di Turing. Oggi milioni di utenti nel mondo dialogano con sistemi del tipo chatGPT, non mettendo in dubbio la loro reale intelligenza, anzi, sfruttandola per ottenere testi scritti di alta qualità per svariate applicazioni. Di per sé l'elaborazione del linguaggio naturale ha poca rilevanza con il tema qui centrale della predizione di esigenze manutentive.

Tuttavia, merita di essere menzionato per due motivi. Anzitutto il successo del deep learning in una sfera talmente complessa e ritenuta irraggiungibile come il linguaggio umano, è un ulteriore riscontro della sua efficacia in generale. Inoltre, la natura sequenziale del linguaggio, e l'estrema rilevanza del saper mettere insieme relazioni tra elementi nella sequenza in finestre molto variabili, si ritrova anche nell'analisi di serie temporali, come quelle prese in considerazione in diverse delle problematiche di manutenzione che verranno illustrate nei prossimi capitoli.

Per perseguire la via cognitiva ai problemi di nostro interesse, risulta certamente più proficuo adottare l'LSTM, nonostante i successi del Transformer. Anche se vi sono in atto tentativi di adottarlo per elaborazioni di immagini, di fatto è un sistema ideato per il linguaggio, e legato a diverse sue specifiche peculiarità. Come visto sopra, l'LSTM implementa una forma di memorizzazione ispirata cognitivamente, che risponde bene all'esigenza generale di selezionare cosa memorizzare, e in relazione a chi, e cosa dimenticare. Indipendentemente se si tratti di parole ascoltate, o impressioni sensoriali. Nei casi che vedremo si tratta proprio di queste ultime, ma prodotte da sensori non biologici di caratteristiche fisiche degli impianti di interesse.

1.3 Come sviluppare modelli Deep Learning

Al fine di sviluppare un prototipo di predizione neurale bisogna anzitutto effettuare delle scelte di campo riguardo gli strumenti opportuni, esistono oggi diverse piattaforme create specificatamente per facilitare lo sviluppo di modelli Deep

Learning: Tensorflow, Theano, Pytorch, Caffe. Tra queste e altre possibili piattaforme la scelta è ricaduta su Tensorflow, prodotto e mantenuto da Google. La motivazione principale è la sua ampia diffusione nel mondo scientifico ed ingegneristico, che ne garantisce affidabilità e una completa documentazione. Nel grafico (figura 1.5) sono riportate le tendenze di utilizzo delle principali piattaforme, in cui emerge chiaramente il primato di Tensorflow. Un discorso a parte merita Keras, nato inizialmente come insieme di librerie ad alto livello, che consentono di slegarsi dai dettagli tecnici minuti delle piattaforme, era concepito per interfacciarsi verso il basso con diverse piattaforme, tra cui Tensorflow, Theano o Cognitive Toolkit. Theano è una libreria di Python utilizzata per calcoli veloci; Cognitive Toolkit è una piattaforma sviluppata da Microsoft; TensorFlow è la più famosa libreria di matematica simbolica utilizzata per la creazione di reti neurali e modelli di deep learning. Negli anni il rapporto con Tensorflow è diventato privilegiato, ed oggi Keras è di fatto integrato in Tensorflow. Pertanto nel progetto si farà uso soprattutto di funzioni di Keras, come linguaggio sintetico ed efficiente per costruire componenti Tensorflow⁵⁶.

TensorFlow è una piattaforma open source che dispone di strumenti, librerie e risorse che consentono ai ricercatori di creare e distribuire facilmente programmi e applicazioni DL.

⁵⁶Cfr. Chollet, *Deep Learning with Python*, p. 62.

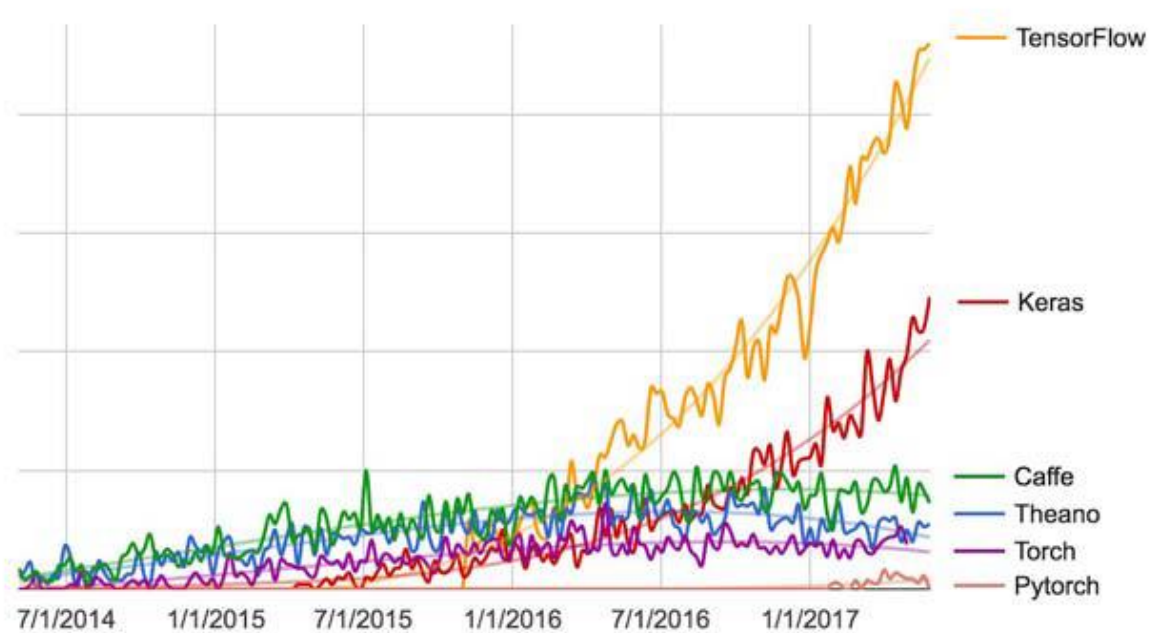


Figure 1.6 Ricerche Google per framework di Deep Learning nel tempo

Il primo passo per sviluppare un modello Deep è creare un *dataset* che comprende due o tre generatori, uno per il training set, uno per il validation set e per addestramenti più complessi anche per il test set.

Inizialmente si addestra il modello con i dati nel training set, poi si validano (separatamente) e infine si testano.

Per elaborare i dati grezzi e renderli oggetto di rete neurale bisogna trasformarli tutti in tensori, un processo che viene chiamato vettorizzazione. I dati devono inoltre essere normalizzati, per esempio, i valori tendenzialmente dovrebbero mantenersi in un range di 0-1.

Dopo aver generato i dati bisogna creare il modello neurale che, in modo non dissimile da quanto raccontato in 1.1 riguardo le reti neurali storiche, si centra sul concetto di layer, o strato. In Keras i modelli si costruiscono organizzandoli in layer, di diverse possibili tipologie. Quello classico, che costituisce l'ossatura delle reti deep, è il

cosiddetto "dense layer" tipico dei livelli hidden, in cui ogni neurone dello strato precedente trova una connessione pesata con il neurone nello strato "dense". Il layer di tipo softmax è invece tipicamente impiegato nell'ultimo strato, e permette di reinterpretare i valori dei singoli neuroni dello strato finale, in termini di probabilità che il dato in esame appartenga ad una tra varie possibili categorie. Siccome il genere di dato degli impianti energetici è tipicamente un insieme di segnali nel tempo, i layer saranno prevalentemente ad una dimensione. Nel prossimo capitolo descriveremo come questi modelli possono essere applicati nella manutenzione predittiva di impianti energetici.

Capitolo 2

Energia eolica e manutenzione

Il proposito di questo capitolo è quello di presentare uno dei campi di energia rinnovabile, quello eolico, in cui oltre al monitoraggio è di fondamentale importanza predire l'insorgere di un guasto al fine di diminuire i costi di manutenzione e aumentare la produzione di energia. In 2.1 si descriverà la manutenzione predittiva che rappresenta un tipo di manutenzione preventiva che aiuta a predire il tempo residuo prima del guasto di un impianto. Il 2.2 porrà in risalto la possibilità di applicare le metodologie del *Deep Learning* alla manutenzione predittiva, sostituendosi a quelle basate sulla conoscenza delle cause dei guasti degli impianti addetti alle energie rinnovabili. Attraverso il *Deep Learning* diventa così possibile sganciarsi definitivamente dal sapere specifico dei fenomeni (in tal caso, ingegneristici), affrontando la manutenzione anche senza competenze specialistiche. Verrà infine descritto il prototipo di predizione neurale dei dati simulati e il modello matematico di riferimento.

La combinazione tra IA e monitoraggio nei settori dell'industria costituisce una frontiera nel campo della manutenzione predittiva, resa possibile solo grazie alla possibilità di raccogliere una grande quantità di informazioni istantanea sui processi produttivi.

2.1 Manutenzione predittiva

Sull'energie rinnovabili e l'ineguagliabile risorsa che rappresentano per molti paesi del mondo⁵⁷, diverse ricerche hanno evidenziato il potenziale (in particolare quello delle

⁵⁷ O. Alnatheer, *The potential contribution of renewable energy to electricity supply in Saudi Arabia*, Energy Policy, 33 (18) (2005), pp. 2298-2312; J.M. Huacuz, *The road to green power in Mexico—reflections on the prospects for the large-scale and sustainable implementation of renewable energy*, Energy

fonti eoliche e solari) di miglioramento dell'efficienza della produzione e sostituzione dei combustibili fossili con fonti inesauribili, e l'impatto futuro sulle reti di distribuzione⁵⁸.

Quando tra il 1887 e il 1888 negli Stati Uniti viene installata e avviata la prima turbina eolica⁵⁹, si rileva che i costi principali da sostenere durante il processo di generazione di energia eolica rimangono quelli relativi al funzionamento e alla manutenzione.

Gli impianti eolici si distinguono in⁶⁰:

- Impianto eolico onshore, installato sulla terra ferma con l'obiettivo di catturare il vento per trasformarlo in energia elettrica. Rappresenta una delle forme più tradizionali di impianto eolico, trovabile nella maggioranza dei paesi europei, negli Stati Uniti, e nel resto del mondo⁶¹.

- Impianto eolico offshore, installato in grandi distese d'acqua, tradizionalmente al largo di una costa marittima. Sono particolarmente presenti in Stati come la Gran

Policy, 33 (16) (2005), pp. 2087- 2099; R. Duke, R. Williams, A. Payne, *Accelerating residential PV expansion: demand analysis for competitive electricity markets*, Energy Policy, 33 (15) (2005), pp. 1912-1929; J.K. Kaldellis, D.S. Vlachou, G. Korbakis, *Techno-economic evaluation of small hydro power plants in Greece: a complete sensitivity analysis*, Energy Policy, 33 (15) (2005), pp. 1969-1985; G.M. Montes, M. del Mar Serrano Lopez, M. del Carmen Rubio Gamez, A.M. Ondina, *An overview of renewable energy in Spain. The small hydro-power case*, Renewable Sustainable Energy Rev, 9 (5) (2005), pp. 521-534; E. Gnansounou, A. Dauriat, C.E. Wyman, *Refining sweet sorghum to ethanol and sugar: economic trade-offs in the context of North China*, Bioresour Technol, 96 (9) (2005), pp. 985-1002.

⁵⁸ Heydar Chamandoust, Ghasem Derakhshan, Seyed Mehdi Hakimi, Salah Bahramara, *Tri-objective scheduling of residential smart electrical distribution grids with optimal joint of responsive loads with renewable energy sources*, Journal of Energy Storage, volume 27, 2020, 101112, ISSN 2352-152X, <https://doi.org/10.1016/j.est.2019.101112>. Yang Li, Zhen Yang, Guoqing Li, Dongbo Zhao, Wei Tian, *Optimal scheduling of an isolated microgrid with battery storage considering load and renewable generation uncertainties*, IEEE Trans. Ind. Electron., 22 (2019), pp. 1565-1575.

⁵⁹ Ingliński, B.; Iglińska, A.; Koziński, G.; Skrzatek, M.; Buczkowski, M. *Wind energy in Poland—History, current state, surveys*, Renewable Energy Sources Act, SWOT analysis. *Renew. Sustain. Energy Rev.* 2016, 64, 19–33.

⁶⁰ IRENA. (2016). Wind Power - Technology Brief. IEA-ETSAP and IRENA.

⁶¹ Burton, T., Jenkins, N., Sharpe, D., & Bossanyi, E. (2011). Wind Energy Handbook. Chichester, United Kingdom: John Wiley & Sons, Ltd.

Bretagna o la Danimarca, e si ergono in acque poco profonde (fino a 100 km dalla costa più vicina), la profondità della pala al fondale raggiunge i 50 metri⁶².

- Impianto eolico offshore galleggiante, prevede l'installazione di impianti offshore in mare aperto, a grande profondità tramite una struttura galleggiante ancorata al fondale tramite dei cavi⁶³.

I costi dei diversi tipi di impianti sono variano significativamente, infatti, se da un lato l'investimento dal peso maggiore di un impianto eolico onshore è rappresentato dalla turbina (60% dei costi), solamente il 36% del costo complessivo è previsto per una pala eolica offshore.

Nel caso dell'energia eolica offshore, per poter collegare l'energia prodotta alla rete di distribuzione elettrica, i costi rappresentano ad oggi mediamente il 33% dell'investimento iniziale complessivo. Come mostra il report annuale di BloombergNEF, un impianto eolico offshore è collegato tramite una rete di cavi sottomarini ad una Offshore station, che a sua volta si collega tramite cavi di esportazione a una Onshore station.⁶⁴

Per il contenimento dei costi, un'adeguata manutenzione per presidiare e contenere le conseguenze di degrado e implementare nuovi piani manutentivi è di primaria importanza. La normativa UNI 10584⁶⁵ definisce il SIM (Sistema Informativo di Manutenzione) come un complesso di norme, procedure e strumenti atti a raccogliere ed elaborare le informazioni necessarie alla gestione delle attività di manutenzione e per il monitoraggio dell'attività degli impianti. Il loro scopo è di migliorare le prestazioni raggiunte dal sistema oggetto delle attività manutentive.

⁶² *Ibidem*.

⁶³ Just Have a Think. (17. 07 2021). Deep Ocean Floating wind power. Tratto da Just Have a Think: <https://www.justhaveathink.com/wp-content/uploads/2021/07/Prog-158- Transcript.docx>.

⁶⁴ Fingersh, L., Hand, M., & Laxson, A. (2006). Wind Turbine Design Cost and Scaling Model. Colorado, USA: National Renewable Energy Laboratory NREL.

⁶⁵ Normativa UNI 10584 Manutenzione – Terminologia, 2003.

Mettono a disposizione diversi tipi di analisi e di supporti nella scelta di efficaci strategie di manutenzione e gestione dei materiali di ricambio.

Nella manutenzione predittiva le macchine vengono monitorate in tempo reale attraverso l'utilizzo di sensori installati e i guasti rilevati vengono trasmesse al software che provvederà ad identificarle, misurarle e quantificarle.

L'ottimizzazione della produzione e la manutenzione predittiva sono i maggiori benefici della correlazione tra IA e industria (prevenzione dei guasti, migliore gestione delle risorse, risparmio di tempo, ottimizzazione delle performance)⁶⁶.

La manutenzione predittiva⁶⁷, infatti, è un tipo di manutenzione preventiva che, attraverso modelli matematici che individuano il tempo residuo prima del guasto critico di un macchinario, monitora le condizioni della produttività. I costi di manutenzione possono rappresentare dal 15 al 60 per cento dei costi di produzione⁶⁸. Sono necessarie competenze tecniche, ingegneristiche, metodologie scientifiche al fine di conservare e ripristinare l'adempimento della produttività di un macchinario.⁶⁹

Si tratta chiaramente di competenze non banali da tenere insieme, e uno dei vantaggi del *Deep Learning* è proprio di non richiederle. Affronteremo questo nella prossima sezione.

⁶⁶ Lars, L.; Elizabeth, T., *The benefits, risks, and future of artificial intelligence in solar and wind energy*. In *Making Renewables Smarter*; Group Technology and Research Position Paper; DNV GL: Oslo, Norway, 2017.

⁶⁷ Y. Xu, Y. Sun, J. Wan, X. Liu, and Z. Song, *Industrial big data for fault diagnosis: Taxonomy, review, and applications*, *IEEE Access*, to be published, doi: 10.1109/ACCESS.2017.2731945. L. Linxia and F. Köttig, *Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction*, *IEEE Trans. Rel.*, vol. 63, no. 1, pp. 191_207, Mar. 2014.

⁶⁸ Cfr. R. Keith Mobley, *An introduction to predictive maintenance*, 2002, Elsevier Science (USA).

⁶⁹ Jardine, A.; Tsang, A. *Maintenance, Replacement, and Reliability—Theory and Applications*, 2nd ed.; Taylor and Francis, Boca: Boca Raton, FL, USA, 2006; ISBN 0-8493-3966-9.

2.2 Reti neurali artificiali per manutenzione di turbine eoliche

La manutenzione predittiva assicura modelli specifici che aumentano il grado di previsione analizzando i dati ricevuti dai sensori, i guasti ripetitivi, cronologie di errori nelle macchine, tutti dati che aiutano a eseguire la previsione della vita utile rimanente⁷⁰. La crescente domanda di energia eolica necessita della riduzione dei costi di manutenzione e quindi di nuove tecniche di monitoraggio delle turbine eoliche⁷¹.

Nel contesto della letteratura complessiva sulla manutenzione predittiva in impianto eolici risalta la marginalità di proposte impieganti deep learning, come sarà discusso più avanti le motivazioni sono probabilmente da cercarsi sia in una divergenza di tradizioni tra ambiti disciplinari, sia nella scarsa disponibilità di dati storici, rispetto ad applicazioni dove il deep learning è dominante. Tra le poche proposte da segnalare ve ne è una, basata sul deep denoising autoencoder (DDA) - una variante dell'autoencoder introdotto, come descritto in 1.2, da Hinton - che propone una sorta di elettrocardiogramma di un macchinario simile a quello utilizzato per l'uomo⁷². Un vantaggio che riduce la necessità di esperti nel settore e quindi diminuisce i costi di manutenzione. L'“elettrocardiogramma” registrerebbe la durata e la qualità di ogni operazione della macchina valutando il tempo residuo di “vita”.⁷³ A differenza di uno strumento come

⁷⁰ Sharifi, V., Abdollahi, A., & Rashidinejad, M. (2022). Flexibility-based generation maintenance scheduling in presence of uncertain wind power plants forecasted by deep learning considering demand response programs portfolio. *International Journal of Electrical Power & Energy Systems*, 141, 108225.

⁷¹ W. Qiao and D. Lu, *A Survey on Wind Turbine Condition Monitoring and Fault Diagnosis—Part I: Components and Subsystems*, in *IEEE Transactions on Industrial Electronics*, vol. 62, no. 10, pp. 6536-6545, Oct. 2015, doi: 10.1109/TIE.2015.2422112.

⁷² H. Yan, J. Wan, C. Zhang, S. Tang, Q. Hua and Z. Wang, *Industrial Big Data Analytics for Prediction of Remaining Useful Life Based on Deep Learning*, in *IEEE Access*, vol. 6, pp. 17190-17197, 2018, doi: 10.1109/ACCESS.2018.2809681.

⁷³ Long Wen, Yan Dong, Liang Gao. *A new ensemble residual convolutional neural network for remaining useful life estimation*. *Mathematical Biosciences and Engineering*, 2019, 16(2): 862-880. doi: 10.3934/mbe.2019040

il *factory information system* (FIS) in grado di riconoscere le criticità dopo che si sono verificati i guasti senza capire quale parte del ciclo di produzione ha causato il problema, il *device electrocardiogram* (DECG) monitora tutte le operazioni all'interno di un ciclo di produzione. In questo caso il *Deep Learning* per predire il tempo residuo di vita (output) utilizza come input i dati sullo stato dei dispositivi guasti, in particolare servendosi di una *convolutional neural network* (CNN)⁷⁴ e una *recurrent neural network* (RNN)⁷⁵.

Una analoga architettura basata su deep denoising autoencoder ha trovato una proposta nell'identificazione dei guasti delle turbine, mirando a classificarle in degradate e non degradate al fine di stimarne le diverse prestazioni e prospettive di funzionamento nel futuro⁷⁶.

Un recente studio ha dimostrato la necessità dell'utilizzo di un algoritmo su un modello di turbina eolica per rilevare i guasti dell'impianto e identificare anche le cause per una migliore prognosi del malfunzionamento dei diversi componenti (aerodinamici, meccanici e elettrici)⁷⁷. Questo modello che utilizza MATLAB è utile per analizzare e simulare diverse e frequenti condizioni della turbina fornendo importanti informazioni.

⁷⁴ Le CNN sono reti neurali costruite per elaborare dati nella forma di array multipli. Un'immagine, ad esempio, possiede la struttura di un'array a tre dimensioni. LeCun, Yann, Léon Bottou, Yoshua Bengio e Patrick Haffner. «Gradient based learning applied to document recognition». In: Proceedings of the IEEE 86.11 (1998), pp. 2278–2324.

⁷⁵ Elman, Jeffrey L. «Finding structure in time». In: Cognitive science 14.2 (1990), pp. 179–211; T. Mikolov, M. T. Kara, L. Burget, J. Cernock, and S. Khudanpur, *Recurrent neural network based language model*, *Interspeech*, vol. 2, p. 3, Sep. 2010.

⁷⁶ S. Sato and K. Sanda, *Degradation Estimation of Turbines in Wind Farm Using Denoising Autoencoder Model*, 2019 IEEE International Conference on Prognostics and Health Management (ICPHM), San Francisco, CA, USA, 2019, pp. 1-6, doi: 10.1109/ICPHM.2019.8819375.

⁷⁷ Ibrahim, Raed; Weinert, Jannis; Watson, Simon (2016): *Neural networks for wind turbine fault detection via current signature analysis*. Loughborough University. Conference contribution. <https://hdl.handle.net/2134/23014>

L'utilizzo di sistemi di reti neurali artificiali per rilevare i guasti del sistema in tempo reale risulta il più tempestivo e il meno dispendioso modello di prevenzione⁷⁸.

Lo studio e lo sviluppo di algoritmi che forniscono la possibilità di predire una grande mole di dati partendo da quella già immagazzinata e prodotta permette e induce i dati a essere facilmente reperibili e necessariamente trasparenti⁷⁹. Da questa breve rassegna si è potuto constatare quanto l'adozione di reti neurali artificiali per la manutenzione degli impianti eolici non sia certo una novità, e vi sia un certo numero di variegato proposte, che rimane tuttavia marginale rispetto a metodi tradizionali, basati sull'effettiva identificazione ingegneristica delle cause di guasti, partendo dalla sistematica modellazione di ogni componente coinvolto. Il deep learning è ancora lontano da costituire un'alternativa praticabile rispetto a questi metodi, questa ricerca tenta di contribuire in tal senso.

2.3 Prototipo di predizione neurale su dati simulati

Il nostro prototipo prevede diverse fasi per l'addestramento della rete neurale. Primariamente si generano i dati su un dataset che sia in qualche modo assimilabile all'effettivo monitoraggio sensoriale di cui di solito sono equipaggiate le turbine eoliche. Da diversi anni si è consolidato in ambito industriale l'uso del concetto sintetizzato dall'acronimo SCADA (Supervisory Control and Data Acquisition), che nello specifico delle turbine eoliche comprende una serie di standard su quali

⁷⁸ M. Nithya, S. Nagarajan and P. Navaseelan, *Fault detection of wind turbine system using neural networks*, 2017 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR), Chennai, 2017, pp. 103-108, doi: 10.1109/TIAR.2017.8273694.

⁷⁹ Per una ricognizione sull'importanza del principio di trasparenza del Machine Learning Cfr. Xiaoli Li and Jun Huan, *Constructivism Learning: A Learning Paradigm for Transparent Predictive Analytics*, 2017; Chiticariu, Laura, Yunyao Li, and Frederick Reiss. *Transparent machine learning for information extraction: state-of-the-art and the future. EMNLP (tutorial)* (2015); Samek, Wojciech, et al. *Toward Interpretable Machine Learning: Transparent Deep Neural Networks and Beyond*, arXiv preprint arXiv:2003.07631 (2020).

parametri monitorare, e con quale cadenza. Naturalmente per la simulazione risulta inutile adottare gli standard SCADA (vedi figure 1 e 2) per il formato dei dati, lo si è preso come riferimento invece per verificare quali dati sia possibile ottenere dalle turbine, e quali siano effettivamente impiegati nella letteratura scientifica sulla manutenzione.

