



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

UNIVERSITA' DEGLI STUDI DI PADOVA

Dipartimento di Ingegneria Industriale DII

Corso di Laurea Magistrale in Ingegneria dell'Energia Elettrica

**Previsione dei prezzi zonali del mercato italiano tramite tecniche di
machine learning**

Relatore: Prof. Bignucolo Fabio

Supervisione Aziendale: Dott. Clementi Riccardo

Laureanda: Basso Alice

N° Matricola: 1205793

Anno Accademico 2019/2020

Indice

Introduzione	9
Capitolo 1 - Il mercato elettrico in Italia	10
1.1 Aspetti storici: dal monopolio alla liberalizzazione.....	10
1.2 La Borsa Elettrica	13
1.2.1 Il Mercato del Giorno Prima	14
1.2.2 Il Mercato Infra-giornaliero e il Mercato dei Servizi di Dispacciamento.....	16
Capitolo 2 - Il prezzo dell'energia elettrica.....	18
2.1 Caratteristiche del prezzo dell'energia elettrica.....	18
2.2 Tecniche di previsione del prezzo dell'energia elettrica: stato dell'arte.....	23
2.2.1 Modelli di simulazione	23
2.2.2 Modelli Multi-Agent.....	24
2.2.3 Modelli statistici.....	25
2.2.4 Modelli di intelligenza computazionale.....	27
2.2.5 Modelli HIM e previsioni combinate.....	28
Capitolo 3 - Il Machine Learning e il Modello LSTM.....	29
3.1 Tipologie di algoritmi	30
3.2 Le reti neurali ricorrenti	31
3.3 Il modello "Long Short Term Memory"	33
3.4 Le librerie Python	38
3.5 La struttura e il codice del modello LSTM	41
3.5.1 La preparazione dei dati.....	41
3.5.2 La normalizzazione dei dati	43
3.5.3 La suddivisione dei dati	46
3.5.4 La definizione del modello e la funzione di attivazione	47
3.5.6 La compilazione del modello.....	52
3.5.7 L'addestramento del modello	55
3.5.8 La valutazione del modello.....	58

Capitolo 4 - Previsione del Prezzo zonale dell'energia elettrica.....	59
4.1 Le tipologie di analisi e la features considerate	59
4.1.1 La potenza assorbita.....	61
4.1.2 Il prezzo del gas	62
4.1.3 La radiazione solare	64
4.1.4 La velocità del vento.....	66
4.1.5 Giorno feriale o festivo	68
4.1.6 Ora e mese di tipo ciclico.....	69
4.2 Il modello LSTM per la previsione oraria e giornaliera	70
4.3 Previsione oraria del prezzo zonale dell'energia elettrica	72
4.4 Previsione giornaliera del prezzo zonale dell'energia elettrica	83
Capitolo 5 – Conclusioni.....	94
5.1 Confronto tra analisi oraria e giornaliera	94
5.2 Considerazioni finali e possibili sviluppi futuri.....	96
Bibliografia.....	97

Elenco delle figure

Figura 1.1 Equilibrio tra domanda e offerta per la determinazione del prezzo.....	15
Figura 2.1 Andamento del prezzo dell'energia in Italia nei giorni Domenica 4 Ottobre 2020 /Lunedì 5 Ottobre 2020.....	19
Figura 2.2 Andamento della domanda reale e prevista in Italia nei giorni Domenica 4 Ottobre 2020 e Lunedì 5 Ottobre 2020.....	20
Figura 2.3 Andamento settimanale del prezzo nelle sei zone del mercato elettrico italiano nella settimana del 5 Ottobre 2020.....	21
Figura 2.4 Andamento del prezzo dell'energia in Italia nel 2015.....	22
Figura 3.1 Algoritmo tradizionale (sinistra) algoritmo di machine learning (destra).....	29
Figura 3.2 Diverse tipologie di machine learning.....	31
Figura 3.3 Struttura rete neurale artificiale.....	32
Figura 3.4 Rete neurale ricorrente.....	32
Figura 3.5 Rete neurale LSTM.....	34
Figura 3.6 Struttura rete neurale tradizionale e rete neurale LSTM.....	34
Figura 3.7 Cella di una rete neurale LSTM.....	35
Figura 3.8 Struttura di un "gate" presente in una cella di una rete neurale LSTM.....	36
Figura 3.9 Step 1 Cella LSTM.....	36
Figura 3.10 Step 2 Cella LSTM.....	37
Figura 3.11 Step 3 Cella LSTM.....	37
Figura 3.12 Step 4 Cella LSTM.....	38
Figura 3.13 Struttura di un DataFrame.....	40
Figura 3.14 DataFrame in formato Excel.....	42
Figura 3.15 Dati reali e dati scalati con MinMaxScaler.....	45
Figura 3.16 Suddivisione dei dati.....	47
Figura 3.17 Funzione di attivazione ReLU.....	51
Figura 3.18 Overfitting e Underfitting.....	57
Figura 3.19 Confronto tra la funzione di perdita valutata sul train set e validation set nei tre diversi	

casi.....	57
Figura 4.1 Previsione oraria.....	59
Figura 4.2 Previsione giornaliera.....	60
Figura 4.3 Prezzo dell'energia elettrica (Euro/MWh).....	60
Figura 4.4 Andamento del prezzo dell'energia elettrica negli anni 2015/2016.....	61
Figura 4.5 Potenza assorbita (MW).....	62
Figura 4.6 Prezzo del gas (Euro/MWh).....	64
Figura 4.7 Produzione da fotovoltaico.....	65
Figura 4.8 Radiazione solare (W/m^2).....	66
Figura 4.9 Distribuzione provinciale della produzione eolica nel 2015.....	67
Figura 4.10 Distribuzione provinciale della produzione eolica nel 2016.....	67
Figura 4.11 Velocità del vento (m/s).....	68
Figura 4.12 Giorno feriale o festivo.....	69
Figura 4.13 Ora