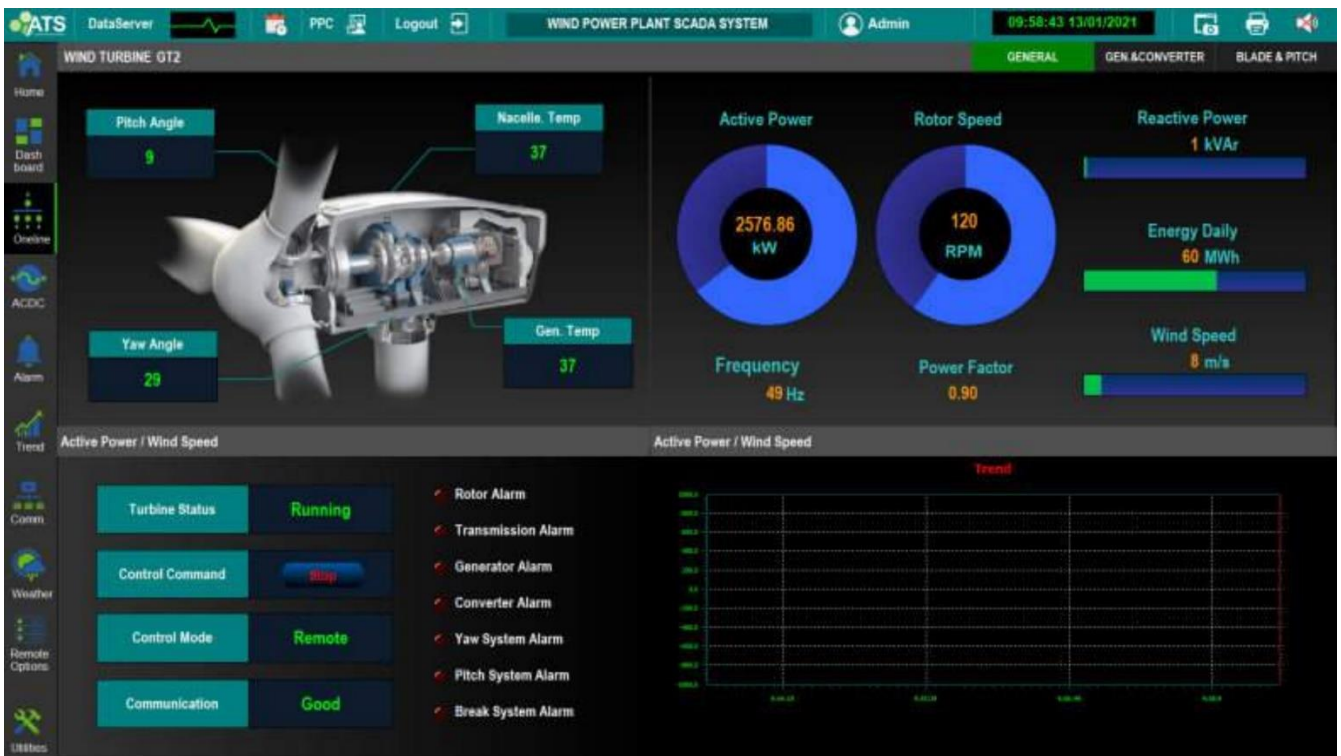


Figura 2.1

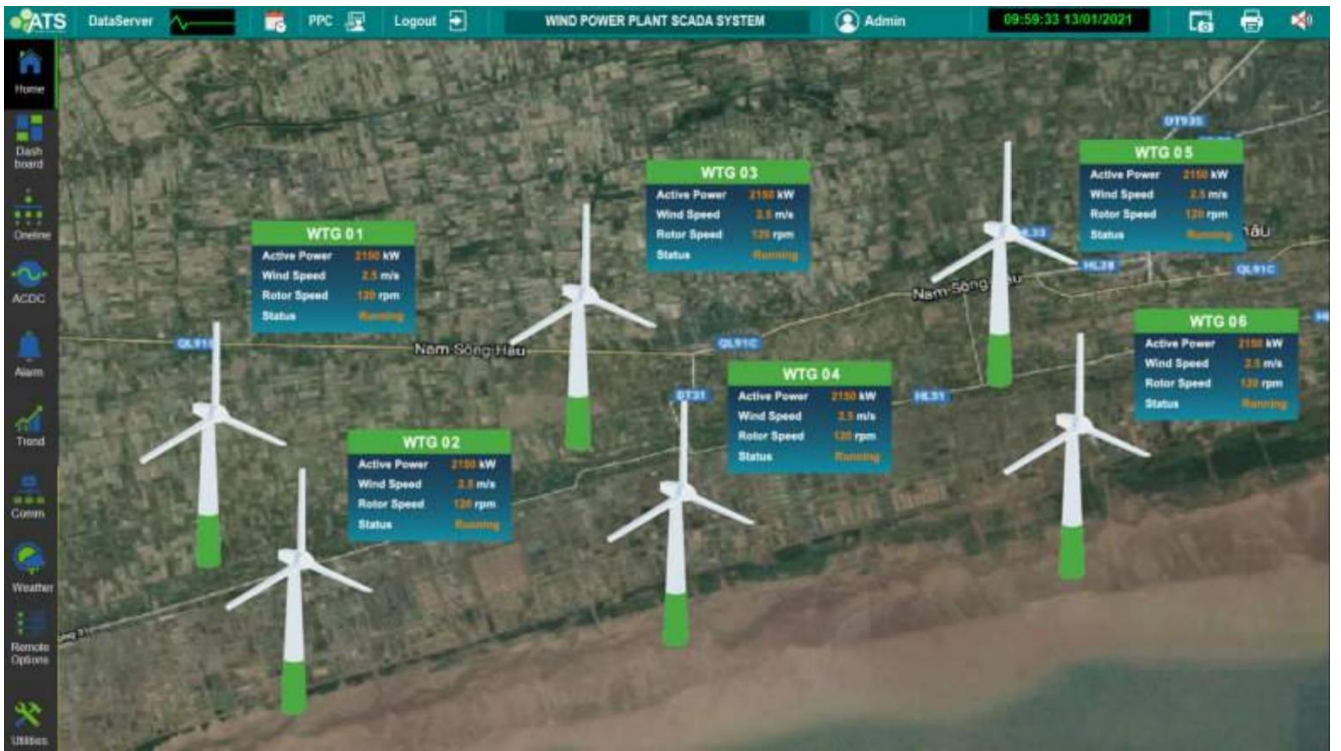


Figura 2.2

Sono stati utilizzati dati di velocità del vento della stazione meteorologica di Fontanarossa di Catania campionati ogni 10 minuti, per tutto l'arco del 2015. Da questi si sono sintetizzate le risposte di potenza delle macchine eoliche ipotetiche, simulando un possibile inizio di rottura mediante variazione dell'inerzia di risposta della macchina.

Tra i dati SCADA che risultano più utilizzati per inferire malfunzionamenti la velocità del vento e la potenza elettrica generata sono quelli più attenzionati.

Abbiamo aggiunto un valore random di *noise* (rumore) per la velocità del vento di ogni campione e potenza elettrica.

Viene considerata sia la velocità del vento per ogni istante sia quella misurata all'intervallo di tempo precedente. Per esempio, in un intervallo tra le ore 8 e le ore 9 considero sia quella prima delle ore 8 sia la media della velocità tra le ore 8 e le ore 9. Questa media è calcolata tra il valore attuale e quello che ho previsto per gli intervalli temporali precedenti.

Se la macchina funzionasse bene e quindi pronta a reagire a ogni velocità non dovrebbe ricordare o ancorarsi ai giorni precedenti, altrimenti andrebbe per inerzia. Grazie ai modelli matematici (in assenza di un cervello umano che a differenza della macchina apprenderebbe implicitamente), agli algoritmi, al linguaggio di programmazione e infine ai framework dei linguaggi, la macchina migliora.

Sia la velocità del vento che produzione elettrica sono presi quindi ogni 10 minuti, quindi 144 campioni al giorno, i vettori (array) hanno dimensione 51264 (144 x 356).

Per concludere, spieghiamo il modello matematico di riferimento alla base del nostro algoritmo e del nostro addestramento.

Ogni campione temporale i -esimo di potenza elettrica e_i viene derivato dalla seguente equazione:

$$e_i = \xi w_i^\alpha + \zeta e_{i-1} + \mathcal{N}$$

$$\begin{aligned} \alpha &: 1.8 \\ \xi &: \begin{cases} 0.6 & \text{funzionante} \\ 0.2 & \text{prossima a guasto} \end{cases} \\ \zeta &: \begin{cases} 0.3 & \text{funzionante} \\ 0.4 & \text{prossima a guasto} \end{cases} \\ \mathcal{N} &: \text{distribuzione uniforme in } [0 \dots 0.1] \end{aligned}$$

dove w_i è il valore della velocità del vento allo stesso istante temporale i -esimo, l'equazione tiene conto ricorsivamente del precedente valore di potenza elettrica e_{i-1} .

L'esponente α modula quanta potenza elettrica deriva da una determinata intensità di vento, e nelle simulazioni qui descritte viene mantenuto uguale per macchina intatta o prossima a guasto. Naturalmente anche questo valore potrebbe essere diversificato per eventuali tipologie di malfunzionamenti. I due parametri ξ e ζ soppesano quanto la potenza elettrica istantanea dipenda dal vento istantaneo (ξ), e dall'inerzia derivante dal vento negli istanti di tempo precedente (ζ). Infine \mathcal{N} è un valore casuale a distribuzione uniforme in un intervallo prefissato.

I parametri per l'addestramento del modello sono:

- le epoche, cioè quanto dura l'addestramento, il numero di iterazioni che devono essere svolte dall' algoritmo.
- la dimensione dei gruppi di vettori addestrati
- la velocità di apprendimento (learning rate)

Per una maggiore trasparenza sul modello addestrato presentiamo qui di seguito alcune righe di codice python che descrivono sia la creazione del modello, sia il successivo addestramento, che infine l'esecuzione del modello addestrato su un set di test, ed elaborazione statistica dei risultati ottenuti:

```

78
79
80 def create_model():
81     """
82     crea il semplice modello neurale
83     notare come ci siano due "primi" strati paralleli, che si prendono carico separatamente
84     del vettore wind e del vettore power, poi convergono verso un unico secondo strato
85
86     return: [keras.engine.training.Model]
87
88     """
89     w = Input( shape=( daily_samples, ) ) # input layer per wind
90     p = Input( shape=( daily_samples, ) ) # input layer per power
91     xw = layers.Dense( cnfg.layer_1, name="dense_w" )( w ) # first hidden layer for wind
92     xp = layers.Dense( cnfg.layer_1, name="dense_p" )( p ) # first hidden layer for electric power
93     x = layers.Concatenate( axis=1 )( [ xw, xp ] ) # concatenate the two first hidden layers
94     x = layers.Dense( cnfg.layer_2, name="dense_2" )( x ) # second hidden layer
95     x = layers.Dense( cnfg.layer_3, name="dense_3" )( x ) # third hidden layer
96     x = layers.Dense( 2, activation='softmax', name="dense_o" )( x ) # output layer
97
98     return Model( inputs=[w,p], outputs=x, name="nn_model" ) # notare le due componenti dell'input

```

```

121
122 def train_model( model, gens ):
123     """
124     addestra il modello
125
126     model: [keras.engine.training.Model]
127     gens: [tuple] generatori per training e validation
128
129     return: [keras.callbacks.History]
130
131     """
132     tgen, vgen = gens # train and valid generators
133     n_train = tgen.n_samples # number of samples
134     n_valid = vgen.n_samples # number of samples
135     train_steps = ceil( n_train / cnfg.batch_size ) # steps for a training epoch
136     valid_steps = ceil( n_valid / cnfg.batch_size ) # steps for a validation
137     optimizer = optimizers.Adam( lr=cnfg.lrate ) # optimizer
138     loss = losses.binary_crossentropy # loss
139
140     # prima dell'addestramento il modello dev'essere "compilato", e' qualcosa di interamente
141     # interno a Keras, che crea automaticamente tutto il necessario per ottimizzare progressivamente
142     # i pesi neurali, in modo da minimizzare la loss
143     model.compile(
144         optimizer = optimizer,
145         loss = loss
146     )
147
148     # una volta compilato, il modello viene addestrato con il comando che prevede l'uso dei generatori
149     history = model.fit_generator(
150         tgen, # il generatore di training
151         validation_data = vgen, # numero di epoche di addestramento
152         steps_per_epoch = train_steps, # il generatore per validation
153         validation_steps = valid_steps, # quanti step occorre eseguire per ogni epoca
154         workers = 0, # lo stesso per la validazione
155         verbose = 1 # faccende interne di Keras su come gestire la GPU
156     ) # visualizza il progresso durante l'addestramento
157
158     return history # un insieme di informazioni riguardo l'addestramento
159

```

```

242
243 def test_model( model, val ):
244     """
245     testa il modello su un dataset, tipicamente quello di validazione
246
247     model:          [keras.engine.training.Model]
248     val:            [data_gen.DataGenerator] generatore per validation
249
250     return:         [dict] matrice di confusione con aggiunta di sensitivity e specificity
251     """
252     n_valid         = val.n_samples          # numero di batch di campioni del validation
253
254     confusion       = {                    # inizializza la matrice di confusione
255         "TP"        : 0,                   # true positive
256         "FP"        : 0,                   # false positive
257         "TN"        : 0,                   # true negative
258         "FN"        : 0,                   # false negative
259     }
260
261     for i in range( n_valid ):             # passa in rassegna tutti i batch nel dataset
262         print( "testing batch {:d}".format( i ) )
263         batch      = val[ i ]
264         b_in, b_targ = batch                # ogni batch ha l'input e il target
265         b_pred     = model.predict_on_batch( b_in ) # calcola la predizione del modello
266
267         # viene mantenuta una matrice di confusione complessiva per il dataset, accumulando quelle di ogni batch
268         conf       = confusion_matrix( b_pred, b_targ )
269         confusion[ 'TP' ] += conf[ 'TP' ]
270         confusion[ 'FP' ] += conf[ 'FP' ]
271         confusion[ 'TN' ] += conf[ 'TN' ]
272         confusion[ 'FN' ] += conf[ 'FN' ]
273
274     # dalla matrice di confusione si ricavano True Positive Rate (sensitivity) e
275     # True Negative Rate (specificity)
276     confusion[ 'TPR' ] = confusion[ 'TP' ] / ( confusion[ 'TP' ] + confusion[ 'FN' ] )
277     confusion[ 'TNR' ] = confusion[ 'TN' ] / ( confusion[ 'TN' ] + confusion[ 'FP' ] )
278
279     return confusion
280

```

Dai grafici potremo facilmente vedere i valori della training loss e della validation loss che dovrebbero presentarsi più vicine possibili. In Figura 2.3 si mostra un addestramento con uno strato di 16 neuroni, un secondo strato di 16 neuroni, il learning rate $1e-04$ e il numero di epoche pari a 20 (n_epochs 20).

In Figura 2.4 gli strati sono entrambi di 8 neuroni e numero di epoche pari a 20 (n_epochs 20); in Figura 2.3 la validation loss tende a migliorare con uno strato di 16 neuroni, un secondo strato di 64, il learning rate $1e-06$ e il numero di epoche pari a 20 (n_epochs 20). In Figura 2.4 la validation loss è quasi identica alla training loss con uno strato di 16 neuroni e un secondo strato di 8, il learning rate di $1e-04$ e il numero di epoche pari a 100 (n_epochs 100).

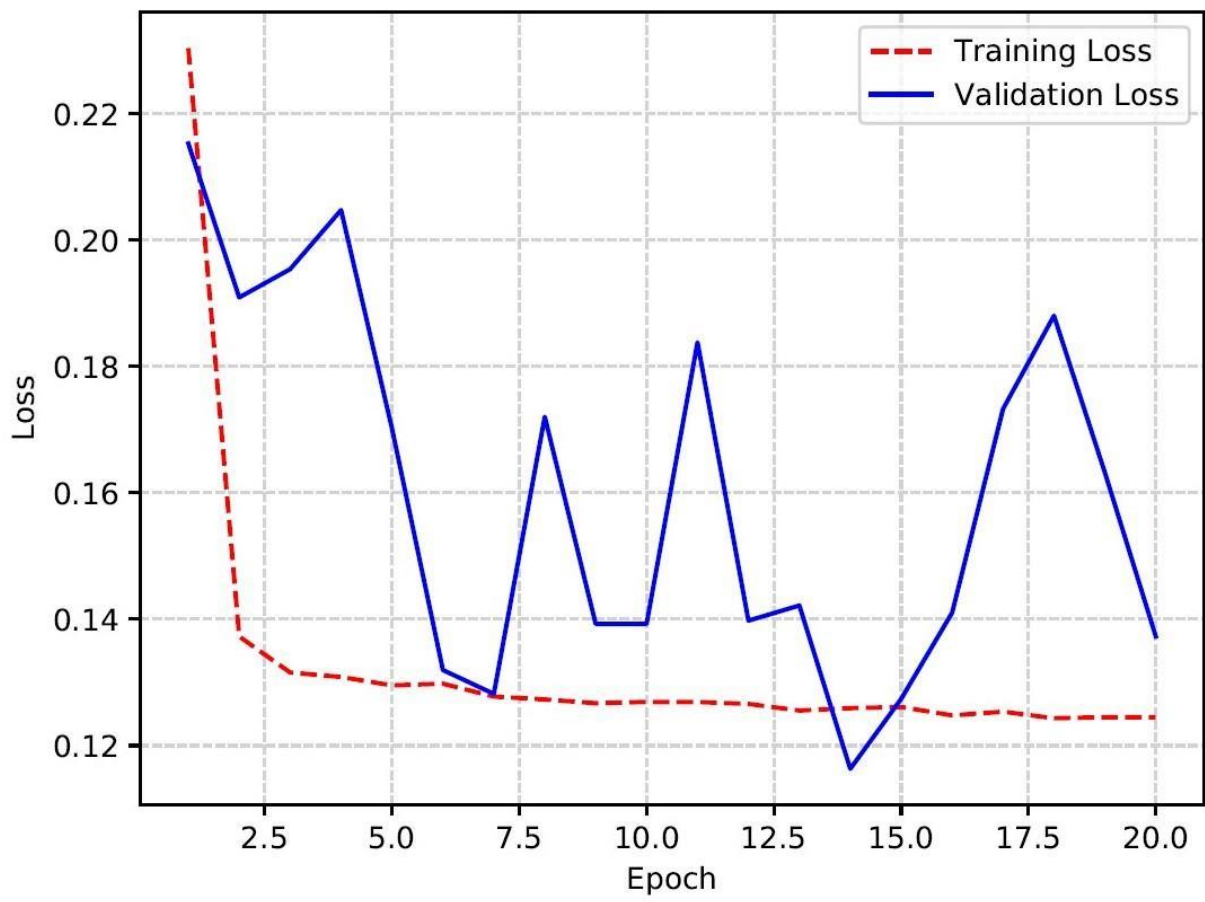


Figura 2.3

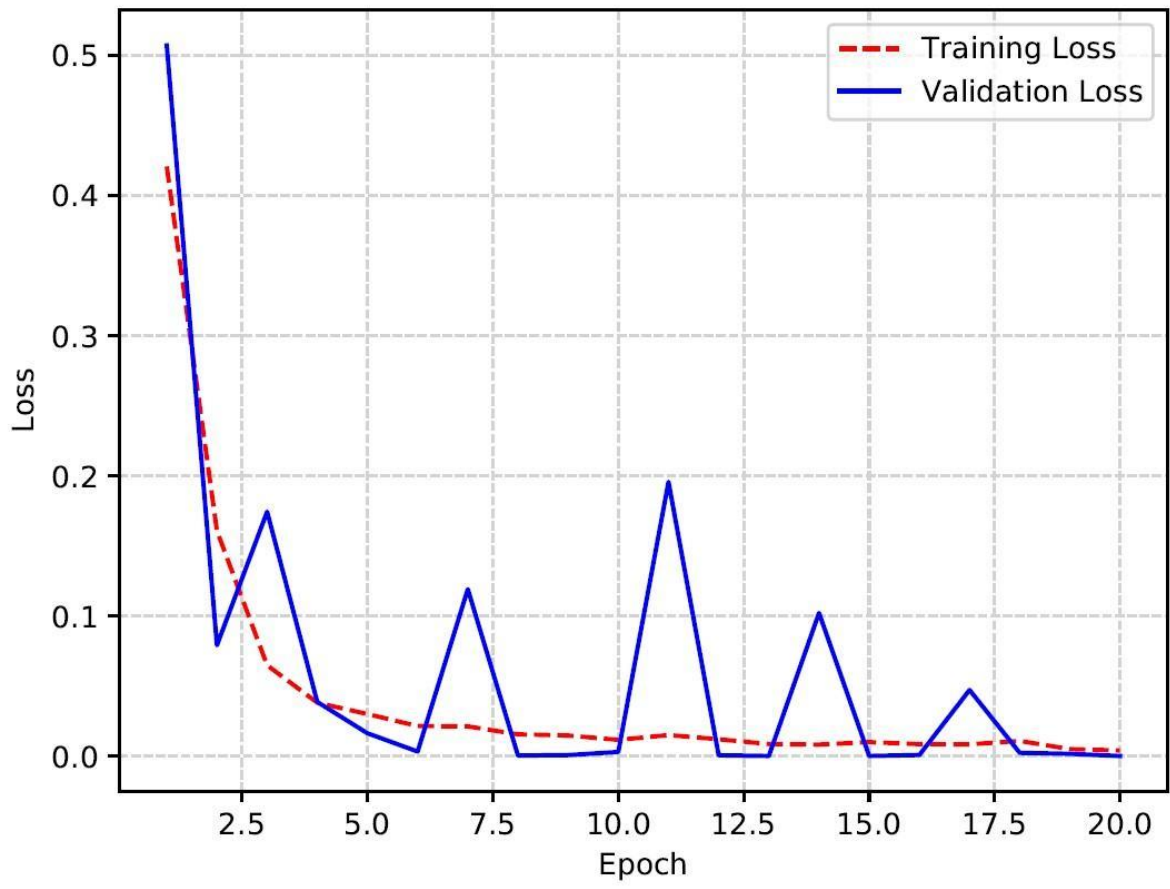


Figura 2.4

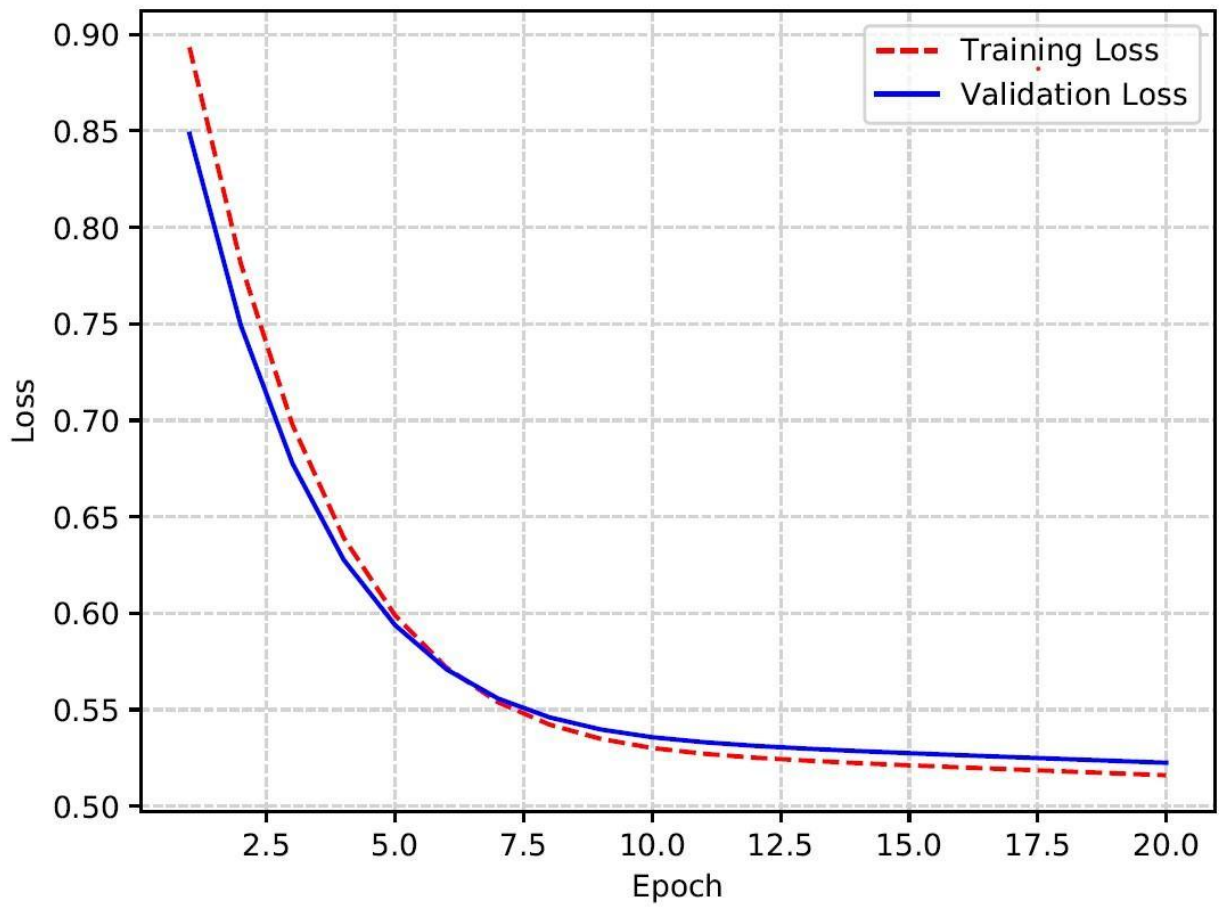


Figura 2.5

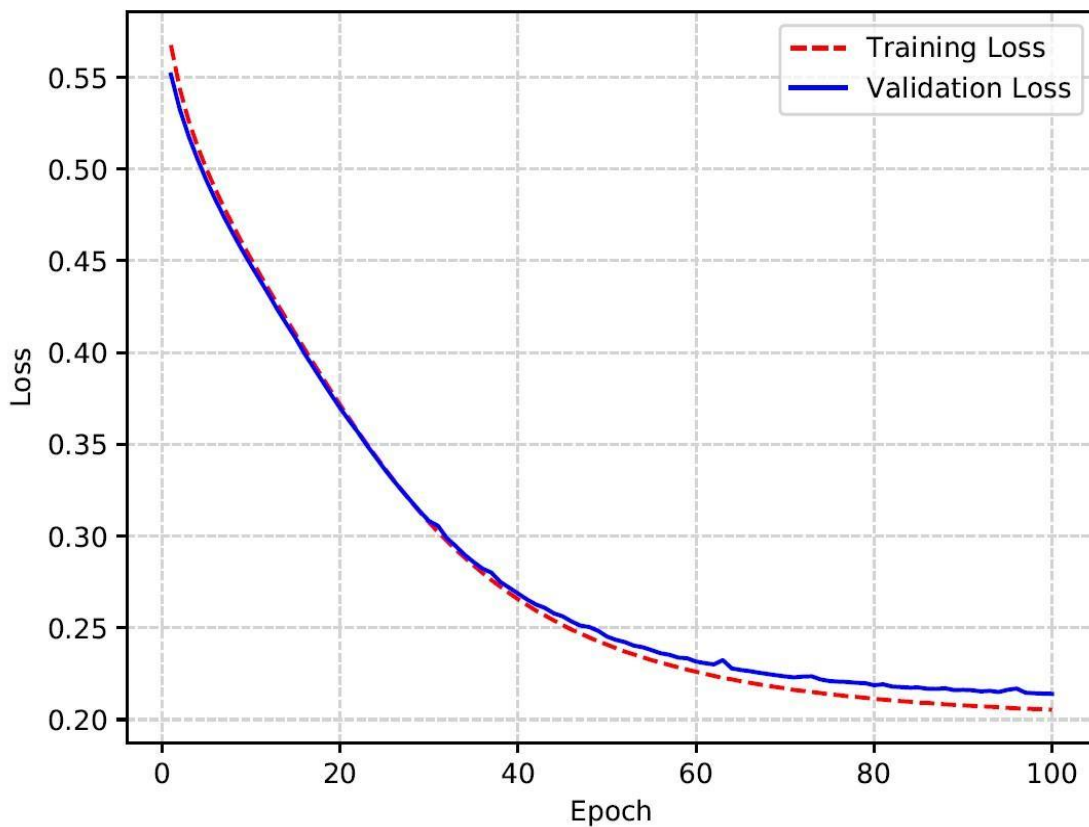


Figura 2.6

L'addestramento a cui è relativa l'ultimo andamento del loss è effettuato su un campione simulato particolarmente ampio, di 10,000 macchine. Di queste 8,400 sono state impiegate per l'addestramento, e le rimanenti 1600 per il test, i cui risultati sono sintetizzati dalla matrice di confusione, riportata qui a seguito:

	predicted bad	predicted good
actual bad	88604	18605
actual good	169	462222

sensitivity	0.826
specificity	1.000

Ovviamente il guasto è un evento raro rispetto al funzionamento regolare, è pertanto ragionevole che il modello tenti a sottostimare la possibilità di guasto, mancando di predirne alcuni casi, viceversa i casi di macchine ben funzionanti di cui si è erroneamente preventivato il malfunzionamento sono molto pochi. Le caratteristiche complessive, sintetizzate come *sensitivity* e *specificity* sono decisamente buone. Ma, occorre ricordare, si tratta solamente di dati simulati. Non è stato possibile accedere a dati reali, e questo è stato il motivo per cui si è dovuti passare ad un altro ambito di sistemi di produzione di energie rinnovabili, dove avevamo garanzie di poter attingere a dati reali.

Capitolo 3

Energia solare e manutenzione

3.1 La manutenzione predittiva negli impianti fotovoltaici

Come descritto nell'introduzione è sopravvenuta la necessità di un cambio di partner industriale per l'impossibilità a proseguire la ricerca su dati reali, un'esigenza che alla fine si è rivelata utile, anzitutto per ampliare l'esplorazione dell'ambito della manutenzione preventiva in almeno due suoi settori, e poi perché quello di cui parliamo, il fotovoltaico, ha oggi un'importanza ancor maggiore rispetto all'eolico.

Tra le forme di energie rinnovabili utilizzate per la produzione di energia elettrica quella solare riveste un ruolo centrale⁸⁰.

Tutti gli impianti fotovoltaici connessi alla rete possiedono sistemi di protezione contro guasti elettrici e condizioni di funzionamento anomale. In particolare, ci sono requisiti tecnici e prestazionali delle protezioni che implementano un confine fra rete di distribuzione pubblica e impianto fotovoltaico; le norme che disciplinano tali protezioni sono la CEI 0-16 e CEI 0-21⁸¹.

⁸⁰ Hansen, K., Breyer, C., & Lund, H. (2019). Status and perspectives on 100% renewable energy systems. *Energy*, 175, 471-480. [10] Child, M., Kemfert, C., Bogdanov, D., & Breyer, C. (2019). Flexible electricity generation, grid exchange and storage for the transition to a 100% renewable energy system in Europe. *Renewable energy*, 139, 80-101. [11] Plessmann, G., & Blechinger, P. (2017). How to meet EU GHG emission reduction targets? A model based decarbonization pathway for Europe's electricity supply system until 2050. *Energy Strategy Reviews*, 15, 19-32. [12] Löffler, K., Burandt, T., Hainsch, K., & Oei, P. Y. (2019). Modeling the low-carbon transition of the European energy system—a quantitative assessment of the stranded assets problem. *Energy Strategy Reviews*, 26, 100422.

⁸¹ Italiano, C. E. (2016). Reference technical rules for the connection of active and passive consumers to the HV and MV electrical networks of distribution Company. (CEI 0–16.) [22] Italiano, C. E. (2019). Reference Technical Rules for the Connection of Active and Passive Users to the LV Electrical Utilities. (CEI 0–21)

Durante la loro vita, infatti, gli impianti fotovoltaici sono soggetti a un normale degrado dei loro componenti e sono caratterizzati di conseguenza da perdite di potenza con decrementi della produzione attesa⁸²

Fattori come deiezioni animali, pollini e polveri possono provocare danni alla superficie trasparente a protezione degli elementi captanti del sensore a cella solare e/o del generatore; Differenze di tecnologia delle celle solari tra sensore e moduli fotovoltaici nel caso sia presente un sensore ad effetto fotovoltaico. Esistono differenti tipi di celle solari per moduli fotovoltaici: in silicio policristallino, monocristallino e celle a film sottile. A volte si può verificare che in un impianto viene installato un sensore con cella fotovoltaica con una tecnologia differente rispetto a quella delle celle dei moduli presenti.

Il modulo fotovoltaico è composto da più celle fotovoltaiche in grado di convertire, mediante effetto fotovoltaico, l'energia solare incidente in energia elettrica, che viene poi convogliata tramite i cosiddetti busbar⁸³ e trasportata alla junction box, una scatola di giunzione posta dietro i pannelli che contiene i diodi di bypass e i connettori. È fondamentalmente un piccolo involucro resistente alle intemperie situato sul lato posteriore del modulo. È qui che avviene la connessione tra il modulo e il circuito esterno, quindi deve essere protetto da umidità e sporco per garantire il corretto funzionamento. Nei primi modelli di pannelli fotovoltaici si contavano 2-3 busbar, mentre ora si trovano in commercio celle fotovoltaiche che possiedono fino a 12 busbar. Il numero di celle collegate in serie dipende dalla dimensione dei pannelli solari, i valori tipici per le tecnologie cristalline variano tra 60-96 celle. Solitamente, nel processo di fabbricazione, la giunzione P-N (creata tra due regioni con drogaggio

⁸² Jordan, D. C., Silverman, T. J., Wohlgemuth, J. H., Kurtz, S. R., & VanSant, K. T. (2017). Photovoltaic failure and degradation modes. *Progress in Photovoltaics: Research and Applications*, 25(4), 318-326.

⁸³ Si tratta di conduttori elettrici realizzati non mediante i consueti cavi, ma con delle barre metalliche rigide, e sono in grado di sopportare flussi di correnti molto elevati.

opposto all'interno del materiale semiconduttore ed è quindi caratterizzata da un'asimmetria elettronica.) viene trattata con un rivestimento antiriflesso che adatta gli indici di rifrazione del silicio e del vetro. Inoltre, ogni cella è dotata di un contatto posteriore e anteriore, costituito da una piastra elettrodica piatta nella parte posteriore e una combinazione di busbars e fingers nella parte anteriore. Questi contatti sono necessari per collegare tra loro le diverse celle e per creare il circuito esterno in cui può fluire la corrente generata.

Il comportamento di una cella solare è strettamente correlato alle condizioni atmosferiche esterne. In questo senso, il primo aspetto da tenere in considerazione è la relazione tra il comportamento della cella solare e l'effetto fotovoltaico. All'aumentare del flusso di fotoni aumenta anche il flusso di elettroni, il che significa che la forma della curva caratteristica I-V è fortemente correlata alle condizioni specifiche dell'irradianza. Inoltre, poiché la cella solare è realizzata in materiale semiconduttore, il suo comportamento dipenderà fortemente dalla temperatura della cella solare. (introdurre la corrosione)

Il campo fotovoltaico produce energia in corrente continua (DC), corrente continua (DC), la quale presenta problemi ad essere trasportata su lunghe distanze, per via dei possibili inneschi di archi voltaici, che vengono invece annullati naturalmente con la corrente alternata (AC), la stessa in uso, a bassa tensione, nelle comuni abitazioni.

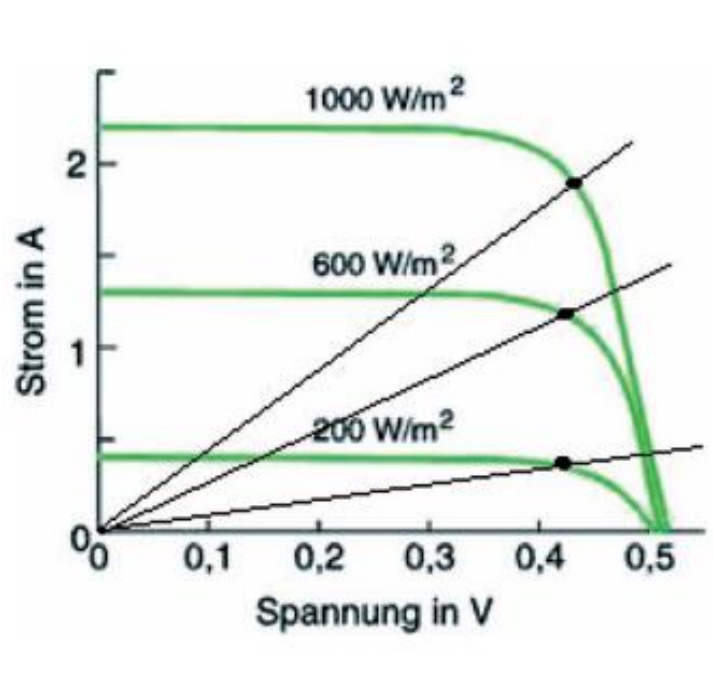
- **3.1.1 Che cos'è un Inverter**

L'inverter è un componente fondamentale, in quanto si occupa della trasformazione della corrente continua in corrente alternata alla tensione di rete di 230 V. Gli inverter vengono confrontati fra loro soprattutto sulla base di tre grandezze: - la potenza continua, espressa in W, che rappresenta la quantità di energia fornita in modo

continuo, ora per ora; - la potenza di picco, espressa in W_p , ovvero quanta potenza e per quanto a lungo può essere fornita per avviare motori o altri carichi; - l'efficienza, cioè il rapporto tra l'energia in ingresso al dispositivo e l'energia effettivamente disponibile all'uscita.

Spesso all'inverter viene integrato un sistema di controllo software che consente di estrarre dai pannelli solari la massima potenza disponibile in qualsiasi condizione meteorologica.

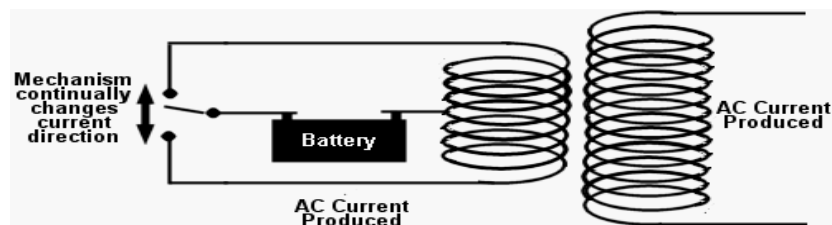
Come prima cosa, l'inverter deve adattare la propria impedenza di ingresso a quella del generatore fotovoltaico, la cui caratteristica non è per niente lineare.



Curve caratteristiche di un generatore fotovoltaico al variare dell'irraggiamento solare

Si può notare che al variare dell'irraggiamento solare corrispondono altrettante curve tensione-corrente del generatore fotovoltaico. In assenza di fenomeni particolari,

ogni curva contiene un punto in cui la potenza è massima ed è quindi necessario che l'inverter sia in grado di variare la propria impedenza di ingresso per intercettare tale punto. La flessibilità del circuito di ingresso dell'inverter non si limita alla ricerca del punto di massima potenza. In particolare, la finestra di tensione deve essere piuttosto ampia al fine di far fronte alle variazioni di tensione del generatore fotovoltaico con la temperatura e permettere una certa libertà nella configurazione impiantistica. Riguardo a questo aspetto, negli inverter commerciali, il rapporto tra le tensioni massime e minime è normalmente compreso tra 1:2 e 1:3



3.1.2 Reti neurali per la manutenzione degli impianti fotovoltaici

Come ampiamente trattato nei capitoli precedenti l'utilizzo di modelli deep per migliorare l'efficienza di impianti eolici e predire il guasto delle turbine è stato, seppur con bassa frequenza, già sperimentato e le ricerche sulla manutenzione predittiva degli impianti eolici vantano poche ma buone sperimentazioni.

Diverso è il caso degli impianti fotovoltaici dove le ricerche sull'utilizzo del deep learning per predirne i guasti sono ancora insufficienti.

Chen et al.⁸⁴ propongono un modello di rete deep (ResNet) per la manutenzione di pannelli fotovoltaici che utilizzano l'uscita I-V (corrente e tensione). Appia et

⁸⁴ Chen Z, Chen Y, Wu L, et al. Deep residual network based fault detection and diagnosis of photovoltaic arrays using current-voltage curves and ambient conditions. *Energy Conv Manage* 2019;198. 111793-.

al.⁸⁵utilizzano la memoria a lungo termine (LSTM)⁸⁶ per estrarre i guasti e le loro caratteristiche utilizzando un modello tipo softmax. Lu et al. convertono la corrente originale e i segnali di tensione in diagrammi di sequenza bidimensionali come l' input di una rete neurale convoluzionale (CNN)⁸⁷.

Liu e colleghi, propongono un metodo di diagnosi dei guasti diverso, basato su SAE (stacked auto-encoder) molto simile a quello di Chen Z. (2019)⁸⁸.

L'autoencoder (AE) si riferisce a una rete neurale connessa a tre strati, che comprende il livello di input, il livello deep e il livello di output per l'apprendimento non supervisionato. Lo strato di ingresso e lo strato nascosto formano la rete codificatore $h = f(x)$, e lo strato deep e il livello di output costituiscono la rete di decodifica $y = f(x)$.

Liu e colleghi propongono un piano di preelaborazione dei dati, di estrazione delle caratteristiche da parte del SAE (stacked auto-encoder)⁸⁹. La curva I-V della cella solare è lo strumento migliore per descrivere il comportamento di una cella solare per una data coppia di irradianza e temperatura. Come accennato nel paragrafo precedente, il comportamento di una cella solare è strettamente correlato alle condizioni atmosferiche esterne. In questo senso, il primo aspetto da tenere in considerazione è la relazione tra il comportamento della cella solare e l'effetto fotovoltaico. All'aumentare del flusso di fotoni aumenta anche il flusso di elettroni, il che significa

⁸⁵ Appiah AY, Zhang X, Ayawli BBK, et al. Long short-term memory networks based automatic feature extraction for photovoltaic array fault diagnosis. *IEEE Access* 2019:1.

⁸⁶ Vedi oltre, p. 67.

⁸⁷ Lu, X., Lin, P., Cheng, S., Lin, Y., Chen, Z., Wu, L., & Zheng, Q. (2019). Fault diagnosis for photovoltaic array based on convolutional neural network and electrical time series graph. *Energy Conversion and Management*, 196, 950-965.

⁸⁸ Liu, Y., Ding, K., Zhang, J., Li, Y., Yang, Z., Zheng, W., & Chen, X. (2021). Fault diagnosis approach for photovoltaic array based on the stacked auto-encoder and clustering with IV curves. *Energy Conversion and Management*, 245, 114603.

⁸⁹ Che come dice la locuzione non è altro che il concetto di autoencoder replicato diverse volte, con ogni stadio autoencoder che prende il suo ingresso dall'uscita del precedente

che la forma della curva caratteristica I-V è fortemente correlata alle condizioni specifiche dell'irradianza. Inoltre, poiché la cella solare è realizzata in materiale semiconduttore, il suo comportamento dipenderà fortemente dalla temperatura della cella solare. A valori di temperatura più elevati, il gap energetico del materiale si riduce e quindi aumenta la porzione sfruttabile dello spettro solare.

Il primo step prevede l'acquisizione dei dati. La quantità di curve I-V viene ottenuta con un modello di simulazione. In seguito è prevista la preelaborazione dei dati. Le curve I-V in condizioni ambientali diverse vengono trasformate in STC per eliminare l'influenza dell'ambiente.

3.2 Sperimentazione preliminare di previsione guasti in impianti fotovoltaici

3.2.1 Collaborazione con Ecoprime

In seguito alla collaborazione tra il Dottorato di Ricerca in Scienze Cognitive del Dipartimento COSPECS (Università di Messina) e Ecoprime Italia S.r.l. abbiamo acquisito i dati utili alla nostra ricerca.

Ecoprime ci ha fornito inizialmente la raccolta di tutti i ticket degli impianti fotovoltaici da loro gestiti, che comprende 18,000 record di 45 campi. Da una prima lettura diretta di questo archivio risultava evidente la natura estremamente eterogenea degli interventi, la maggior parte dei quali non era diretta conseguenza del guasto di uno specifico componente critico. Inoltre, dal genere di descrizione testuale del ticket appariva decisamente problematico ipotizzare un'analisi automatica che permettesse di selezionare i casi di interesse. Per poter effettuare una prima sperimentazione, l'ing. Coco di Ecoprime si è dedicato ad una selezione manuale dei soli casi in cui il guasto fosse sicuramente connesso ad un componente critico, in particolare l'inverter, il dispositivo su cui vi è maggior interesse a una possibile previsione di

guasto. La selezione realizzata dall'ing. Coco consisteva di 76 casi. Essi presentavano il problema di un'ampia distribuzione su impianti diversi, con pochissimi ticket per impianto. Questa condizione esponeva decisamente i dati ad una dipendenza dalle tipologie di impianto e loro localizzazione, in grado di sovrastare e confondere ogni possibile variazione dei dati contenenti informazioni previsionali sull'imminenza di guasti.

Pertanto, si è deciso di partire sperimentando un modello sui soli impianti che avessero un numero minimo di ticket, e sono Aligrup, Enim1 ed Enim2. Di questi si sono ulteriormente presi in considerazione solamente ticket legati con certezza all'inverter, avendo alla fine 25 casi.

Si tratta certamente di un numero molto esiguo, su cui pende comunque la variabilità delle tipologie di impianti e locazione, anche se ristretto a tre.

Inoltre Enim1 ed Enim2 impiegano due diversi modelli di inverter, a loro volta diversi da quello di Aligrup, anche se della stessa casa costruttrice.

Stante questo limite, è stato impossibile ripartire ulteriormente l'insieme dei casi, in modo da avere - come buona pratica - un insieme di dati per l'addestramento, e uno diverso per validare il modello.

L'indicazione che ci si può attendere da un numero esiguo di dati ha comunque un suo valore di massima: se i dati disponibili non contengono alcuna informazione che in qualche modo, anche complesso, risulti predittiva di guasti, allora non ci potrebbe essere convergenza nemmeno in fase di addestramento.

Se invece si ottiene un modello che apprenda a predire in modo robusto, allora vi sono con buona probabilità elementi predittivi annidati nei dati, e il modello ha la potenza sufficiente per catturarli. Naturalmente non è possibile stimare quale sia la percentuale di errore del modello, una volta generalizzato a dati mai visti precedentemente. Altre considerazioni verranno fatte nella sezione che illustra i risultati ottenuti.

Per ogni ticket utilizziamo i dati di 48 ore prima del guasto, e quelli in cui l'inverter funziona bene di 96 ore prima nell'anno precedente.

Il lavoro si è strutturato in 4 principali step:

1. Creazione di script per eseguire curl con i parametri appropriati e interrogare meteocontrol.de e ricavare tutti i dati d'interesse.
2. Interfaccia dei dati dal formato jsons al formato H5 organizzati in maniera adeguata per il modello.
3. Sviluppo di un modello deep learning per la previsione del guasto dell'inverter sulla base dei dati giornalieri
4. Ottimizzazione dei parametri del modello sulla base dei risultati

```
112
113 def create_model():
114     """
115     create the full neural model, made of recurrent and dense layers
116
117     return: [keras.engine.training.Model]
118     """
119     x = Input( shape=( cnfg.ns_input, input_size, ) ) # input layer
120     if hasattr( cnfg, "double_rnn" ): # uses two recurrent layers
121         return_sequences = cnfg.double_rnn
122     else:
123         return_sequences = False
124     y = layers.LSTM( # recurrent layer
125         input_size, # dimensionality of the output space
126         activation = cnfg.act_rnn, # activation function to use
127         return_sequences = return_sequences, # return the last output or the full sequence
128         return_state = False, # return the last state in addition to the output
129         stateful = False, # use last state as initial state of next batch
130         unroll = False, # unroll the network
131         name = 'rcrr_1'
132     )( x )
133     if return_sequences: # uses two recurrent layers
134         y = layers.LSTM( # second recurrent layer
135             input_size,
136             activation = cnfg.act_rnn, # flexible choice of activation function in configuration
137             return_sequences = False, # return the last output of the sequence only
138             return_state = False,
139             stateful = False,
140             unroll = False,
141             name = 'rcrr_2'
142         )( y )
143     y = layers.Dense( # first dense layer
144         cnfg.dense_1,
145         activation = cnfg.act_dn1,
146         name = "dense_1"
147     )( y )
148     y = layers.Dense( # second dense layer
149         2, # two possible outputs: working/failure
150         activation = 'softmax', # cast the outputs as probabilities
151         name = "dense_2"
152     )( y )
153
154     return Model( inputs=x, outputs=y, name="nn_model" ) # the complete model
155
```

```

178
179 def train_model( model, gen, val=None, callback=None ):
180     """
181     train the model using a given dataset
182
183     model:         [keras.engine.training.Model]
184     gen:           [tuple] ( x, y ) dataset with temporal data as x, and labels (working/failure) as y
185     val:           [tuple] ( x, y ) validation, may be missing
186     callback:      [keras.callbacks] task called during training, typically to save the best intermediate model
187
188     return:        [keras.callbacks.History]
189     """
190     # the loss function and the optimizer are flexible, and set in the configuration file
191     loss = loss_code[ cnfg.loss_code ]
192     optimizer = optimiz_code[ cnfg.optimiz_code ]( learning_rate=cnfg.lrate )
193     x, y = gen
194
195     # compile the neural model with the loss and the optimizer
196     model.compile(
197         optimizer = optimizer,
198         loss = loss
199     )
200
201     history = model.fit(
202         x, # start training
203         y, # dataset inputs
204         validation_data = ( x, y ), # dataset outputs
205         batch_size = cnfg.batch_size, # in cas of lack of data, use the same dataset for validation
206         epochs = cnfg.n_epochs, # size of training batches
207         shuffle = True, # number of training epochs
208         callbacks = callback, # shuffle samples at every epoch
209         verbose = 2 # additional task executed during training
210     )
211
212     return history # a set of information about training
213
214

```

3.2.2 Interrogazione meteocontrol.de

Ai fini del prelievo dei dati degli inverter che hanno subito guasti si sono realizzati degli script preliminari per ricavare le chiavi identificative degli impianti, degli inverter e delle loro misure. In particolare, abbiamo scelto le misure utilizzando un ampio criterio senza riferimenti a un modello causale predefinito sfruttando ogni parametro con una variazione significativa. Si sono scelte le seguenti misure:

- E_INT: energia generata nell'intervallo temporale
- I_DC: corrente DC
- U_DC: tensione DC
- P_AC: power AC

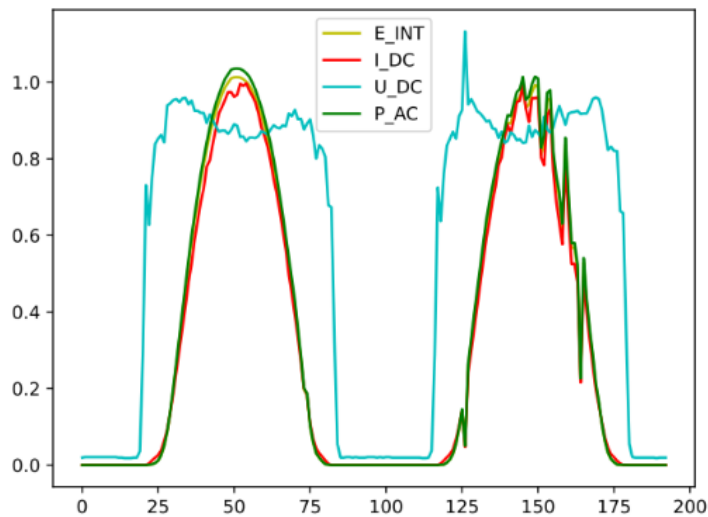
In alcuni casi si sono usati dei giorni dell'anno precedente diversi dall'attuale per ovviare ad alcuni buchi di dati.

Dei tre impianti sono stati recuperati i dati relativi ai seguenti numeri di ticket:

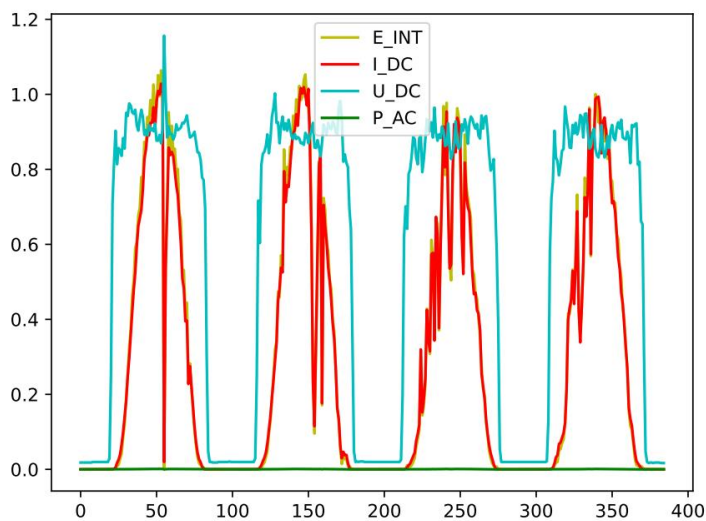
- Aligrup: 13
- Enim1: 6
- Enim2: 6

Riportiamo a titolo di esempio alcuni dei dati originali che fanno parte dell'archivio H5, ricavati tramite interfaccia *meteocontrol.de* dai dati grezzi degli impianti. Per ogni ticket si sono illustrati gli andamenti delle quattro misure di interesse identificate con i codici descritti precedentemente. I grafici con prefisso "fault" sono due giornate a

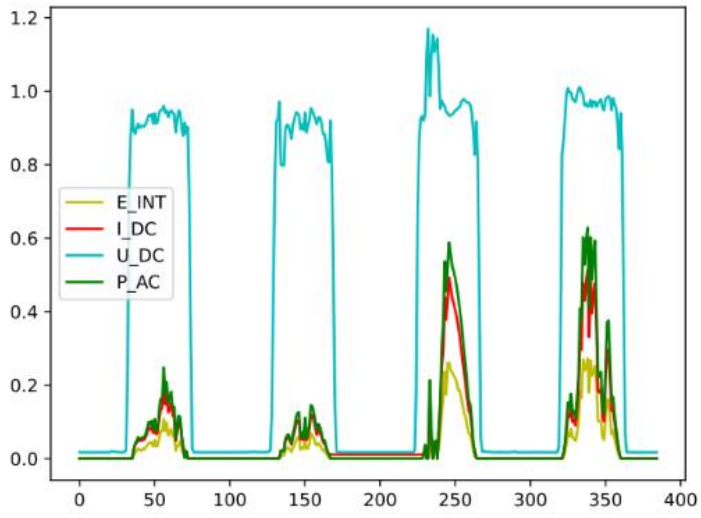
ridosso del guasto, mentre quelli con prefisso "work" sono quattro giornate di un anno precedente, quando l'impianto funzionava bene. Naturalmente la produzione di energia avviene solo nell'intervallo diurno di irraggiamento solare, e anche tutte le misure correlate (tensione, corrente) sono significative soltanto in quell'intervallo. Esiste una notevole variabilità legata alla condizione di irraggiamento solare, ed è evidente come tale variabilità domini ogni minimo presagio di malfunzionamento, percepibile dal semplice confronto visivo dei grafici.



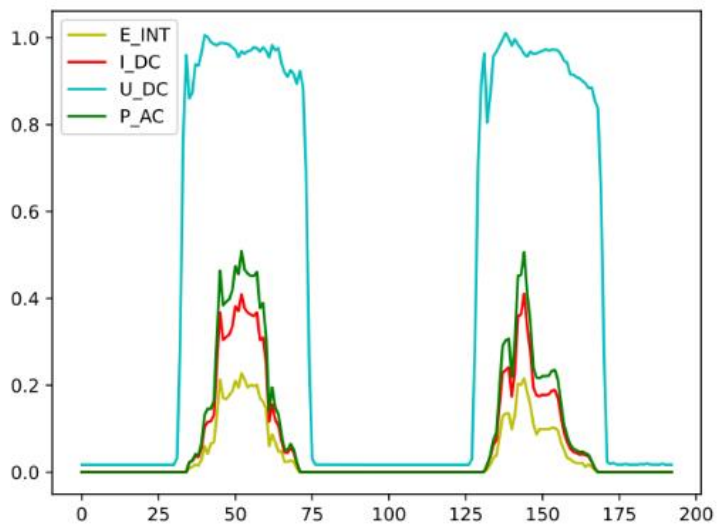
work_enim_01



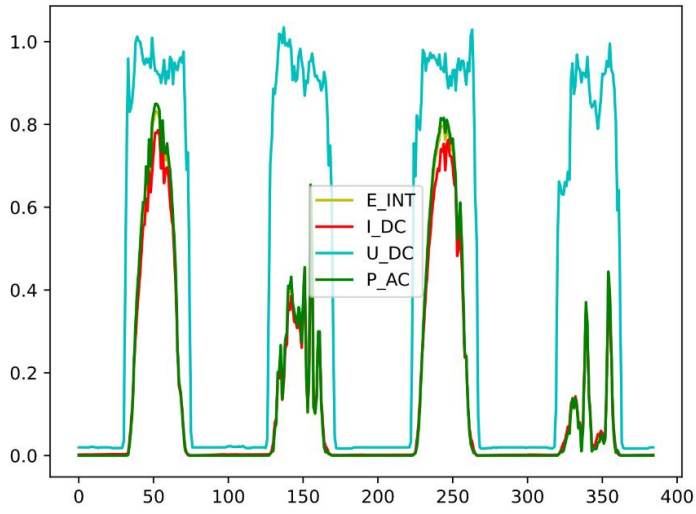
fault_enim_01



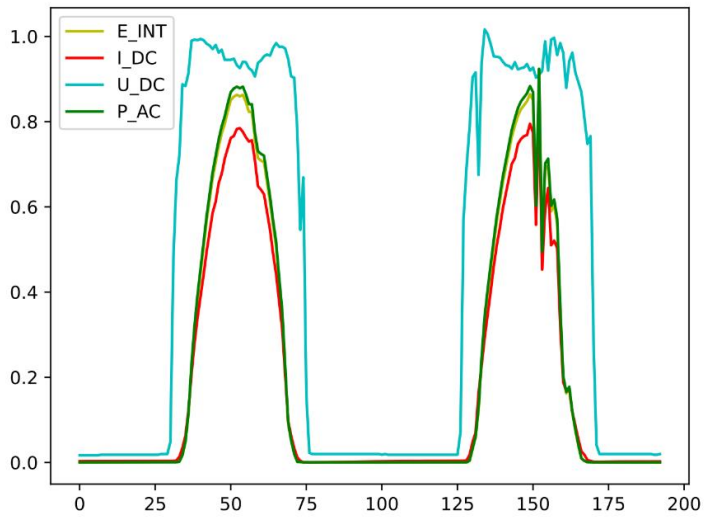
work_alias_01



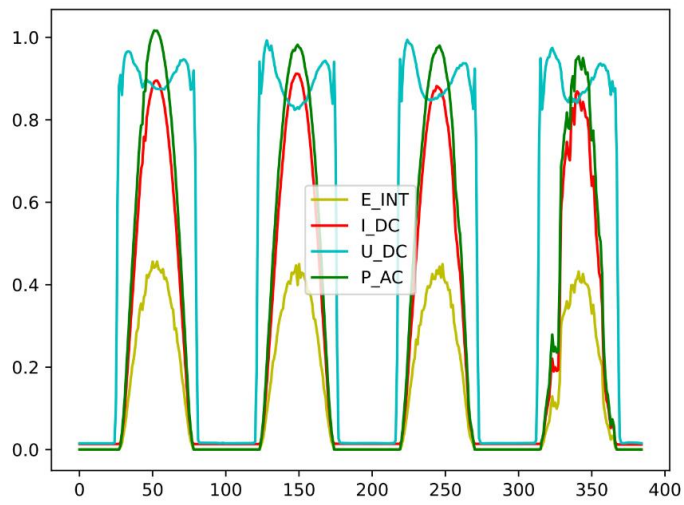
fault_alias_01



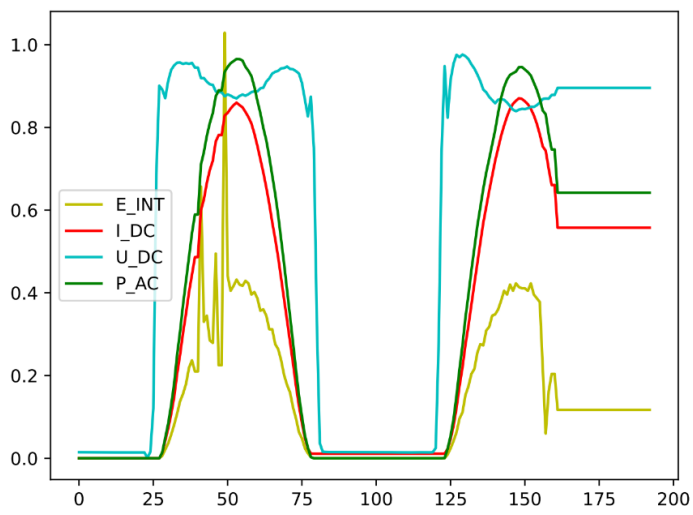
work_enim_05



fault_enim_05



work_ali_05



fault_ali_05

3.2.3 Interfaccia dei dati

I dati recuperati da Meteocontrol.de hanno una risoluzione temporale di un quarto d'ora, pertanto, un'intera giornata comprende 96 campioni, di questi se ne utilizzano 32 che coprono la parte centrale della giornata. Abbiamo normalizzato la scala di tutte le misure nell'intervallo [0-1], con fattori di normalizzazione separati per ogni impianto.

Nel modello abbiamo costruito i seguenti dataset:

ali: tutti i dati relativi ad Aligrup

enim: tutti i dati relativi a Enim1 e a Enim2

ali_enim: l'unione dei dataset ali e enim

L'attuale ripartizione dei campioni temporali risulterebbe sbilanciata, in quanto abbiamo solamente due giornate associate a situazioni che preludono un guasto, contro quattro giornate associate a funzionamento regolare. Non è prudente voler prendere quattro giornate anche nell'immediatezza del guasto in quanto si andrebbe troppo indietro nel tempo e quindi in giornate in cui ancora non c'era nessun segnale premonitore. Per ovviare a questo sbilanciamento abbiamo utilizzato quella che si chiama "data augmentation", che si basa sul creare dati fittizi, un aumento artificiale dei dati partendo dai dati esistenti. Come piccole perturbazioni casuali dei dati originali disponibili, dai due giorni precedenti se ne sono ricavati altri due in modo da avere quattro campioni per ogni guasto simmetricamente con i quattro campioni di buon funzionamento.

I dati nell'archivio in formato H5 sono organizzati in modo da avere una coppia (input, output) per ogni campione con le seguenti dimensioni:

- input: 32 x 4
- output: 2

Il vettore di output ha dimensione 2 in quanto esprime la condizione di *guasto/funzionante*, con i seguenti valori:

- guasto [1, 0]
- funzionante [0, 1]

3.2.4 Modello

Si è costruita un'architettura che parte dal presupposto che i 32 vettori di misure fanno parte di una sequenza temporale e pertanto lo strumento più appropriato per il loro trattamento sono le reti ricorsive. Il secondo presupposto è che il genere di responso desiderato dal modello consiste in due valori che esprimono il primo la probabilità di guasto e il secondo la probabilità alternativa ovvero di non guasto. avendo pertanto fissato questi due elementi progettuali, per la parte interna del modello si è predisposta una ampia flessibilità architettonica. Si tratta di una scelta necessaria, trattandosi di un'applicazione del tutto nuova per la quale non vi sono esempi funzionanti disponibili, si impone quindi di lasciare la maggior libertà possibile alla sperimentazione. Per questo scopo si è strutturato il software in modo da avere un modulo che costituisce la configurazione dei parametri del modello, e un altro modulo che interfaccia la configurazione con il modello stesso.

Come rete ricorsiva d'ingresso, si è utilizzata la LSTM (Long Short Term Memory)⁹⁰, che è la più potente nella gamma delle reti ricorsive abitualmente utilizzate.

Assodata quindi la necessità di usare anzitutto una rete ricorsiva, il sistema di sviluppo Keras qui utilizzato ne mette a disposizione diverse, la più semplice è la rete ricorsiva originale di Elman, del genere illustrato in figura 1.2. La più potente è la LSTM. Come introdotto nella sezione 1.2, la LSTM è uno dei tentativi più riusciti di imitazione di strategie cognitive in reti di neuroni artificiali, nella fattispecie la strategia di filtrare

⁹⁰ Vedi pp. 29-30.

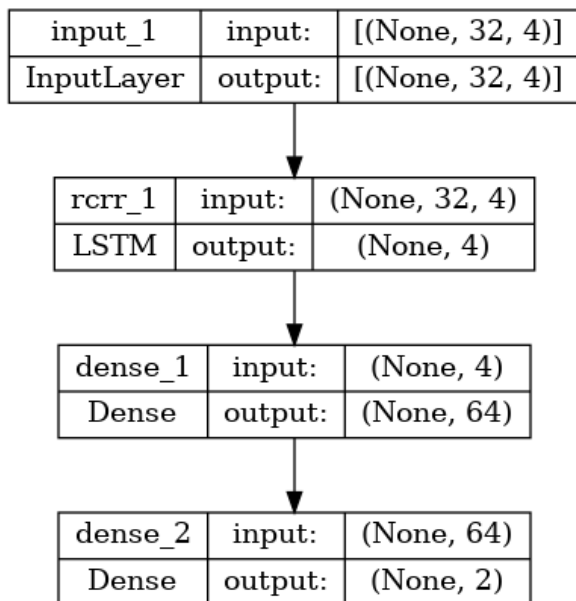
selettivamente informazioni da ricordare nonostante la loro datazione remota, e informazioni da scartare anche se recenti. Come si era visto, l'ambizioso traguardo per cui era stata ideata questa rete ricorsiva era il linguaggio naturale, ma il fatto che basilamente le reti neurali artificiali conoscono solamente vettori di numeri - e non parole - comporta il gran vantaggio di poterle esportare ad ambiti che con le parole non hanno nulla a che fare. Quindi anche misurazioni di energia elettrica, tensione, corrente, come nel caso in esame. Quel che si vuole andare a ricercare nelle sequenze temporali di queste misure sono configurazioni istantanee di segnali, che meritano di essere tenuti a mente, e confrontati con altri passati e recenti. Stavolta il criterio su cosa mantenere in memoria e cosa buttar via, è quanto le configurazioni di segnali siano predittive ai fini di discriminare guasto o non guasto. Pertanto, la via cognitiva qui adottata come strategia elettiva suggerisce proprio di utilizzare LSTM come primo strato del modello. Il principio di manipolare vettori numerici, comune a tutte le reti neurali artificiali, non solo consente di inserire uno strato LSTM che prenda direttamente in ingresso le sequenze temporali delle 4 misure in gioco, ma di replicare la sua funzione su dei nuovi input astratti, costituiti dall'uscita della stessa LSTM. È come se applicassimo sempre la funzione di gestire una risorsa di memoria limitata vagliando cosa ricordare e cosa dimenticare, ma non più ai dati sensoriali diretti, bensì a informazioni elaborate, che già riflettono un primo stadio di memorizzazione controllata. Il modello attua questa configurazione a stack di 2 LSTM come un'opzione, è quindi possibile confrontare modelli con un solo strato LSTM, e modelli con due, organizzati come descritto.

La prima produce un'uscita di dimensioni 32x4 e la seconda prende in ingresso questa uscita producendo un vettore di 4 elementi. Nel caso in cui non si scelga il raddoppio della LSTM sarà la prima a produrre il vettore di 4 elementi. Segue uno strato interamente connesso costituito da un numero di neuroni definibile nel file di configurazione. Altri parametri configurabili riguardano la scelta delle funzioni non

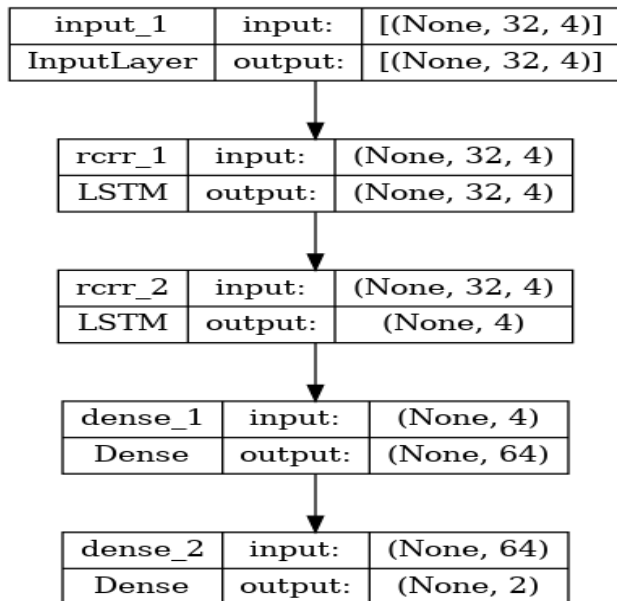
lineari di attivazione neurale, il numero di epoche, il learning-rate, la scelta dell'algoritmo di ottimizzazione per l'addestramento, la dimensione dei batch per l'addestramento.

Illustriamo qui di seguito mediante schemi a blocchi le quattro architetture che sono state utilizzate per produrre i risultati:

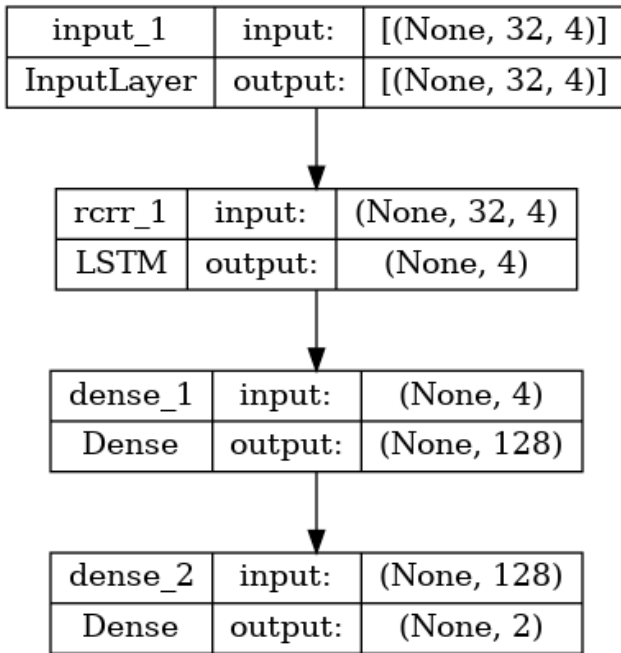
Arch1



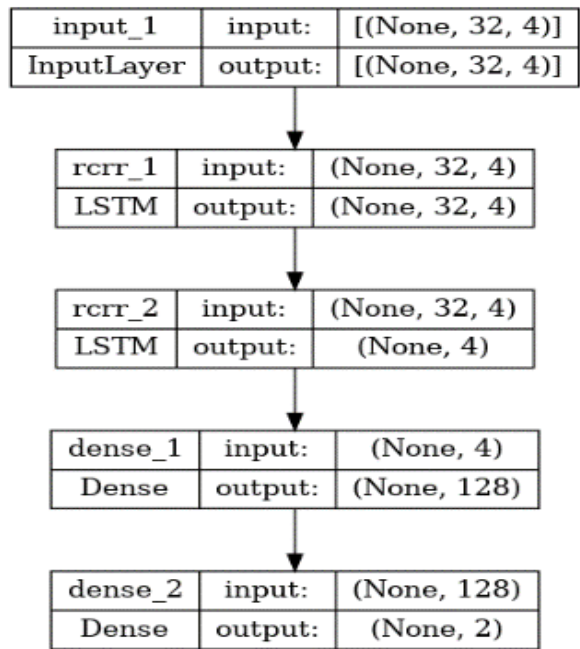
Arch2



Arch3



Arch4



3.2.5 Risultati

Una fase preliminare della sperimentazione è consistita in uno screening veloce di tutti i parametri solamente sul dataset ali (quello con più campioni), usando pertanto un numero basso di epoche senza puntare a risultati ottimali ma solo per avere un'idea comparativa. Si omette di documentare interamente questa parte. Nella fase principale si è passati ad un numero adeguato di epoche di addestramento (30000) che comportano circa 80 ore di addestramento, limitandosi al perfezionamento solamente dei parametri più critici del modello mantenendo per gli altri fissi i valori stimati nella fase preliminare.

I parametri fissi sono:

- learning rate: $1e-4$
- batch size: 4
- optimizer: ADAM
- activation: RELU

Sono stati variati i seguenti parametri:

- raddoppio LSTM: si/ no
- numero neuroni dello strato intermedio: 64/ 128 per un totale di 4 combinazioni.

La tabella seguente riassume i parametri fondamentali delle quattro architetture di cui sono stati mostrati precedentemente gli schemi a blocchi, dove indicheremo con “double_rnn” il raddoppio e con “dense” il numero dei neuroni.

I quattro modelli sono stati sperimentati separatamente per i dataset ali e enim.

Il dataset complessivo ali_enim avendo più campioni richiede un tempo maggiore di addestramento e pertanto è stato sperimentato solamente con la combinazione di parametri migliore ottenuta con i dataset separati.

Type	Arch1	Arch2	Arch 3	Arch 4
Dense:	64	64	128	128
Double_rnn:	False	True	False	True

Directory	Epochs:	H5_file:	Model	Double_rnn	Dense_1	Sensitivity	Specificity	Correct predictions
22-11-06_10-29	30000	"ali"	Arch1	False	64	0.923	0.731	0.83
22-11-04_19-21	30000	"ali"	Arch2	True	64	0.942	0.577	0.76
22-11-07_08-49	30000	"ali"	Arch3	False	128	0.885	0.769	0.83
22-11-03_16-50	30000	"ali"	Arch4	True	128	0.981	0.846	0.91
22-11-11_17-01	30000	"enim"	Arch1	False	64	0.792	0.958	0.88
22-11-14_09-21	30000	"enim"	Arch3	False	128	0.917	0.896	0.91
22-11-12_11-35	30000	"enim"	Arch2	True	64	1.000	1.000	1.00
22-11-08_15-12	30000	"enim"	Arch4	True	128	1.000	1.000	1.00
22-11-09_15-26	30000	"ali_enim"	Arch4	True	128	0.960	0.960	0.96

Trattandosi di un modello sostanzialmente diagnostico di un evento negativo, la sintesi statistica idonea risulta la consueta matrice di confusione 2x2, di cui i caratteristici parametri derivati sono la sensibilità (rapporto tra veri positivi predetti e attuali positivi, e la specificità (rapporto tra veri negativi predetti e numero complessivo di campioni). Questi due valori sono riportati in tabella, insieme con l'accuratezza complessiva (rapporto tra predizioni corrette e numero complessivo di campioni). Come si può evincere dalla tabella, i risultati risultano nel complesso incoraggiati, ben oltre le aspettative. Appaiono leggermente più problematici i risultati per l'impianto Aligrup, per il quale occorre far ricorso alla architettura più potente, la Arch4, per avere un risultato di elevata accuratezza. Per il dataset "enim" (unione dei due impianti Enim1 ed Enim2), solamente il modello più semplice, Arch1, risulta inadeguata, ma subito gli altri sono più che sufficiente, con addirittura accuratezza esatta usando sia Arch2 che Arch4. Si sarebbe potuto ipotizzare che Aligrup risulti leggermente più difficile semplicemente per il numero relativamente maggiore di campioni, ipotesi subito smentita dall'esperimento più ardito: un solo modello addestrato contemporaneamente sui tra impianti, il dataset "ali_enim", che quindi ha come numero di campioni la somma degli impianti Aligrup, Enim1 ed Enim2. Questo modello ha addirittura prestazioni leggermente migliori rispetto a quello specializzato per Aligrup. Più precisamente, rispetto al dataset "ali" con "ali_enim" si perde in sensibilità, ma compensando con la specificità. I due valori sono uguali, 0.96, che quindi rappresentano un perfetto equilibrio di probabilità di errore del modello. Come anticipato, questi risultati sono certamente ben più generosi rispetto alle prestazioni che il modello potrebbe avere, se validato su degli eventi sconosciuti. Tuttavia, anche in assenza di questo passaggio, impossibile per la limitatezza dei dati, il risultato pone delle buone premesse sulla possibilità che un'architettura neurale

come quella proposta, sia in grado di catturare dai dati di uno o due giorni precedenti il guasto, degli elementi che consentano un certo grado di predizione.

Risultato di 22-11-06_10-29

ticket y00d-1 y00d-2 y00a-1 y00a-2 y-1d-1 y-1d-2 y-1d-3 y-1d-4 correct

13269	True	False	False	True	True	True	False	False	4
12633	False	True	True	False	True	True	False	False	4
3811	True	True	True	True	False	False	False	True	5
5897	True	True	True	True	True	True	False	True	7
14422	True	True	True	True	True	True	True	False	7
7156	True	True	True	True	True	True	True	True	8
7125	True	True	True	True	True	True	True	True	8
4196	True	True	True	True	True	True	True	True	8
4338	True	True	True	True	True	False	True	True	7
4885	True	True	True	True	True	True	False	False	6
3845	True	True	True	True	True	True	False	False	6
4917	True	True	True	True	True	True	True	True	8
4449	True	True	True	True	True	True	True	True	8

fraction of correct predictions: 0.83

Risultato di 22-11-04_19-21

ticket y00d-1 y00d-2 y00a-1 y00a-2 y-1d-1 y-1d-2 y-1d-3 y-1d-4 correct

13269	True	True	True	True	True	True	True	False	7
12633	True	True	True	True	False	False	False	False	4
3811	True	True	True	True	False	False	False	False	4
5897	True	True	True	True	False	False	False	False	4
14422	True	True	True	False	True	True	True	False	6
7156	True	True	True	True	True	True	True	True	8
7125	True	True	True	True	True	True	True	True	8
4196	True	True	True	True	False	False	True	True	6
4338	True	True	True	True	True	False	True	True	7
4885	True	False	False	True	True	True	False	True	5
3845	True	True	True	True	False	False	False	False	4
4917	True	True	True	True	True	True	True	True	8
4449	True	True	True	True	True	True	True	True	8

fraction of correct predictions: 0.76

Risultato di 22-11-07_08-49

ticket y00d-1 y00d-2 y00a-1 y00a-2 y-1d-1 y-1d-2 y-1d-3 y-1d-4 correct

13269	False	False	False	False	True	True	False	True	3
12633	True	True	True	True	False	True	False	True	6
3811	True	True	True	True	False	True	False	True	6
5897	True	False	False	True	True	False	True	False	4
14422	True	True	True	True	True	True	True	False	7
7156	True	True	True	True	True	True	True	True	8
7125	True	True	True	True	True	True	True	True	8

4196	True	True	True	True	True	False	True	True	7
4338	True	True	True	True	True	True	True	True	8
4885	True	True	True	True	True	True	False	False	6
3845	True	True	True	True	True	True	False	True	7
4917	True	True	True	True	True	True	True	True	8
4449	True	True	True	True	True	True	True	True	8

fraction of correct predictions: 0.83

Risultato di 22-11-03_16-50

ticket y00d-1 y00d-2 y00a-1 y00a-2 y-1d-1 y-1d-2 y-1d-3 y-1d-4 correct

13269	True	True	True	True	True	True	True	True	8
12633	True	True	True	True	True	False	True	True	7
3811	True	True	True	True	False	True	True	False	6
5897	True	True	True	True	True	True	True	True	8
14422	True	True	True	True	True	True	True	False	7
7156	True	True	True	True	True	True	True	True	8
7125	True	True	True	True	True	True	True	True	8
4196	True	True	True	True	True	True	True	True	8
4338	True	True	True	True	True	False	True	True	7
4885	True	False	True	True	True	True	False	True	6
3845	True	True	True	True	False	True	True	False	6
4917	True	True	True	True	True	True	True	True	8
4449	True	True	True	True	True	True	True	True	8

fraction of correct predictions: 0.91

Risultato di 22-11-11_17-01

ticket y00d-1 y00d-2 y00a-1 y00a-2 y-1d-1 y-1d-2 y-1d-3 y-1d-4 correct

9356	True	True	True	True	True	True	True	True	8
14378	True	True	True	True	True	False	True	True	7
6649	True	True	True	True	True	True	True	True	8
1491	True	True	True	True	True	True	True	True	8
13448	True	True	True	True	True	True	True	True	8
5836	True	True	True	True	True	True	True	True	8
9688	True	True	True	True	True	True	True	True	8
14145	True	False	False	True	True	True	True	True	6
11959	True	True	True	True	True	True	True	True	8
11047	False	False	False	False	True	True	False	True	3
12084	False	False	False	False	True	True	True	True	4
11994	True	True	True	True	True	True	True	True	8

fraction of correct predictions: 0.88

Risultato di 22-11-14_09-21

ticket y00d-1 y00d-2 y00a-1 y00a-2 y-1d-1 y-1d-2 y-1d-3 y-1d-4 correct

9356	True	True	True	True	True	True	True	True	8
------	------	------	------	------	------	------	------	------	---

14378	True	True	True	True	True	False	True	True	7
6649	True	True	True	True	True	True	True	True	8
1491	True	True	True	True	True	True	True	True	8
13448	True	True	True	True	True	False	True	True	7
5836	False	False	False	False	True	True	True	True	4
9688	True	True	True	True	True	True	True	True	8
14145	True	True	True	True	True	True	True	True	8
11959	True	True	True	True	True	True	True	False	7
11047	True	True	True	True	True	True	True	True	8
12084	True	True	True	True	True	True	True	True	8
11994	True	True	True	True	False	True	True	False	6

fraction of correct predictions: 0.91

Risultato di 22-11-12_11-35

ticket y00d-1 y00d-2 y00a-1 y00a-2 y-1d-1 y-1d-2 y-1d-3 y-1d-4 correct

9356	True	True	True	True	True	True	True	True	8
14378	True	True	True	True	True	True	True	True	8
6649	True	True	True	True	True	True	True	True	8
1491	True	True	True	True	True	True	True	True	8
13448	True	True	True	True	True	True	True	True	8
5836	True	True	True	True	True	True	True	True	8
9688	True	True	True	True	True	True	True	True	8
14145	True	True	True	True	True	True	True	True	8
11959	True	True	True	True	True	True	True	True	8

11047	True	True	True	True	True	True	True	True	8
12084	True	True	True	True	True	True	True	True	8
11994	True	True	True	True	True	True	True	True	8

fraction of correct predictions: 1.00

Risultato di 22-11-08_15-12

ticket y00d-1 y00d-2 y00a-1 y00a-2 y-1d-1 y-1d-2 y-1d-3 y-1d-4 correct

9356	True	True	True	True	True	True	True	True	8
14378	True	True	True	True	True	True	True	True	8
6649	True	True	True	True	True	True	True	True	8
1491	True	True	True	True	True	True	True	True	8
13448	True	True	True	True	True	True	True	True	8
5836	True	True	True	True	True	True	True	True	8
9688	True	True	True	True	True	True	True	True	8
14145	True	True	True	True	True	True	True	True	8
11959	True	True	True	True	True	True	True	True	8
11047	True	True	True	True	True	True	True	True	8
12084	True	True	True	True	True	True	True	True	8
11994	True	True	True	True	True	True	True	True	8

fraction of correct predictions: 1.00

Risultato di 22-11-09_15-26

ticket y00d-1 y00d-2 y00a-1 y00a-2 y-1d-1 y-1d-2 y-1d-3 y-1d-4 correct

13269	True	True	True	True	True	True	True	True	8
12633	True	True	True	True	False	False	True	True	6
3811	True	True	True	True	True	True	True	True	8
5897	False	True	True	False	True	True	True	True	6
14422	True	True	True	True	True	True	True	True	8
7156	True	True	True	True	True	True	True	True	8
7125	True	True	True	True	True	True	True	True	8
4196	True	True	True	True	True	True	True	True	8
4338	True	True	True	True	True	True	True	True	8
4885	False	False	True	True	True	True	True	True	6
3845	True	True	True	True	True	False	True	True	7
4917	True	True	True	True	True	True	True	True	8
4449	True	True	True	True	True	True	True	True	8
9356	True	True	True	True	True	True	True	True	8
14378	True	True	True	True	True	True	True	True	8
6649	True	True	True	True	True	True	True	True	8
1491	True	True	True	True	True	True	True	True	8
13448	True	True	True	True	True	True	False	True	7
5836	True	True	True	True	True	True	True	True	8
9688	True	True	True	True	True	True	True	True	8
14145	True	True	True	True	True	True	True	True	8

11959	True	True	True	True	True	True	True	True	8
11047	True	True	True	True	True	True	True	True	8
12084	True	True	True	True	True	True	True	True	8
11994	True	True	True	True	True	True	True	True	8

fraction of correct predictions: 0.96

Capitolo 4

Corrosione dei materiali: una fonte di degrado

Un fattore di malfunzionamento che accomuna sia gli impianti eolici descritti nella prima parte di questo elaborato sia i pannelli fotovoltaici è la corrosione dei materiali esposti in ambiente esterno; infatti, per qualunque installazione tecnologica in cielo aperto, una delle possibili fonti di degrado è costituita proprio dalla corrosione. Nel caso delle turbine eoliche, per esempio, il riduttore è un elemento critico dell'intero impianto e il suo guasto può causare lunghi tempi di inattività e di conseguenza elevati costi operativi e di manutenzione. Alcune ricerche hanno mostrato metodi di previsione dell'affidabilità⁹¹ per stimare il tasso di guasto totale di un cambio di turbina eolica offshore. Inoltre, viene utilizzato un caso di studio per dimostrare l'importanza di questa tecnica per confrontare il tasso di guasto dei componenti successivi del cambio delle turbine eoliche onshore e offshore⁹².

Questi meccanismi di guasto influiscono sulla durata di servizio della turbina e possono provocare guasti strutturali catastrofici. Inoltre, è possibile l'inquinamento ambientale derivante dai sottoprodotti della corrosione. Di conseguenza, ogni anno vengono effettuati ingenti investimenti finanziari sia per la prevenzione che per il recupero di questi inconvenienti⁹³. Il tasso di corrosione di un turbine eoliche dipende dalle diverse caratteristiche di attacco che sono influenzate dall'accesso all'ossigeno e all'umidità.

⁹¹ Eom S-H, Kim S-S, Lee J-B. Assessment of Anti-Corrosion Performances of Coating Systems for Corrosion Prevention of Offshore Wind Power Steel Structures. *Coatings*. 2020; 10(10):970. <https://doi.org/10.3390/coatings10100970>.

⁹² Price SJ, Figueira RB. Corrosion Protection Systems and Fatigue Corrosion in Offshore Wind Structures: Current Status and Future Perspectives. *Coatings*. 2017; 7(2):25. <https://doi.org/10.3390/coatings7020025>

⁹³ Maienza, C., Avossa, A. M., Ricciardelli, F., Coiro, D., Troise, G., & Georgakis, C. T. (2020). A life cycle cost model for floating offshore wind farms. *Applied Energy*, 266, 114716.

Anche nel caso dell'energia fotovoltaica, l'affidabilità a lungo termine dei moduli solari è un criterio importante, tuttavia, il degrado dovuto all'ingresso di umidità dei sistemi di pannelli solari affligge ancora la maggior parte degli attuali moduli solari e alla fine si traduce in una perdita di potenza o un guasto completo. Comprendere la fisica del meccanismo di guasto e prevedere la durata del modulo in condizioni operative è fondamentale per sviluppare strategie per mitigare tale degrado. Alcuni ricercatori si sono impegnati per prevedere la durata del funzionamento della struttura di pannelli fotovoltaici sviluppando un modello computazionale dettagliato basato sulla fisica per prevedere la durata del modulo in base a condizioni ambientali e di test in tempo reale e affrontare meccanismi di degradazione associati all'ingresso di umidità, come la corrosione e la delaminazione⁹⁴.

La corrosione attivata dall'umidità e dalla temperatura delle dita metalliche, la delaminazione indotta da sollecitazioni meccaniche e il cedimento dei legami di saldatura sono tra i principali meccanismi di cedimento dei moduli solari. Altre ricerche hanno indagato le implicazioni elettriche della corrosione e della delaminazione, infatti la perdita di efficienza del pannello fotovoltaico dovuta a corrosione, delaminazione comporta una complessa interazione di redistribuzione di tensione/corrente⁹⁵.

Quello che abbiamo avviato grazie alla collaborazione con l'università di Praga è il tentativo di monitorare la corrosione dei materiali, di cui questo istituto, diretto da Thomas Prosek, costituisce la massima competenza a livello europeo. Già all'interno dell'unità operativa di ricerca praghese si sta sperimentando un sensore innovativo

⁹⁴ R. Asadpour, R. V. K. Chavali and M. A. Alam, "Physics-Based computational modeling of moisture ingress in solar modules: Location-specific corrosion and delamination," *2016 IEEE 43rd Photovoltaic Specialists Conference (PVSC)*, Portland, OR, USA, 2016, pp. 0840-0843, doi: 10.1109/PVSC.2016.7749725.

⁹⁵ R. Asadpour, X. Sun and M. A. Alam, "Electrical Signatures of Corrosion and Solder Bond Failure in c-Si Solar Cells and Modules," in *IEEE Journal of Photovoltaics*, vol. 9, no. 3, pp. 759-767, May 2019, doi: 10.1109/JPHOTOV.2019.2896898.

di corrosione basato su variazione di resistenza elettrica. Pur essendo una tecnica innovativa e promettente, attualmente sussistono problematiche di rumore del segnale, risolto solo parzialmente con convenzionali metodi di filtraggio. Abbiamo dunque lavorato alla creazione di un modello deep learning che prende in ingresso una finestra temporale di campioni di resistenze, unitamente a parametri ambientali, e predice in uscita il fattore correttivo da adottare per la formula standard che dalle resistenze inferisce l'indice di corrosione.

4.1 Modello deep learning per il filtraggio del rumore

4.1.1 Collaborazione con UCT di Praga

Qui di seguito presentiamo un primo tentativo di ridurre il rumore nella misura resistometrica della corrosione, utilizzando un modello di deep learning. La prima sezione comprende una breve formalizzazione del modello con riferimento ai suoi input e output, la sezione §3 descrive i passi compiuti per produrre i risultati, e la sezione §4 riassume i risultati.

Come nei modelli precedenti qui di seguito riportiamo alcune righe di codice per una maggiore comprensione del modello:

- Funzione che crea il modello

```

92
93
94 ▼ def create_model():|
95     """
96     create the simple neural model
97
98     return: [keras.engine.training.Model]
99     """
100     x = Input( shape=( cnfg.ns_input, input_size, ) ) # input layer
101     return_sequences = hasattr( cnfg, "double_rnn" ) # uses two recurrent layers
102 ▼ y = layers.LSTM( # recurrent layer
103         input_size, # dimensionality of the output space
104         activation = cnfg.act_rnn, # activation function to use
105         return_sequences = return_sequences, # return the last output or the full sequence
106         return_state = False, # return the last state in addition to the output
107         stateful = False, # use last state as initial state of next batch
108         unroll = False, # unroll the network
109         name = 'rcrr_1'
110     )( x )
111 ▼ if hasattr( cnfg, "double_rnn" ): # uses two recurrent layers
112 ▼ y = layers.LSTM( # second recurrent layer
113         input_size,
114         activation = cnfg.act_rnn,
115         return_sequences = False,
116         return_state = False,
117         stateful = False,
118         unroll = False,
119         name = 'rcrr_2'
120     )( y )
121 ▼ y = layers.Dense( # first dense layer
122     cnfg.dense_1,
123     activation = cnfg.act_dn1,
124     name = "dense_1"
125 )( y )
126 ▼ y = layers.Dense( # second dense layer
127     1,
128     activation = cnfg.act_dn2,
129     name = "dense_2"
130 )( y )
131
132     return Model( inputs=x, outputs=y, name="nn_model" ) # the complete model
133
134

```

- Funzione che addestra il modello

```
156
157 def train_model( model, gen, val=None ):
158     """
159     train the model
160
161     model:         [keras.engine.training.Model]
162     gen:           [keras.preprocessing.sequence.TimeseriesGenerator] training generator
163     val:           [keras.preprocessing.sequence.TimeseriesGenerator] validation generator, may be missing
164
165     return:        [keras.callbacks.History]
166     """
167     loss           = loss_code[ cfg.loss_code ]
168     optimizer      = optimiz_code[ cfg.optimiz_code ]( lr=cfg.lr )
169
170     model.compile(
171         optimizer  = optimizer,
172         loss       = loss
173     )
174
175     history = model.fit_generator(
176         gen,                               # the training generator
177         validation_data = val,             # the validation generator, or None
178         epochs         = cfg.n_epochs     # number of training epochs
179     )
180
181     return history                          # a set of information about training
182
183
```

- Funzione che aggrega i dati temporali in modo fruibile dalla rete in addestramento:

```

145
146 def build_generators( h5_files, ns_target, ns_input, batch_size ):
147     """
148     Build the generators producing batches of data for training and validation, using specialized
149     keras generator for time series
150
151     h5_files:      [list] list of base name of the HDF5 files, the last one is for validation
152     batch_size:   [int] batch size
153     ns_target:    [int] number of temporal samples to be used for the target average
154     ns_input:     [int] number of temporal samples to be used for the input vectors
155
156     return:       [tuple] train generator, validation generator
157     """
158
159     all_data = read_dataset( h5_files[ -1 ] ) # the last file is used for validation
160     thick = avg_thick( all_data[ "thick" ], ns_target, ns_input )
161     targets = np.expand_dims( thick, axis=-1 ) # target value is the thickness given by the sensor
162     # get as input the physical data useful for prediction
163     inputs = np.array( [ all_data[ "r_ref" ], all_data[ "r_meas" ], all_data[ "temp" ] ] )
164     data = inputs.transpose() # arrange data for Tensorflow
165
166     val_gen = TimeseriesGenerator(
167         data, # input
168         targets, # target
169         length = ns_input, # number of temporal samples in input
170         batch_size = batch_size, # size of batches
171         stride = 1, # stride in time steps between samples
172         sampling_rate = 1, # temporal sampling rate
173         start_index = 0, # time of first sample
174         shuffle = False # do not shuffle the samples
175     )
176
177     for f in fields:
178         all_data[ f ] = np.empty( ( 0, ) )
179     for h5_file in h5_files[ :-1 ]: # now build data for all files except the last one
180         data = read_dataset( h5_file )
181         for f in fields: # read all fields available in the dataset
182             all_data[ f ] = np.append( all_data[ f ], data[ f ] )
183
184     thick = avg_thick( all_data[ "thick" ], ns_target, ns_input )
185     # target value is the thickness given by the sensor
186     targets = np.expand_dims( thick, axis=-1 )
187     # get as input the physical data useful for prediction
188     inputs = np.array( [ all_data[ "r_ref" ], all_data[ "r_meas" ], all_data[ "temp" ] ] )
189     data = inputs.transpose() # arrange data for Tensorflow
190
191     train_gen = TimeseriesGenerator( # same arguments as in val_gen = TimeseriesGenerator()
192         data,
193         targets,
194         length = ns_input,
195         batch_size = batch_size,
196         stride = 1,
197         sampling_rate = 1,
198         start_index = 0,
199         shuffle = True # do shuffle the samples
200     )
201
202     return train_gen, val_gen
203

```

4.1.2 Descrizione del modello

Ho utilizzato il più possibile le stesse notazioni fornite da University of Chemistry and Technology Prague sul monitoraggio della corrosione in tempo reale come strumento per comprendere la corrosione atmosferica:

$R_{ref,x}$	masked resistance ad time x
$R_{meas,x}$	measuring resistance ad time x
Δh_x	thickness loss at time x , as computed with UCTP formula
T_x	temperature at time x

Il modello DL è rappresentato dalla seguente funzione:

$$y_x = f_{\theta} \left(\vec{r}_{ref,x}, \vec{r}_{meas,x}, \vec{T}_x \right)$$

dove θ è l'insieme dei parametri del modello e i vettori di input sono:

$$\vec{r}_{ref,x} = \begin{bmatrix} R_{ref,x+M} \\ \dots \\ R_{ref,x} \\ \dots \\ R_{ref,x-M} \end{bmatrix} \quad \vec{r}_{meas,x} = \begin{bmatrix} R_{meas,x+M} \\ \dots \\ R_{meas,x} \\ \dots \\ R_{meas,x-M} \end{bmatrix} \quad \vec{T}_x = \begin{bmatrix} T_{x+M} \\ \dots \\ T_x \\ \dots \\ T_{x-M} \end{bmatrix}$$

L'output del modello DL dovrebbe essere il fattore di correzione da applicare alla perdita di spessore stimata al tempo x , supponendo che sia derivato facendo la media su $2N + 1$ campioni intorno a x . La perdita di spessore corretta $\widetilde{\Delta h}_x$ sarà:

$$\widetilde{\Delta h}_x = \frac{1}{2N + 1} \sum_{x-N \leq i \leq x+N} \Delta h_i + y_x$$

I parametri θ del modello saranno appresi mediante supervisione, utilizzando la seguente funzione di perdita $L(x)$ del campione x :

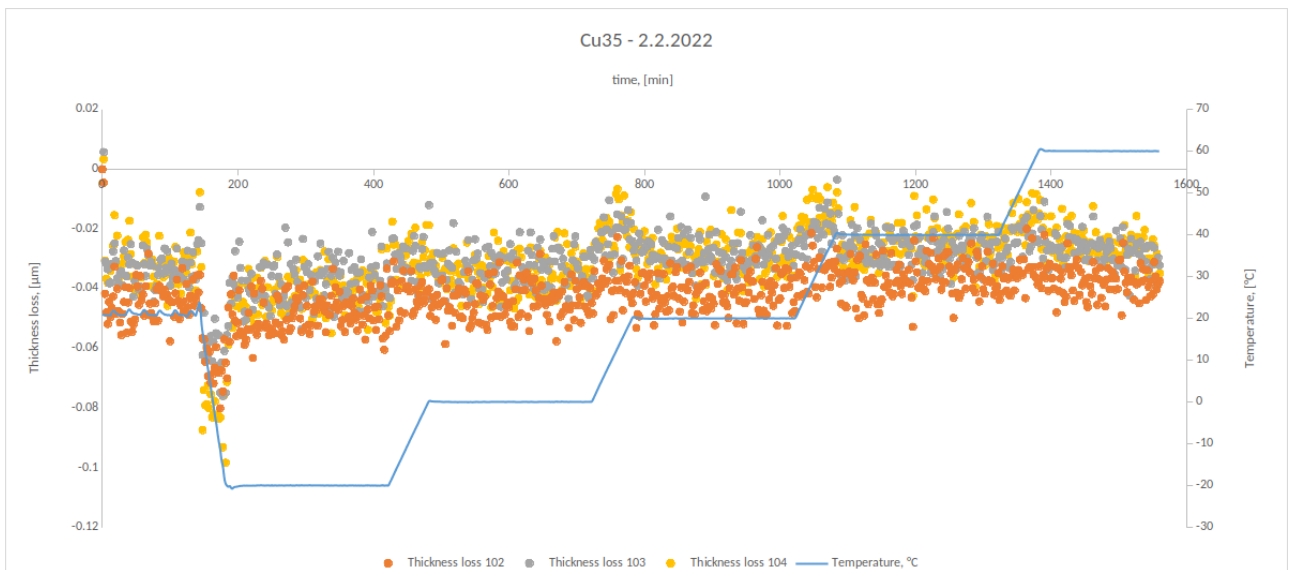
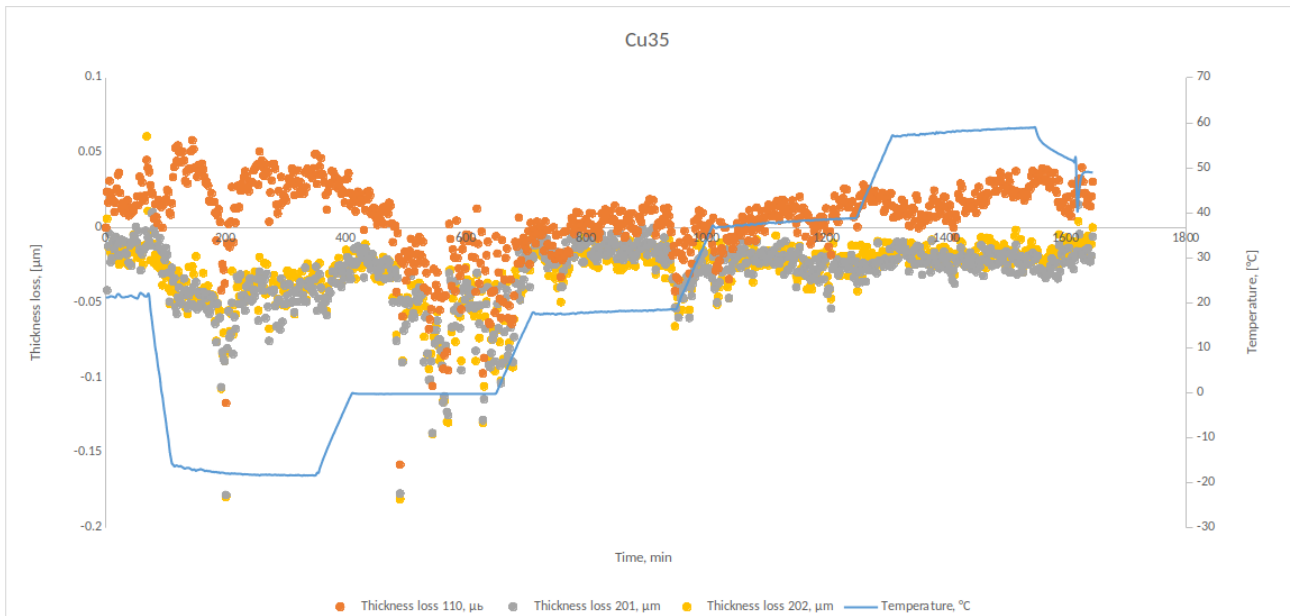
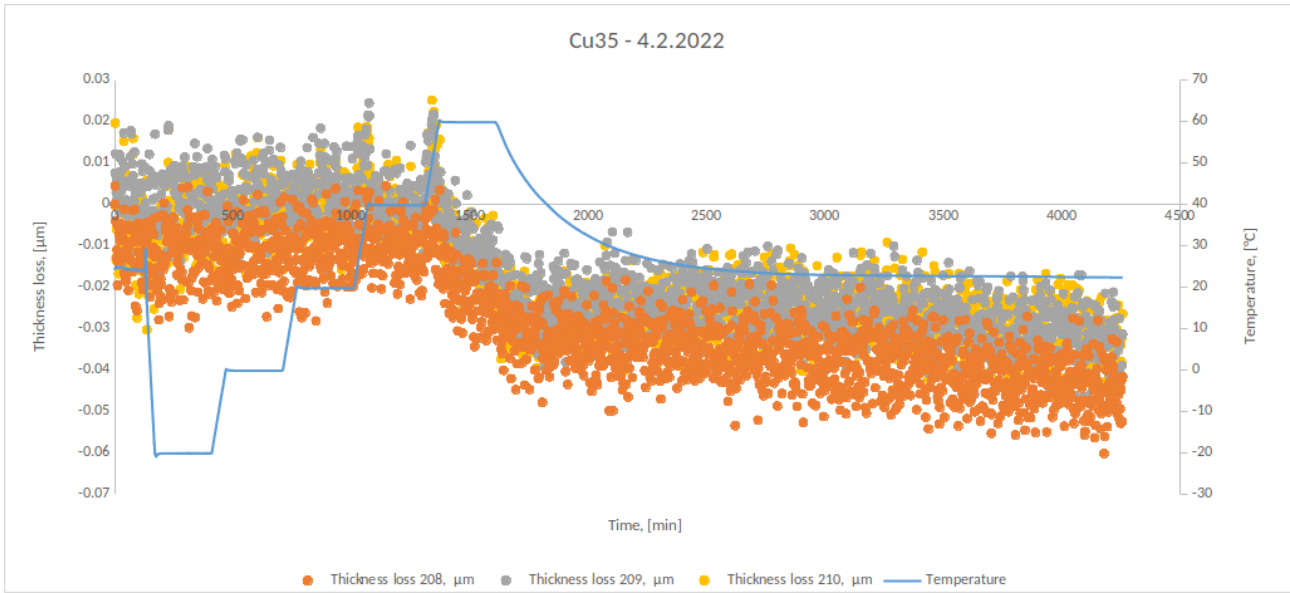
$$\mathcal{L}(x) = \mathcal{E} \left(y_x, -\frac{1}{2N+1} \sum_{x-N \leq i \leq x+N} \Delta h_i \right)$$

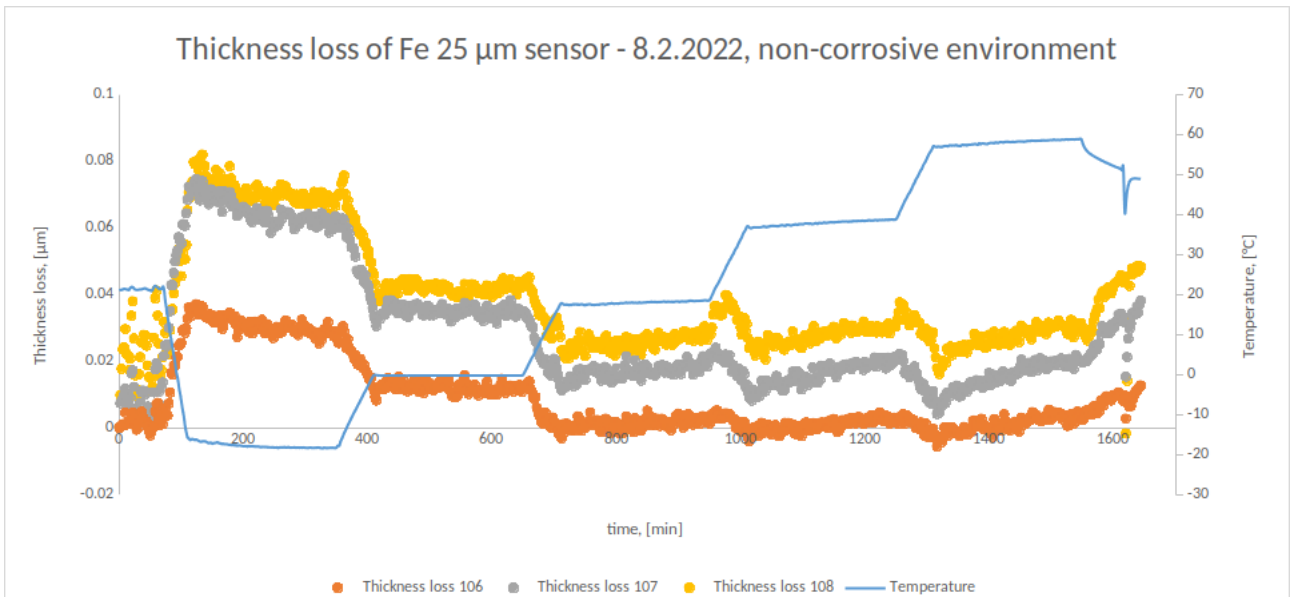
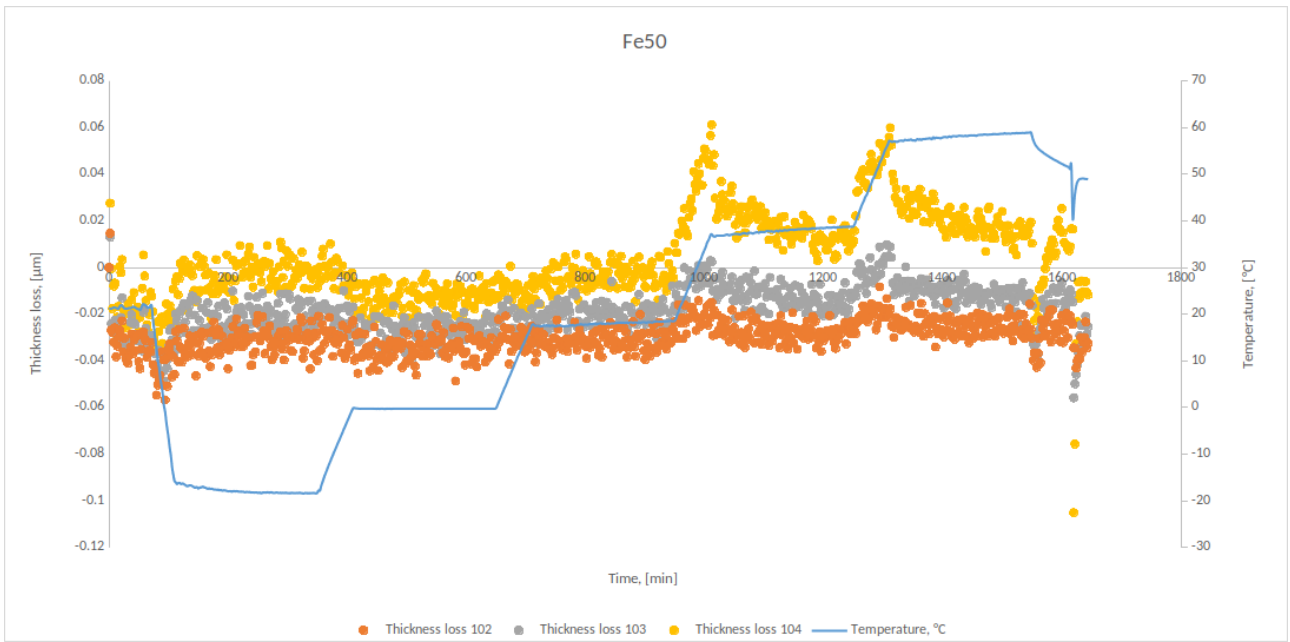
Dove \mathcal{E} è una funzione di misura dell'errore, come l'errore quadrato.

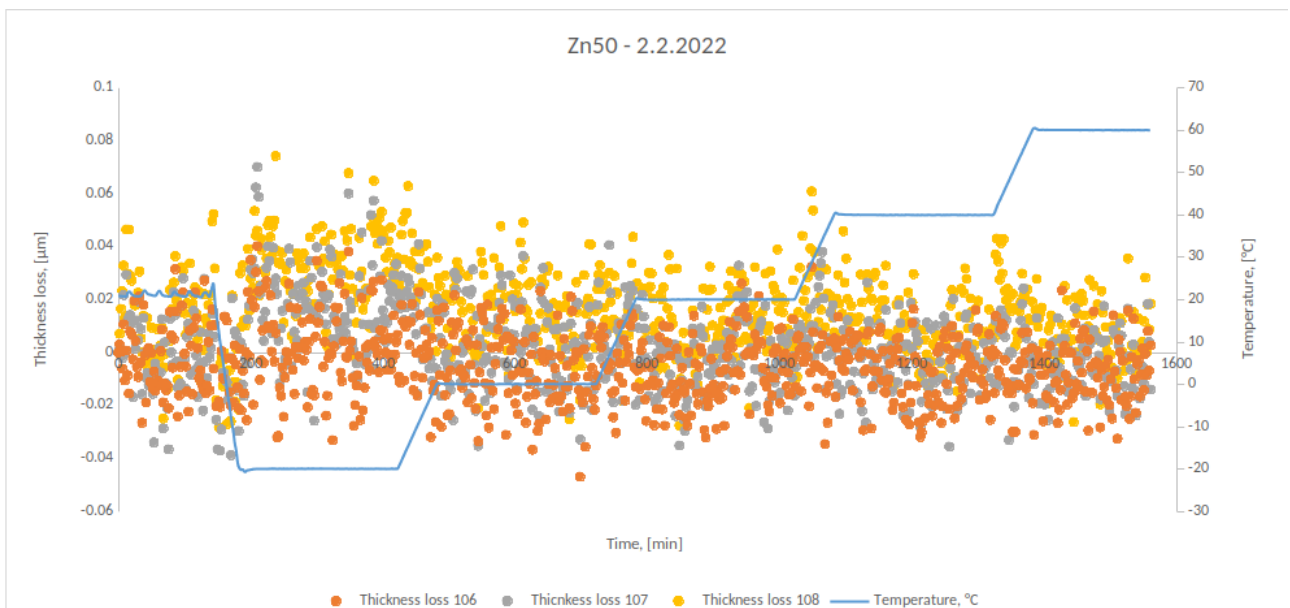
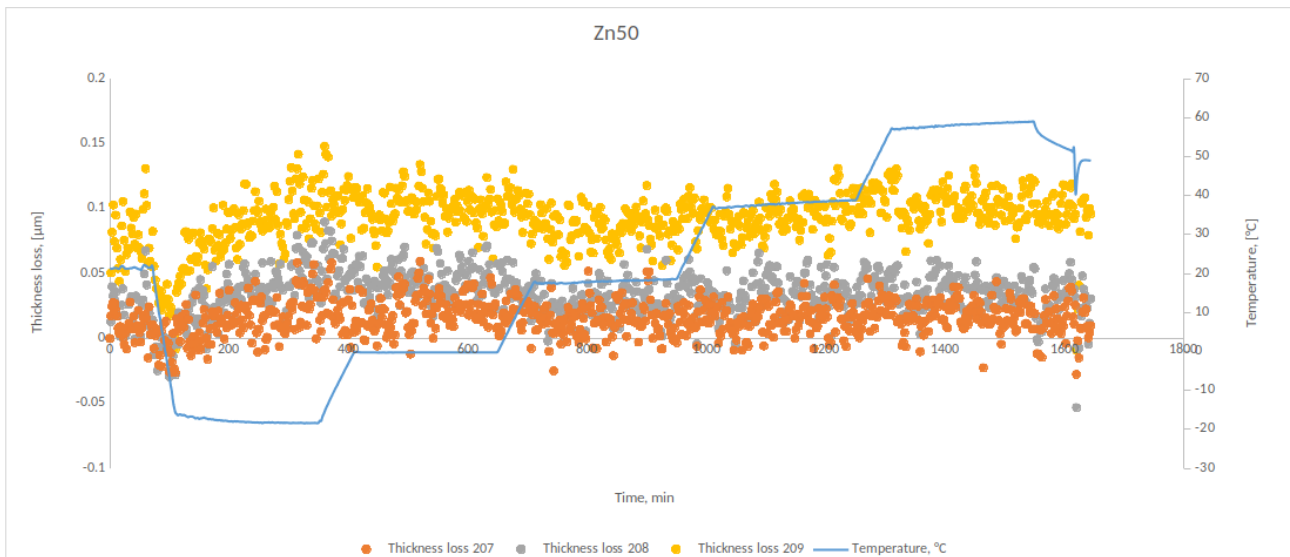
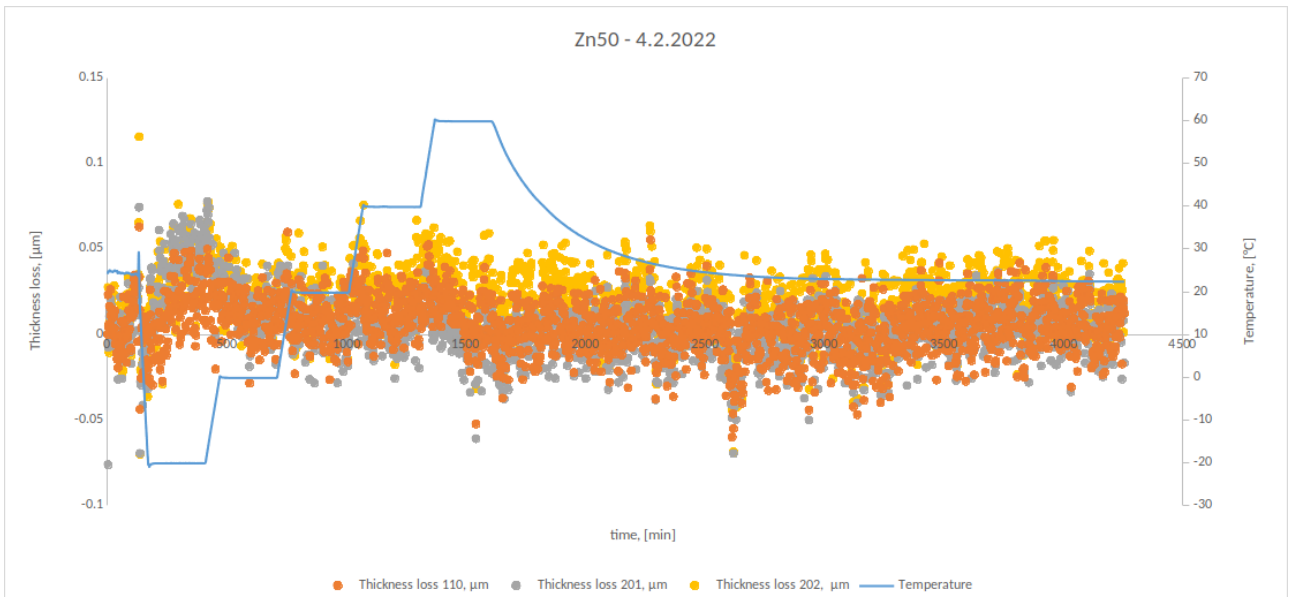
Sto usando due finestre temporali distinte M e N dove la prima è la finestra su cui il modello può stimare al meglio la correzione, e quest'ultima è la media temporale desiderata per la stima della perdita di spessore. Nel caso in cui la perdita di spessore è stimata su 5 campioni temporali, sarà $N = 2$. Potrebbe essere che alla fine $M = N$ sarà una scelta ragionevole, ma usando due finestre distinte c'è più spazio per gli esperimenti.

Indipendentemente dall'ampiezza della finestra temporale, va sottolineato che anche questa problematica condivide la caratteristica saliente già vista per gli impianti fotovoltaici, di tentare una predizione basandosi su segnali memorizzati nel tempo. Ancora una volta, non è dato sapere quali configurazioni di questi segnali potranno risultare importanti da fissare in memoria oppure no, indipendentemente da quanto siano remote o recenti, rispetto all'istante temporale di predizione. Valgono pertanto le stesse considerazioni espresse a p. 71, sul seguire la via cognitiva che suggerisce la LSTM come architettura neurale più promettente per questa applicazione. In questo caso, vista la densità temporale dei campioni, e la loro impercettibile correlazione con la misura di interesse, si è proceduto direttamente ad un raddoppio di strati LSTM in stack, dove un primo impiega come input le variabili dette sopra, e un secondo che prende in input l'output del primo LSTM, ovvero tracce mnemoniche già processate dei dati misurati.

Nelle figure a seguire si sono riportati in grafico i dati originali degli esperimenti condotti dal gruppo praghese, in cui sono sovrapposti i punti di misura di tre sensori sullo stesso supporto metallico, sensori individuati dal numero del canale dello strumento di acquisizione. Vi sono tre serie diverse, per tre substrati metallici: il rame (Cu) da 35 micron, il ferro (Fe) da 50 e 25 micron, e lo zinco (Zn) da 50 micron. Durante gli esperimenti il parametro che viene variato è la temperatura, in un intervallo molto ampio, tra -20 e 60 gradi. È evidente in tutti i grafici come vi sia una notevole variabilità delle misure di tipo assolutamente casuale, con sovrapposta una debole dipendenza dalla temperatura. Spesso la variabilità risulta diversa tra le tre repliche dello stesso sensore, rendendo quindi ancor più problematica la correzione. Si evince inoltre come lo zinco sia il substrato con errore di stima dello spessore meno legato alla temperatura; infatti, la dispersione dei punti di misura è pressoché uguale durante tutta l'evoluzione temporale. Per il rame esiste un effetto visibile della temperatura, che però si manifesta in modi irregolari nei tre esperimenti, e anche per le tre repliche del sensore in uno stesso esperimento:







4.1.3. Descrizione dei passaggi

Il modello è stato sviluppato e sperimentato sui dati acquisiti da University of Chemistry and Technology Prague in ambiente controllato non corrosivo il 2, 4 e 8 febbraio 2022 e è stato fornito nei seguenti file in formato excel:

```
Cu35 Zn50 Fe50 Fe25 4x1 4.2.2022.xlsx  
Cu35 Zn50 Fe50 Fe25 4x1 8.2.2022.xlsx  
Cu35 Zn50 Fe50 Fe50 4x1 2.2.2022.xlsx
```

La procedura comprende i passaggi descritti di seguito

4.1.4 Da excel a file CSV

Per preparare i *datasets* adatti al modello *deep*, in questo primo passaggio sono stati estratti manualmente dai file excel singoli fogli, ciascuno per un singolo tipo di sensore –, ottenendo i seguenti 8 file:

```
Cu35 2022 02 02.csv  
Cu35 2022 02 04.csv  
Cu35 2022 02 08.csv  
Fe25 2022 02 08.csv  
Fe50 2022 02 08.csv  
Zn50 2022 02 02.csv  
Zn50 2022 02 04.csv  
Zn50 2022 02 08.csv
```

4.1.5 Da file CSV a file HDF5

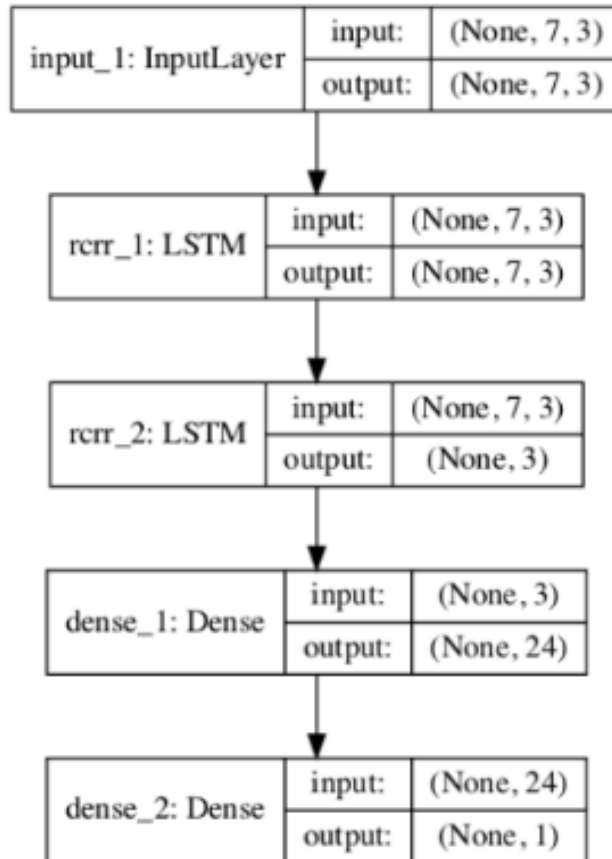
Per ogni combinazione R_{ref}/R_{meas} disponibile nei file CSV è stato generato un unico file HDF5. Ci sono dentro generale tre diversi canali di misura per ogni resistenza di riferimento. I file HDF5 contengono quanto segue campi:

variable	HDF5 field key
R_{ref}	r_ref
R_{meas}	r_meas
Δh	thick
T_x	temp

La generazione del file HDF5 avviene con il programma `get_csv.py` e non è completamente automatizzata, il nome del file sorgente CSV, così come i nomi dei suoi campi, devono essere specificati all'interno di questo programma. Questo programma esegue anche una normalizzazione, in modo da avere tutti gli ingressi all'incirca nell'intervallo $[-1 \dots 1]$ e l'uscita nell'intervallo $[0 \dots 1]$. Sono stati generati in totale 22 file HDF5, con nomi che seguono questa sintassi:

`<CSV_source>_<channel>.h5`

per esempio `Fe25_2022_02_08_107.h5` è il file HDF5 derivato dal file CSV `Fe25_2022_02_08.csv` dove R_{meas} è il canale di misura 107 e Δh è la perdita di spessore calcolata rispetto a questo canale.



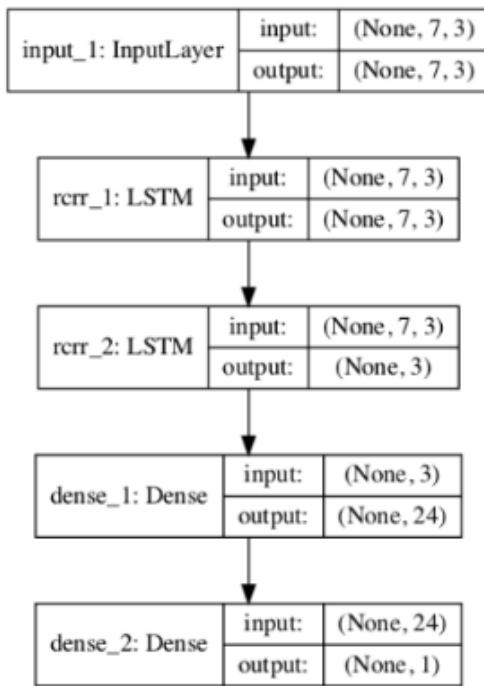
Arch. 1

Figura 1: Architettura del modello deep (Arch1). La forma del tensore di input è 7×3 perché la finestra temporale del campione di input è $M = 7$, e per ogni punto abbiamo tre dati: $Rref$, $Rmeas$, T . La forma dell'output è solo 1 poiché la previsione è lo scalare Δh mediato su $N = 5$ passi temporali.

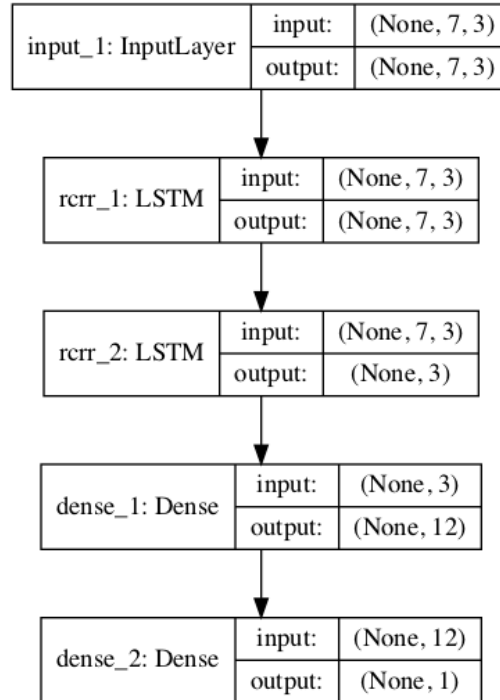
M	ns_target	number of temporal samples for the target
N	ns_input	number of temporal samples for the inputs
D	dense_1	number of neurons in the first dense
\mathcal{L}	loss_code	code for keras losses
E	n_epochs	number of epochs
B	batch_size	size of batch
η	lr_rate	learning rate

training	M	N	D	\mathcal{L}	E	B	η	improvement
22.05_26	5	7	24	MAP	2000	64	$2E^{-5}$	0.3474
22.05_31	5	7	24	MAP	2000	64	$2E^{-5}$	0.3482
22.06_07	5	7	24	MAP	200	64	$2E^{-5}$	0.3364
22.06_09	5	7	24	MAP	200	64	$1E^{-5}$	0.3330
22.06_24	5	7	24	MAP	200	64	$4E^{-5}$	0.3344
22.07_07	5	7	24	MAP	200	32	$2E^{-5}$	0.3390
22.07_12	5	7	24	MAP	200	32	$2E^{-5}$	0.3389
22.08_11	5	7	24	MAP	200	128	$2E^{-5}$	0.3313
22.08_12	5	7	24	MAP	200	16	$2E^{-5}$	0.3411
22.08_18	5	7	12	MAP	200	16	$2E^{-5}$	0.3473
22.08_20	5	7	12	MAP	200	64	$2E^{-5}$	0.3387
22.08_22	5	7	48	MAP	200	64	$2E^{-5}$	0.3420
22.08_27	5	11	12	MAP	200	16	$2E^{-4}$	0.3370
22.08_29	5	7	12	MAP	200	16	$2E^{-4}$	0.3474

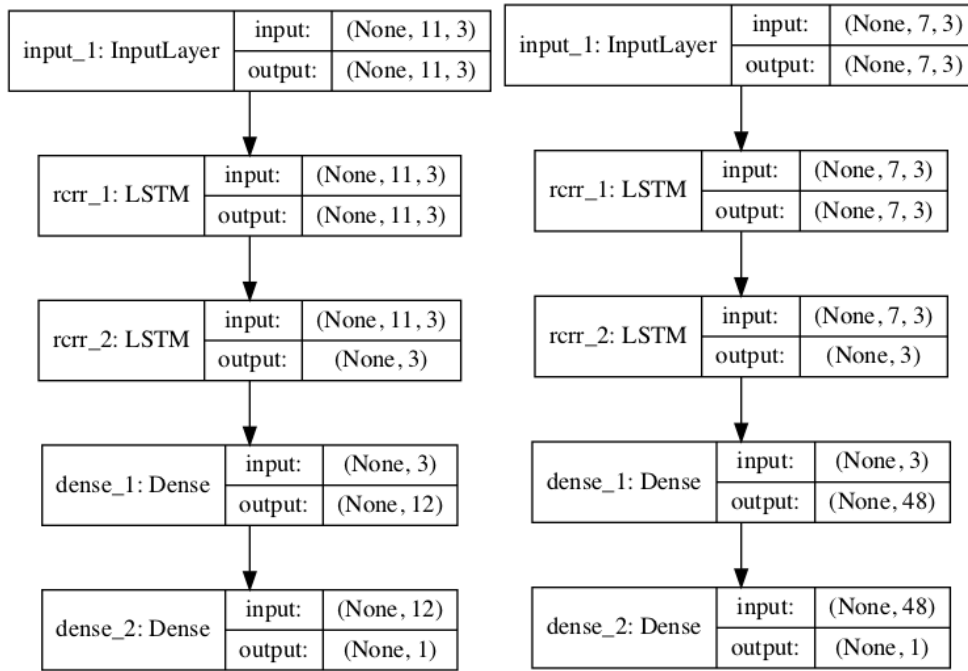
Qui di seguito riportiamo le quattro diverse architetture del modello deep:



Arch. 1



Arch2



Arch. 3

Arch. 4

4.1.6 Addestramento del modello deep ed *evaluation*

Per ogni combinazione R_{ref}/R_{meas} – pertanto, per ogni file HDF5, viene addestrato e valutato un modello di deep learning. La generazione dei campioni di dati di addestramento e convalida dal file HDF5 viene eseguita con il modulo

`dati gen.py`.

Si noti che a causa dello scopo preliminare di questa indagine i campioni disponibili non sono stati suddivisi in insiemi disgiunti di addestramento e validation. Tutti i campioni sono stati utilizzati per entrambi gli scopi, con la sola differenza che nel primo caso i campioni sono stati mescolati, mentre per la valutazione è stata conservata la loro sequenza nel tempo.

L'addestramento e la valutazione vengono eseguiti con il programma `model.py` i cui parametri principali vengono letti da un file di configurazione. Vengono utilizzati 22 file di configurazione, denominati `cfg <XX> .py` dove XX è un numero compreso tra 00 e 21. I parametri architettonici del modello sono identici per tutte le configurazioni, l'unica differenza è la specificazione dei file di dati HDF5.

È stato effettuato un rapido screening preliminare dei parametri architettonici solo per il dataset `Zn50_2022_02_04_110`, ed è stato poi adottato per tutti gli altri dati.

Le finestre temporali per input e output sono:

$$M = 7$$

$$N = 5$$

Il modello, come mostrato in figura in Fig. 1 è costituito da due layers LSTM (*Long-Short Term Memory*), seguiti da due dense layers.

4.1.7 Risultati

I risultati del modello sono prodotti dallo stesso programma di `model.py` usando il modello addestrato. Le 22 esecuzioni vengono serializzate mediante lo script `train.all`, e ogni esecuzione del programma crea una cartella di output individuale con il nome che contiene la data e l'ora dell'addestramento. La Thickness loss corretta dal modello viene confrontata con quella originaria, mediata nelle finestre temporali di 5 campioni. Il confronto è rappresentato nei grafici denominati:

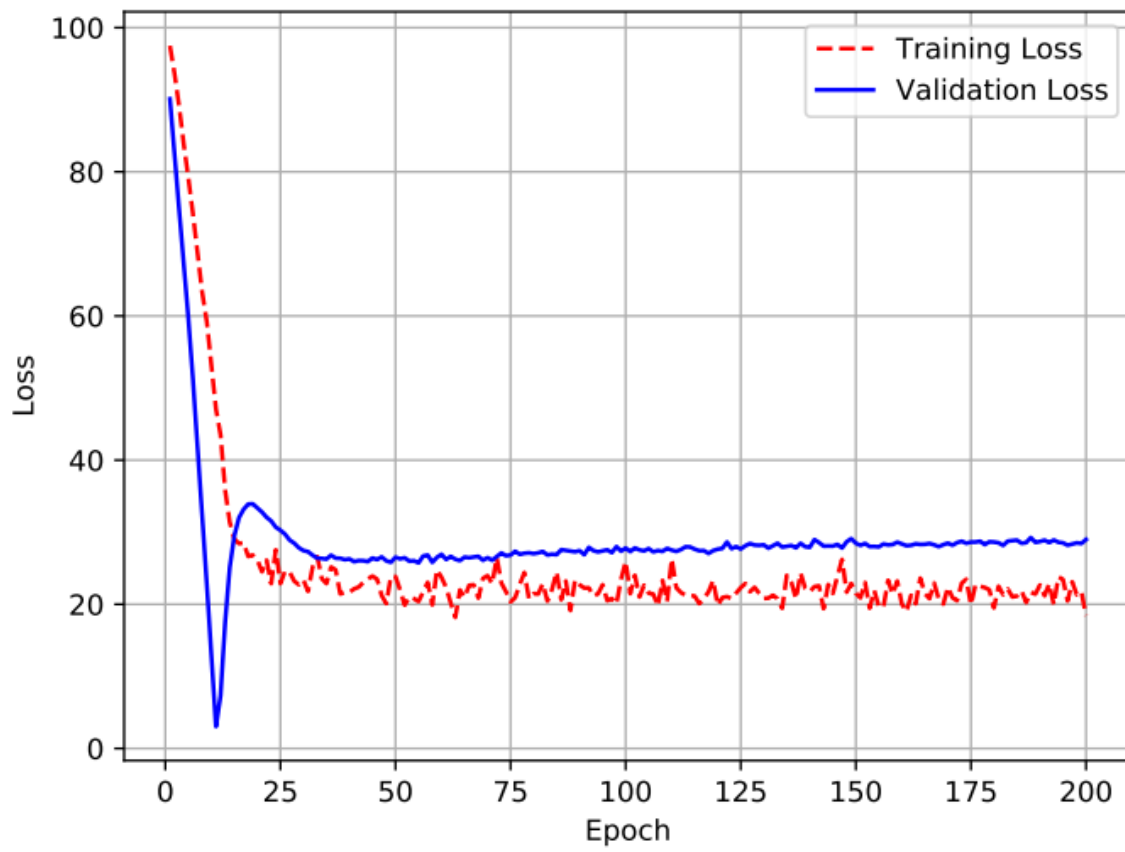
`<HDF5_source>_corr.pdf`

per esempio, `Cu35_2022_02_02_102_corr.pdf` è il grafico della correzione eseguita dal modello addestrato e valutato sul dataset `Cu35_2022_02_02_102.h5`.

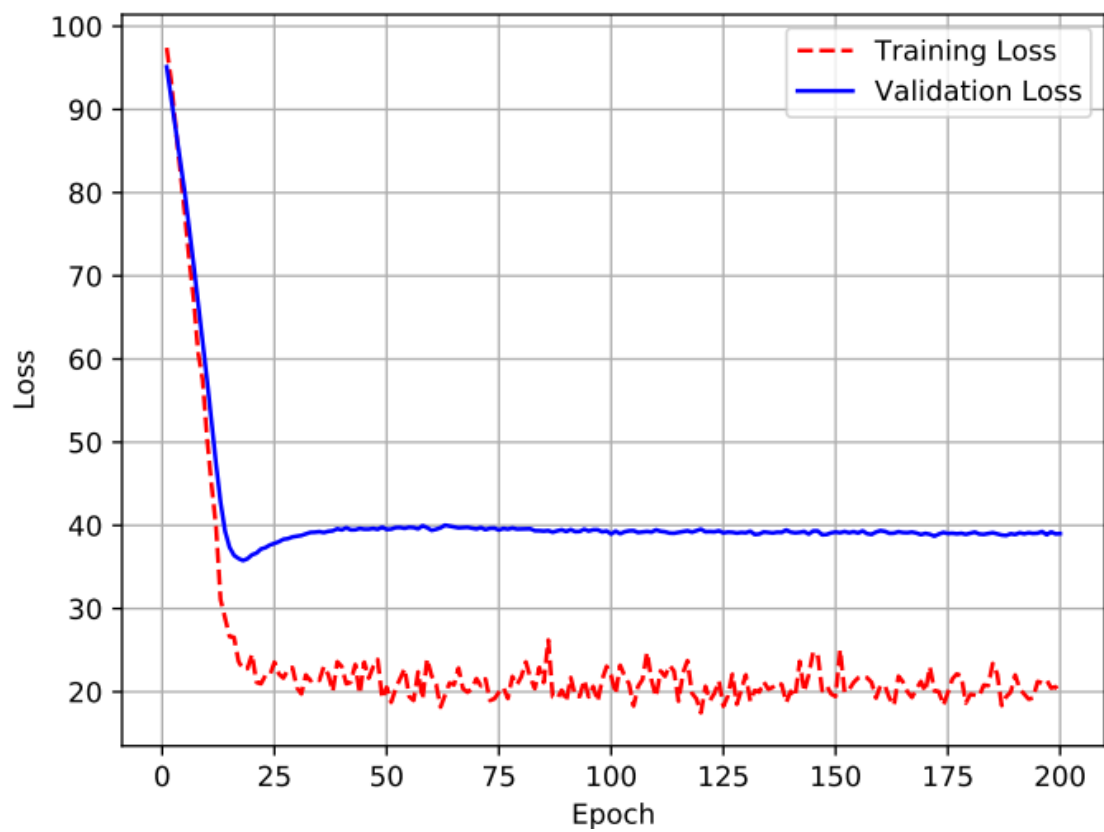
Per ogni dataset viene calcolata una valutazione complessiva. Poiché thickness loss nota è sempre 0, l'errore è solo il valore assoluto per ciascun campione, sia per la perdita di spessore originale mediata nelle finestre temporali dei 5 campioni, sia per la previsione del modello. La media e la deviazione standard degli errori assoluti $|\Delta h|$ (originale) e $|\Delta \hat{h}|$ (modello corretto) vengono confrontati per ciascun dataset nella tabella seguente:

dataset	$ \Delta h $		$ \widetilde{\Delta h} $	
	mean	std	mean	std
Cu35_2022.02.02.102	0.042	0.006	0.003	0.003
Cu35_2022.02.02.103	0.031	0.007	0.004	0.004
Cu35_2022.02.02.104	0.031	0.011	0.004	0.007
Cu35_2022.02.04.208	0.028	0.013	0.005	0.006
Cu35_2022.02.04.209	0.018	0.011	0.007	0.008
Cu35_2022.02.04.210	0.019	0.011	0.007	0.008
Cu35_2022.02.08.110	0.019	0.014	0.016	0.013
Fe25_2022.02.08.106	0.009	0.011	0.002	0.001
Fe25_2022.02.08.107	0.029	0.019	0.003	0.004
Fe25_2022.02.08.108	0.038	0.017	0.004	0.004
Fe50_2022.02.08.102	0.030	0.005	0.002	0.002
Fe50_2022.02.08.103	0.019	0.008	0.003	0.004
Fe50_2022.02.08.104	0.014	0.011	0.006	0.007
Zn50_2022.02.02.106	0.007	0.005	0.006	0.005
Zn50_2022.02.02.107	0.008	0.007	0.007	0.006
Zn50_2022.02.02.108	0.017	0.011	0.009	0.007
Zn50_2022.02.04.110	0.010	0.008	0.008	0.007
Zn50_2022.02.04.201	0.010	0.011	0.008	0.007
Zn50_2022.02.04.202	0.021	0.012	0.010	0.008
Zn50_2022.02.08.207	0.017	0.008	0.007	0.006
Zn50_2022.02.08.208	0.035	0.012	0.009	0.009
Zn50_2022.02.08.209	0.092	0.017	0.012	0.012

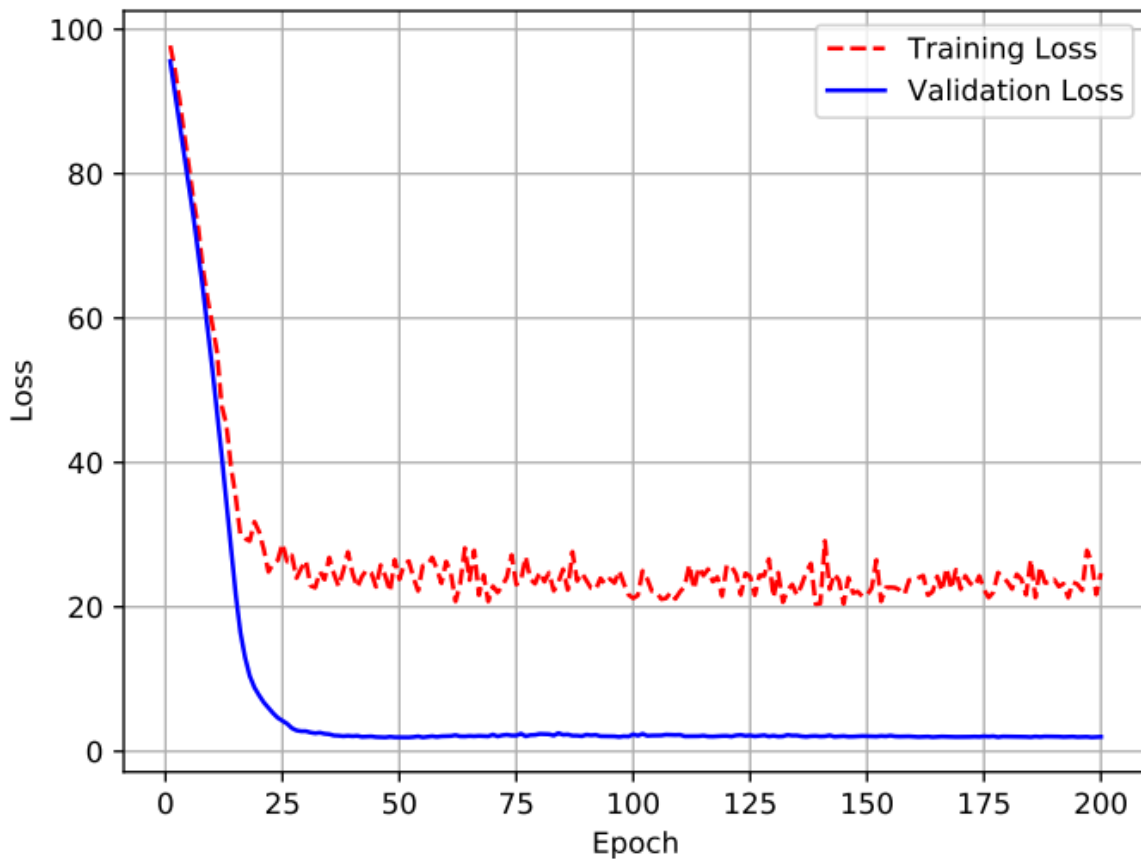
Qui di seguito riporteremo tre dei grafici scelti che rappresentano alcuni andamenti della funzione di loss alle varie epoche di addestramento, in particolare Cu35_2022_02_02_102_loss, Cu35_2022_02_08_110_loss, Cu35_2022_02_04_210_loss. Va evidenziata l'anomalia di quest'ultimo, in cui l'errore sul validation set dopo una ventina di epoche diventa addirittura minore rispetto al training set. La risposta si trova confrontando il grafico degli esperimenti originali (Cu35_2022_02_04 a p. 110) in cui si trova la misura del canale 210 del sensore, si tratta di uno dei casi più favorevoli, in cui la temperatura risulta un buon predittore dell'errore. Essendo questa serie quella presa per la validazione, il modello la corregge bene pur se i suoi dati sono sconosciuti, a comprova che l'addestramento tende a fare tesoro dei pochi indizi presenti nei dati che meglio correggono le risposte dei sensori. Negli altri due casi invece la misura presa come validazione risulta negli esperimenti originali più erratica (Cu35_2022_02_02 e Cu35_2022_02_08 a p. 109-111), e quindi anzitutto la loss nel validation set si mantiene peggiore rispetto al training set, ma soprattutto mostra un picco verso il basso, marcatamente in Cu35_2022_02_02_102_loss, che corrisponde ad una fase in cui il modello non ha ancora assunto la funzione di correzione ottimale per l'intero set, ma una che casualmente trovava corrispondenza in quell'unica misura erratica, sconosciuta al modello.



Cu35_2022_02_02_102_loss



Cu35_2022_02_08_110_loss



Cu35_2022_02_04_210_loss

Come si evince, il modello provvede mediamente ad un miglioramento intorno al 30% sulla precisione del sensore, rispetto al metodo di filtraggio matematico attualmente adottato. Va sottolineato l'unico caso di miglioramento negativo sui nove esperimenti, in cui la correzione va addirittura a peggiorare l'accuratezza, il Cu35_2022_02_08_110.

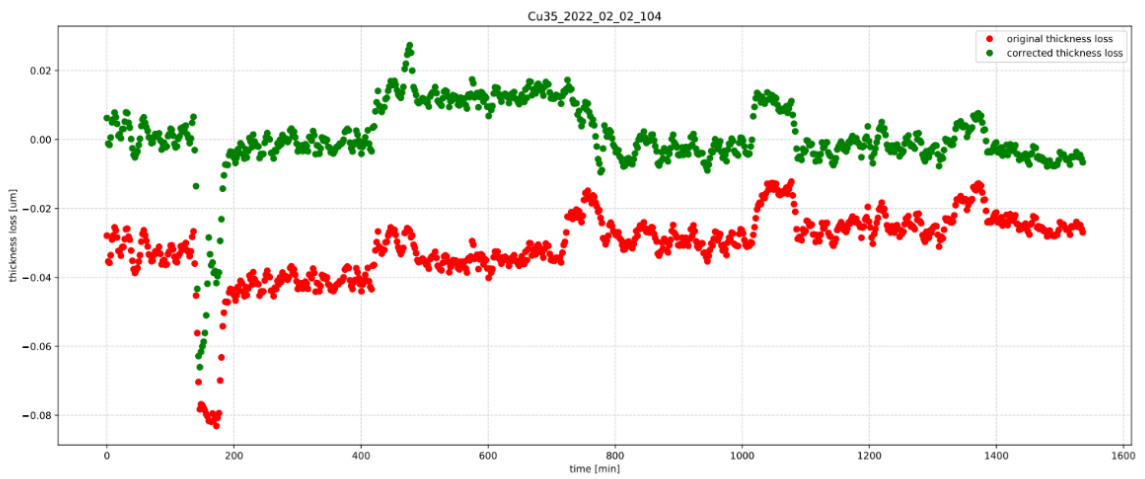
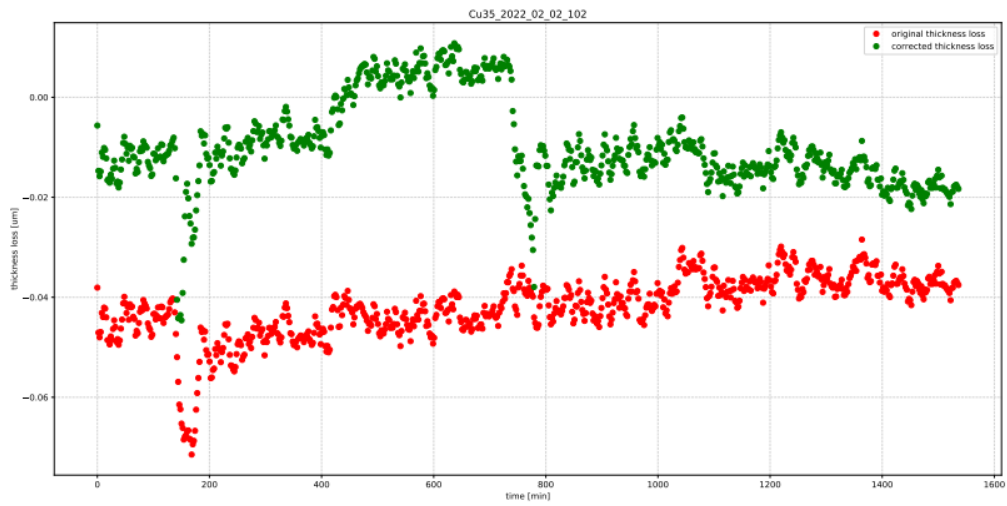
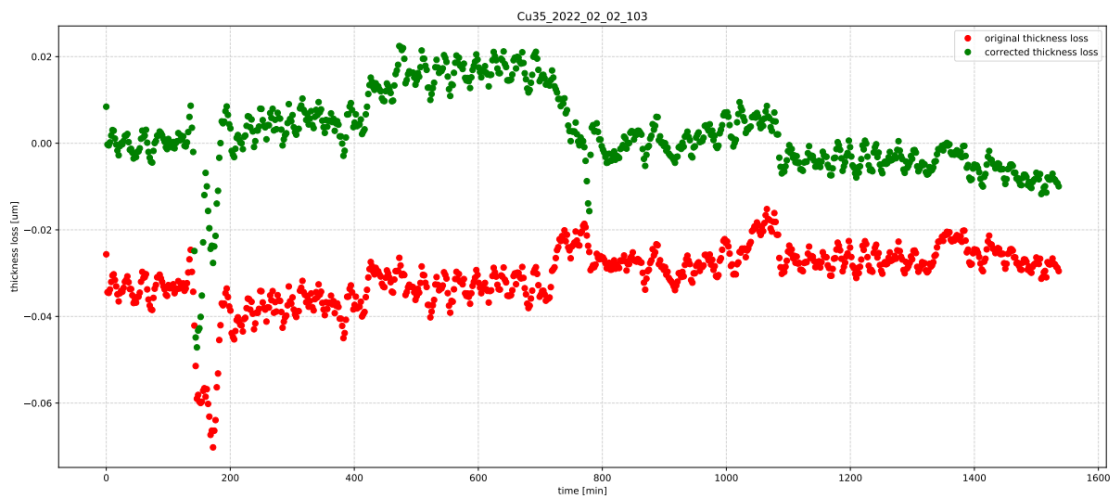
Il risultato diventa perfettamente coerente analizzando l'esperimento originale (Cu35_2022_02_08 a p. 111), in cui questo sensore (canale di misura 110, in colore arancione) mostra un andamento molto diverso dagli altri due sensori, e con una correlazione anomala rispetto alla temperatura. Il sensore fornisce un errore positivo a temperature basse, fino a 0 gradi, poi un errore negativo fino ai 15 gradi, infine un

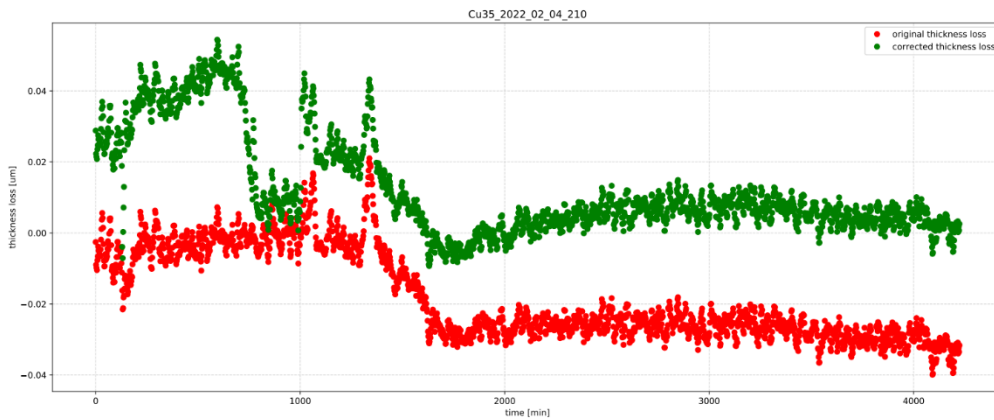
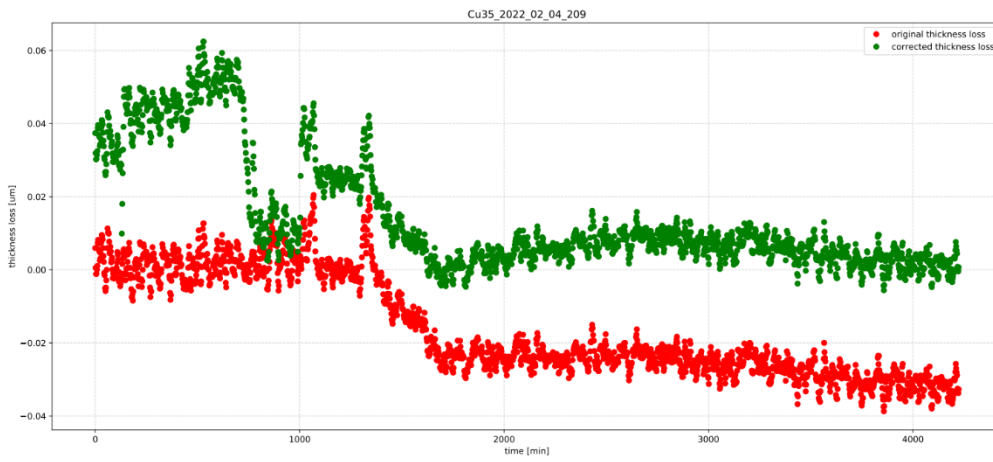
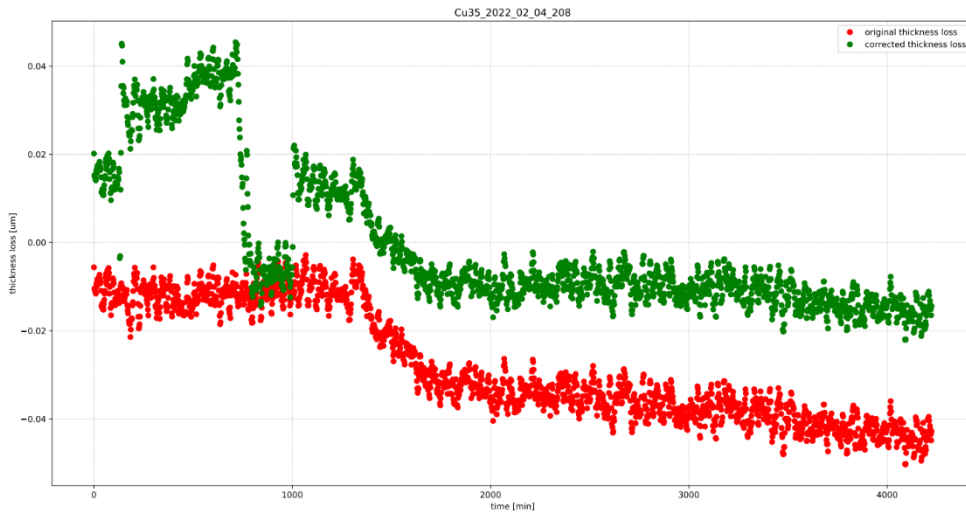
errore nuovamente positivo sopra ai 25 gradi. I casi in cui il modello fornisce un miglioramento maggiore, intorno al 60% (Cu35_2022_02_02_102, Cu35_2022_02_02_103, Cu35_2022_02_02_104) Cu35_2022_02_08_201), riguardano risposte dei sensori con una minima regolarità rispetto all'andamento della temperatura, come si vede dall'esperimento originale Cu35_2022_02_02 (a p. 109).

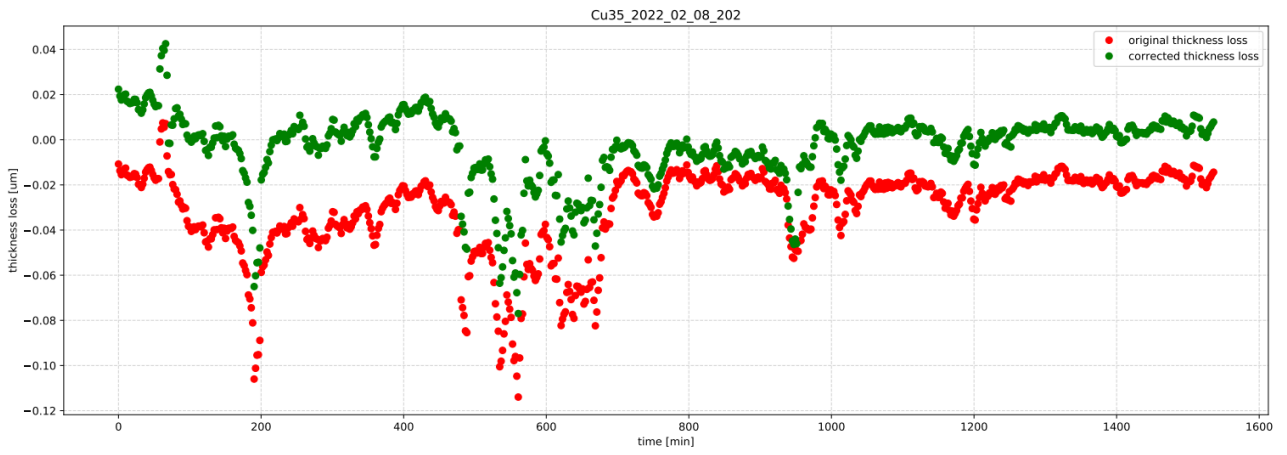
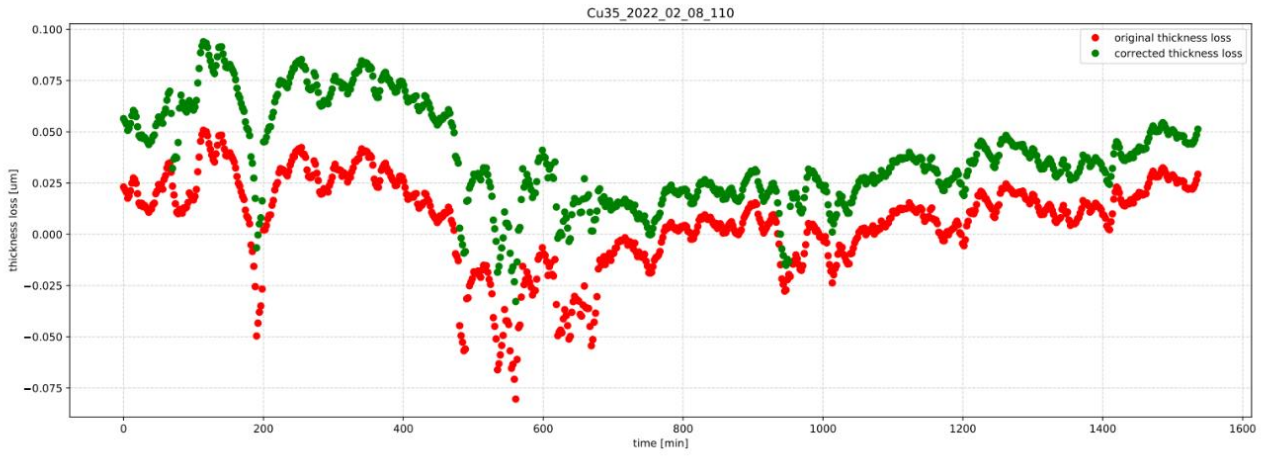
Queste considerazioni sono confermate dall'andamento puntuale delle correzioni apportate dal modello nel tempo, riportate nei grafici delle pagine seguenti.

dataset	original err		corrected err		improvement
	mean	std	mean	std	
Cu35_2022_02_02_102	0.042	0.006	0.012	0.006	0.5632
Cu35_2022_02_02_103	0.031	0.007	0.007	0.007	0.6320
Cu35_2022_02_02_104	0.031	0.011	0.007	0.008	0.6462
Cu35_2022_02_04_208	0.028	0.013	0.014	0.009	0.3363
Cu35_2022_02_04_209	0.018	0.011	0.015	0.016	0.0922
Cu35_2022_02_04_210	0.019	0.011	0.013	0.014	0.1817
Cu35_2022_02_08_110	0.019	0.014	0.039	0.023	-0.3510
Cu35_2022_02_08_201	0.032	0.020	0.010	0.012	0.5282
Cu35_2022_02_08_202	0.031	0.019	0.010	0.012	0.4975

			mean improvement		0.3474







Conclusione

Il periodo di svolgimento di questo dottorato, dal 2019 ad oggi, ha coinciso con una rivoluzione profonda nel mondo del deep learning, e di conseguenza dell'Intelligenza Artificiale, con la rapida diffusione dei modelli del linguaggio, a seguito dell'invenzione dell'architettura Transformer, di cui si è accennato in 1.2. Una direzione per certi versi molto lontana da quanto portato avanti in questa ricerca. Pur trattandosi sempre alla base di reti neurali artificiali, sistemi costituiti sulla falsariga della connessione tra un gran numero di elementi computazionali tra loro identici, memori dei neuroni del nostro cervello, e pur aderendo entrambi a principi rigorosamente empiristi, ricavando interamente le loro capacità dell'esperienza, esiste una differenza non tanto di principio quanto di scala. Una volta adottato l'approccio felice del Transformer, il progresso a cui si è assistito è andato di pari passo con l'aumento del numero di parametri complessivi dei modelli, che oggi ammontano a centinaia di miliardi, e delle dimensioni dei campioni forniti durante l'addestramento. Per i modelli del linguaggio con prestazioni più sofisticate i dataset per l'addestramento corrispondono ad ordini di grandezza superiori alla quantità di testo che un normale essere umano potrebbe aver letto nell'intera sua vita.

Il panorama della manutenzione preventiva è risultato essere caratterizzato da uno scenario ben lontano. Di fronte alla estrema variabilità delle condizioni degli impianti, e delle possibili motivazioni dei loro malfunzionamenti, i dati storici completi ed affidabili risultano estremamente scarsi. Talvolta, come è stato per il primo ambito applicativo delle turbine eoliche, la disponibilità di dati per la ricerca è ulteriormente aggravata dai comprensibili vincoli di riservatezza dei gestori privati della manutenzione. Esiste notoriamente una precisa corrispondenza tra l'ampiezza del campione storico su cui addestrare un modello neurale, e il numero

di parametri che esso contenga, pertanto i modelli qui impiegati sono necessariamente agli antipodi rispetto ai correnti modelli del linguaggio. Il modello predittivo dei malfunzionamenti per gli impianti fotovoltaici, nelle sue diverse versioni illustrate nel capitolo 3, non supera mai il paio di migliaia di parametri, il modello per la correzione della misura di corrosione, descritto nel quarto capitolo, è ancor più parsimonioso, non superando i 500 parametri.

Sicuramente questa situazione ha contribuito alla bassa incidenza finora del deep learning nella letteratura sulla manutenzione preventiva, unitamente alla distanza disciplinare tra l'ambito ingegneristico della produzione di energie alternative e quello dell'Intelligenza Artificiale e delle reti neurali in particolare.

Se da un lato queste caratteristiche, qui sottolineate tramite il paragone con l'attuale applicazione del deep learning di maggior successo, pongono un obiettivo ostacolo per la sua adozione, da un altro verso rappresentano pure un'interessante opportunità di esplorare quali possibilità possano avere le reti neurali artificiali profonde, quando applicate in scenari con premesse davvero poco incoraggianti.

Tuttavia, proprio l'ambito dove attualmente il deep learning riscuote il maggior successo, il linguaggio naturale, ha stimolato l'invenzione di varie soluzioni precedenti il Transformer, che sono state prese a prestito per questa ricerca. L'anello di congiunzione è la necessità di ricordare o dimenticare, a seconda di quanto sia di volta in volta vantaggioso farlo. Sia per le parole che si ascoltano durante una conversazione, di cui siamo maestri nel buttar via quelle inutili e mantenere le poche su cui vogliamo organizzare le nostre risposte. Sia per energie, tensioni e correnti che vengono misurate in un impianto, di cui certe sottili configurazioni temporali possono essere foriere di imminenti malfunzionamenti.

In conclusione, di questa ricerca, si può affermare che i margini di utilizzabilità del deep learning per la manutenzione preventiva, nonché per tematiche ad essa correlate come il monitoraggio della corrosione nei materiali, sembrano

promettenti nonostante la povertà di campioni per addestramento. Sono sicuramente risultati che vanno presi con cautela, anzitutto per il primo caso applicativo, le macchine eoliche. Si tratta di un caso diverso, stavolta non problematico per il numero di campioni per l'addestramento, quanto per la loro natura. Come raccontato nel secondo capitolo, non è stato possibile accedere ad alcun dato reale, pertanto il modello è stato sviluppato e testato solamente con dati SCADA simulati.

L'unico elemento di realismo deriva dall'aver preso dei dati storici reali dell'andamento del vento nell'arco di un anno, ma il guasto è stato interamente simulato. Ovviamente questa condizione permette di generare un numero arbitrario di campioni, e infatti era stato simulato un parco macchine di 100 unità, ma rimane incognito quanto la simulazione si avvicini a condizioni di guasti reali. Per gli impianti fotovoltaici sono stati impiegati dati reali, ma piombando nel pieno della problematica della loro scarsità, rispetto all'estrema variabilità delle tipologie di impianti presenti, e soprattutto del genere di guasti e delle loro possibili cause. In queste condizioni avere un unico modello in grado di effettuare buone predizioni su tre impianti distinti (i due Enim, simili ma in aree distanti, l'impianto Aligrup completamente diverso), è stato un risultato insperato. Anche qui la cautela è comunque d'obbligo, in quanto l'esiguità dei campioni non ha concesso la loro divisione in un insieme da usare solo per l'addestramento, e un altro insieme non visibile al modello, da usare solo per validazione. Tuttavia, il risultato permette quantomeno la confidenza che la scala e tipologia del modello lo rende in grado di cogliere certe relazioni tra i dati di ingresso in grado di discriminare se si sia nelle prossimità di un malfunzionamento (nello specifico del componente inverter). Il grado di generalizzazione che un modello di questo tipo possa avere, per impianti fotovoltaici diversi da quelli su cui è stato addestrato, rimane incognito.

Infine, nel caso della corrosione i dati a disposizione erano senz'altro limitati, ma non come in quello precedente, e quindi si è potuto procedere come di prassi, a relegare parte dei dati per la validazione del modello. Lo si è fatto nel modo più drastico, ovvero anziché destinare alla validazione intervalli temporali, relativi a sensori di cui comunque altri intervalli temporali erano affidati al modello come training, di tenere un sensore completamente sconosciuto al modello. Si è proceduto a rotazione, mantenendo dei nove sensori testati negli esperimenti dell'Università di Praga otto per training e uno per validazione. Anche in questa applicazione i risultati sono stati ben più confortanti delle aspettative, il modello mostra di poter correggere il rumore del sensore in modo migliore rispetto al filtraggio matematico attualmente impiegato. Va ancora una volta ribadita la cautela, stavolta legata all'innovazione dell'oggetto di modellazione stesso: il sensore di corrosione, non ancora in produzione e su cui sono in corso gli esperimenti di valutazione. In particolare, questi esperimenti sono fisicamente possibili nella condizione di assenza totale di ossigeno, sapendo quindi che la perdita di spessore per corrosione deve essere rigorosamente zero.

Diventa però difficile poter valutare come si comporti il modello nel correggere il sensore, nel momento in cui i valori di perdita di spessore metallico non sono più zero.

Come ricordato nel primo capitolo, le reti neurali artificiali sono state il tentativo di importare elementi strutturali della cognizione umana nella computazione artificiale, a beneficio di una varietà di applicazioni pratiche. Un tentativo in vari periodi storici considerato azzardato, bizzarro, mal posto, ma che da una decina di anni si è dimostrato particolarmente felice.

In definitiva, pur se con tutte le attenzioni appena sottolineate, l'insieme delle ricerche e degli sviluppi modellistici descritti in questa tesi, portano a considerare questo traghettamento della mente in software applicabile anche in ambiti, come

quello della manutenzione preventiva in impianti di energie rinnovabili, come fattibile, nonostante certe premesse poco favorevoli.

BIBLIOGRAFIA

- Alnatheer, *The potential contribution of renewable energy to electricity supply in Saudi Arabia*, Energy Policy, 33 (18) (2005), pp. 2298-2312.
- AM Paraschiv, M Danubianu, *BIG DATA - Present Opportunities and Challenges*. Brain. Broad Research in Artificial Intelligence & Neuroscience. 2019 Special Issue, Vol. 10, p.15-21; Terzi, D.S.
- Appiah AY, Zhang X, Ayawli BBK, et al. Long short-term memory networks based automatic feature extraction for photovoltaic array fault diagnosis. IEEE Access 2019:1.
- Arslan, B.; Sagiroglu, S. Smart Grid Security Evaluation with a Big Data Use Case. In Proceedings of the 2018 IEEE 12th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG 2018).
- Burton, T., Jenkins, N., Sharpe, D., & Bossanyi, E. (2011). Wind Energy Handbook. Chichester, United Kingdom: John Wiley & Sons, Ltd.
- Chen Z, Chen Y, Wu L, et al. Deep residual network based fault detection and diagnosis of photovoltaic arrays using current-voltage curves and ambient conditions. Energy Conv Manage 2019;198. 111793.
- Child, M., Kemfert, C., Bogdanov, D., & Breyer, C. (2019). Flexible electricity generation, grid exchange and storage for the transition to a 100% renewable energy system in Europe. Renewable energy, 139, 80-101.
- Chiticariu, Laura, Yunyao Li, and Frederick Reiss. *Transparent machine learning for information extraction: state-of-the-art and the future*. EMNLP (tutorial) (2015).

- Chollet F., *Deep Learning with python*, Manning publications Co, Shelter Island 2018.
- Chomsky, N. (1956). Three models for the description of languages. *IRE Transaction on Information Theory*, 2: 113–124.
- Churchland Paul M. e Patricia S., Could a Machine think? in «Scientific America» N. 262, January 1990 in *Le Scienze* Vol. 44 N. 259, Milano Marzo 1990.
- Cichy, R. M., Khosla, A., Pantazis, D., Torralba, A. and Oliva, A. (2016). Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. *Scientific Reports*, 6: 27755.
- Clarke, A., Devereux, B. J., Randall, B. and Tyler, L. K. (2015). Predicting the time course of individual objects with MEG. *Cerebral Cortex*, 25: 3602–3612.
- E. Gnansounou, A. Dauriat, C.E. Wyman, *Refining sweet sorghum to ethanol and sugar: economic trade-offs in the context of North China*, *Bioresour Technol*, 96 (9) (2005), pp. 985-1002.
- Elman, Jeffrey L. «Finding structure in time». In: *Cognitive science* 14.2 (1990), pp. 179–211.
- *Enciclopedia filosofica*, volume VI, p. 6579, Bompiani.
- Eom S-H, Kim S-S, Lee J-B. Assessment of Anti-Corrosion Performances of Coating Systems for Corrosion Prevention of Offshore Wind Power Steel Structures. *Coatings*. 2020.
- Fingersh, L., Hand, M., & Laxson, A. (2006). Wind Turbine Design Cost and Scaling Model. Colorado, USA: National Renewable Energy Laboratory NREL.
- Fukushima, K. (1980). Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202.

- Gashler MS. *Deep Learning in robotics: a review of recent research*. arXiv:1707.07217. 1–41.
- Gauthier, I. and Tarr, M. J. (2016). Visual object recognition: Do we (finally) know more now than we did? *Annual Review of Vision Science*, 2: 16.1–16.20.
- Gers FA, Schraudolph NN, Schmidhuber J (2002) Learning precise timing with LSTM recurrent networks. *J Mach Learn Res* 3:115– 143.
- Buehler H., L. Gonon, J. Teichmann e B. Wood. “Deep hedging”. In: *Quantitative Finance* 19.8 (2019), pp. 1271–1291. Christian Bayer e Benjamin Stemper. Deep calibration of rough stochastic volatility models. 2018. arXiv: 1810.03399 [q-fin.PR]. Henry Stone. Calibrating rough volatility models: a convolutional neural network approach. 2019. arXiv: 1812.05315 [q-fin.CP].
- H. Yan, J. Wan, C. Zhang, S. Tang, Q. Hua and Z. Wang, *Industrial Big Data Analytics for Prediction of Remaining Useful Life Based on Deep Learning*, in *IEEE Access*, vol. 6, pp. 17190-17197, 2018, doi: 10.1109/ACCESS.2018.2809681.
- Hansen, K., Breyer, C., & Lund, H. (2019). Status and perspectives on 100% renewable energy systems. *Energy*, 175, 471-480.
- Helbing G, Ritter M., *Deep Learning for fault detection in wind turbines*. *Renew Sustain Energy Rev.* 2018;98:189-198. <https://doi.org/10.1016/j.rser.2018.09.012>.
- Heydar Chamandoust, Ghasem Derakhshan, Seyed Mehdi Hakimi, Salah Bahramara, *Tri-objective scheduling of residential smart electrical distribution grids with optimal joint of responsive loads with renewable energy sources*, *Journal of Energy Storage*, volume 27, 2020, 101112, ISSN 2352-152X, <https://doi.org/10.1016/j.est.2019.101112>.
- Hobbes, T. (1651). *Leviathan*. London. English trans. by Edwin Curley, 1994, Indianapolis: Hackett.

- Ibrahim, Raed; Weinert, Jannis; Watson, Simon (2016): *Neural networks for wind turbine fault detection via current signature analysis*. Loughborough University. Conference contribution. <https://hdl.handle.net/2134/23014>
- Ilya Sutskever, James Martens, George Dahl, Geoffrey Hinton ; *Proceedings of the 30th International Conference on Machine Learning*, PMLR 28(3):1139-1147, 2013.
- Ingliński, B.; Iglińska, A.; Koziński, G.; Skrzatek, M.; Buczkowski, M. *Wind energy in Poland—History, current state, surveys*, Renewable Energy Sources Act, SWOT analysis. *Renew. Sustain. Energy Rev.* 2016, 64, 19–33.
- IRENA. (2016). Wind Power - Technology Brief. IEA-ETSAP and IRENA.
- Italiano, C. E. (2016). Reference technical rules for the connection of active and passive consumers to the HV and MV electrical networks of distribution Company. (CEI 0–16.)
- Italiano, C. E. (2019). Reference Technical Rules for the Connection of Active and Passive Users to the LV Electrical Utilities. (CEI 0–21)
- J. Wang, C. Jiang, H. Zhang, Y. Ren, K. Chen and L. Hanzo, *Thirty Years of Machine Learning: The Road to Pareto-Optimal Wireless Networks*, in *IEEE Communications Surveys & Tutorials*, doi: 10.1109/COMST.2020.2965856.
- J.K. Kaldellis, D.S. Vlachou, G. Korbakis, *Techno-economic evaluation of small hydro power plants in Greece: a complete sensitivity analysis*, *Energy Policy*, 33 (15) (2005), pp. 1969-1985.
- J.M. Huacuz, *The road to green power in Mexico—reflections on the prospects for the large-scale and sustainable implementation of renewable energy*, *Energy Policy*, 33 (16) (2005), pp. 2087- 2099.
- Jardine, A.; Tsang, A. *Maintenance, Replacement, and Reliability—Theory and Applications*, 2nd ed.; Taylor and Francis, Boca: Boca Raton, FL, USA, 2006; ISBN 0-8493-3966-9.

- Jefferson, G. (1949). The mind of mechanical man. *British Medical Journal*, 1(4616), 1105.
- John R. Searle, (1980) Minds, brains, and programs. *Behavioral and Brain Sciences* 3 (3): p. 3.
- Jordan, D. C., Silverman, T. J., Wohlgemuth, J. H., Kurtz, S. R., & VanSant, K. T. (2017). Photovoltaic failure and degradation modes. *Progress in Photovoltaics: Research and Applications*, 25(4), 318-326.
- Just Have a Think. (17. 07 2021). Deep Ocean Floating wind power. Tratto da Just Have a Think: <https://www.justhaveathink.com/wp-content/uploads/2021/07/Prog-158- Transcript.docx>.
- Krizhevsky A, Sutskever I, Hinton GE. *ImageNet classification with deep convolutional neural networks*. *Adv Neural Inf Process Syst*. 2012;1:1-9.
- Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). *ImageNet classification with deep convolutional neural networks*. In *Advances in neural information processing systems* (pp. 1090–1098).
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). *ImageNet classification with deep convolutional neural networks*. In *Advances in Neural Information Processing Systems*, pages 1090–1098.
- Lars, L.; Elizabeth, T., *The benefits, risks, and future of artificial intelligence in solar and wind energy*. In *Making Renewables Smarter*; Group Technology and Research Position Paper; DNV GL: Oslo, Norway, 2017.
- LeCun, Yann, Léon Bottou, Yoshua Bengio e Patrick Haffner. «Gradient based learning applied to document recognition». In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.
- Litjens G, Kooi T, Bejnordi BE, et al. *A survey on Deep Learning in medical image analysis*. *Med Image Anal J*. 2017;42:60- 88. <https://doi.org/10.1016/j.media.2017.07.005>.

- Liu, Y., Ding, K., Zhang, J., Li, Y., Yang, Z., Zheng, W., & Chen, X. (2021). Fault diagnosis approach for photovoltaic array based on the stacked auto-encoder and clustering with IV curves. *Energy Conversion and Management*, 245, 114603.
- Löffler, K., Burandt, T., Hainsch, K., & Oei, P. Y. (2019). Modeling the low-carbon transition of the European energy system-a quantitative assessment of the stranded assets problem. *Energy Strategy Reviews*, 26, 100422.
- Long Wen, Yan Dong, Liang Gao. *A new ensemble residual convolutional neural network for remaining useful life estimation*. *Mathematical Biosciences and Engineering*, 2019, 16(2): 862-880. doi: 10.3934/mbe.2019040.
- Love, B.C. *Comparing supervised and unsupervised category learning*. *Psychonomic Bulletin & Review* 9, 829–835 (2002).
- Lu, X., Lin, P., Cheng, S., Lin, Y., Chen, Z., Wu, L., & Zheng, Q. (2019). Fault diagnosis for photovoltaic array based on convolutional neural network and electrical time series graph. *Energy Conversion and Management*, 196, 950-965.
- Maienza, C., Avossa, A. M., Ricciardelli, F., Coiro, D., Troise, G., & Georgakis, C. T. (2020). A life cycle cost model for floating offshore wind farms. *Applied Energy*, 266, 114716.
- Martinez-Mosquera, D.; Navarrete, R.; Lujan-Mora, S. *Modeling and Management Big Data in Databases - A Systematic Literature Review*. *Sustainability* 2020, 12, 63
- McCarthy, J., Minsky, M. L., Rochester, N. and Shannon, C. E. (1955). A proposal for the Dartmouth summer research project on artificial intelligence, August 31, 1955. reprinted in *AI Magazine*, 27: 12–14.
- Min S, Lee B, Yoon S. *Deep Learning in bioinformatics*. *Brief, Bioinform.* 2017;18(5):851 869.

- Minsky, M. and Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge (MA).
- Montes G.M., del Mar Serrano Lopez M., M. del Carmen Rubio Gamez, A.M. Ondina, *An overview of renewable energy in Spain. The small hydro-power case*, *Renewable Sustainable Energy Rev*, 9 (5) (2005), pp. 521-534.
- Newell, A., Shaw, J. C., & Simon, H. A. (1959, June). Report on a general problem solving program. In *IFIP congress* (Vol. 256, p. 64).
- Nguyen VN, Jenssen R, Roverso D., *Electrical power and energy systems automatic autonomous vision-based power line inspection: a review of current status and the potential role of Deep Learning*. *Electr Power Energy Syst*. 2018;99:107- 120. <https://doi.org/10.1016/j.ijepes.2017.12.016>.
- Nithya M., S. Nagarajan and P. Navaseelan, *Fault detection of wind turbine system using neural networks*, *2017 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR), Chennai, 2017*, pp. 103-108, doi: 10.1109/TIAR.2017.8273694
- Normativa UNI 10584 Manutenzione – Terminologia, 2003.
- Novaes, C. D., 2012. *Formal languages in logic: a philosophical and cognitive analysis*. Cambridge University Press, Cambridge (UK).
- NVIDIA. *Deep Learning for self-driving cars*. <https://www.nvidia.com/en-us/deep-learning-ai/industries/automotive/>. Accessed May 10, 2018.
- Olshausen, B. A. (2014). Perception as an inference problem. In Gazzaniga, M. S., editor, *The Cognitive Neurosciences*, pages 295–304. MIT Press, Cambridge (MA). Fifth edition.
- Patricia Fara (2020) *Ada Lovelace: the making of a computer scientist*, *Women's History Review*, 29:1, 168 170.

- Perconti, Pietro & Plebe, Alessio (2020). *Deep Learning and cognitive science*. *Cognition* 203:104365.
- Plebe, A., & Grasso, G. (2016). *The brain in silicon: History, and skepticism*. In F. Gadducci & M. Tavosanis (Eds.), *History and philosophy of computing* (pp. 273–286). Berlin: Springer.
- Plebe, A., & Grasso, *The Unbearable Shallow Understanding of Deep Learning*, In *Minds and Machine* 29:515-553.
- Plebe, A., Perconti, P. *The Future of the Artificial Mind*. Taylor&Francis Group, LLC 2022.
- Plessmann, G., & Blechinger, P. (2017). How to meet EU GHG emission reduction targets? A model based decarbonization pathway for Europe's electricity supply system until 2050. *Energy Strategy Reviews*, 15, 19-32.
- Price SJ, Figueira RB. Corrosion Protection Systems and Fatigue Corrosion in Offshore Wind Structures: Current Status and Future Perspectives. *Coatings*. 2017; 7(2):25. <https://doi.org/10.3390/coatings7020025>
- R. Asadpour, R. V. K. Chavali and M. A. Alam, "Physics-Based computational modeling of moisture ingress in solar modules: Location-specific corrosion and delamination," *2016 IEEE 43rd Photovoltaic Specialists Conference (PVSC)*, Portland, OR, USA, 2016, pp. 0840-0843, doi: 10.1109/PVSC.2016.7749725.
- R. Asadpour, X. Sun and M. A. Alam, "Electrical Signatures of Corrosion and Solder Bond Failure in c-Si Solar Cells and Modules," in *IEEE Journal of Photovoltaics*, vol. 9, no. 3, pp. 759-767, May 2019, doi: 10.1109/JPHOTOV.2019.2896898.
- R. Keith Mobley, *An introduction to predictive maintenance*, 2002, Elsevier Science (USA).

- R. Duke, R. Williams, A. Payne, *Accelerating residential PV expansion: demand analysis for competitive electricity markets*, Energy Policy, 33 (15) (2005), pp. 1912-1929
- Rogers, T. T. and McClelland, J. L. (2014). Parallel distributed processing at 25: Further explorations in the microstructure of cognition. Cognitive Science, 38: 1024–1077.
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organisation in the brain. Psychological Review, 65: 386–408.
- Rosenblatt, F. (1958). *The perceptron: A probabilistic model for information storage and organization in the brain*. Psychological Review, 65(6), 386 - 408. <https://doi.org/10.1037/h0042519>;
- Rosenblatt, F. (1959). Two theorems of statistical separability in the perceptron. In Mechanisation of thought processes: Proceedings of symposium No. 10, pages 419–472, London. H. M. Stationery Office.
- Rosenblatt, F. (1962). *Principles of neurodynamics: Perceptron and the theory of brain mechanisms*. Washington (DC): Spartan.
- Rumelhart, D. E., McClelland, J. L., 1986a. *On learning the past tenses of English verbs*. In: Rumelhart and McClelland (1986b), pp. 216{271.
- S. Sato and K. Sanda, *Degradation Estimation of Turbines in Wind Farm Using Denoising Autoencoder Model*, 2019 IEEE International Conference on Prognostics and Health Management (ICPHM), San Francisco, CA, USA, 2019, pp. 1-6, doi: 10.1109/ICPHM.2019.8819375.
- Samek, Wojciech, et al. *Toward Interpretable Machine Learning: Transparent Deep Neural Networks and Beyond*, arXiv preprint arXiv:2003.07631 (2020).
- Schmidhuber, J. (2015). *Deep Learning in neural networks: An overview*. Neural Networks, 61, 85–117.

- Searle, J. R. (1980). Mind, brain and programs. *Behavioral and Brain Science*, 3: 417–424.
- Shafer, T. *The 42 V's of Big Data and Data Science*. Available online: <https://www.elderresearch.com/blog/42-v-of-big-data>;
- Sharifi, V., Abdollahi, A., & Rashidinejad, M. (2022). Flexibility-based generation maintenance scheduling in presence of uncertain wind power plants forecasted by deep learning considering demand response programs portfolio. *International Journal of Electrical Power & Energy Systems*, 141, 108225.
- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., et al. (2016). *Mastering the game of Go with deep neural networks and tree search*. *Nature*, 529, 484–489.
- T. Mikolov, M. T. Kara, L. Burget, J. Cernock, and S. Khudanpur, *Recurrent neural network based language model*, *Interspeech*, vol. 2, p. 3, Sep. 2010.
- Turing, A. (1936). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42: 230–265.
- Turing, A., 1948. *Intelligent machinery*. Tech. rep., National Physical Laboratory, London, raccolto in Ince, D. C. (ed.) *Collected Works of A. M. Turing: Mechanical Intelligence*, Edinburgh University Press, 1969.
- Turing, *Computing Machinery and Intelligence*, 1950, *Mind* (vol.49, pp. 433-460).
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.
- Vespignani, A., *L'algoritmo e l'oracolo*, il saggiatore, Milano 2019, pp. 71 e sg.

- W. Qiao and D. Lu, *A Survey on Wind Turbine Condition Monitoring and Fault Diagnosis—Part I: Components and Subsystems*, in *IEEE Transactions on Industrial Electronics*, vol. 62, no. 10, pp. 6536-6545, Oct. 2015, doi: 10.1109/TIE.2015.2422112.
- Wallace, R. S. (2008). The anatomy of A.L.I.C.E. In Epstein et al. (2008), pages 181–210.
- Xiaoli Li and Jun Huan, *Constructivism Learning: A Learning Paradigm for Transparent Predictive Analytics*, 2017.
- Y. Xu, Y. Sun, J. Wan, X. Liu, and Z. Song, *Industrial big data for fault diagnosis: Taxonomy, review, and applications*, *IEEE Access*, to be published, doi: 10.1109/ACCESS.2017.2731945. L. Linxia and F. Köttig, *Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction*, *IEEE Trans. Rel.*, vol. 63, no. 1, pp. 191_207, Mar. 2014.
- Yang Li, Zhen Yang, Guoqing Li, Dongbo Zhao, Wei Tian, *Optimal scheduling of an isolated microgrid with battery storage considering load and renewable generation uncertainties*, *IEEE Trans. Ind. Electron.*, 22 (2019), pp. 1565-1575.
- Yue T, Wang H., *Deep Learning for genomics: a concise overview*. Handbook of Deep Learning Applications; 2018. <https://doi.org/10.1021/acs.molpharmaceut.5b00982>.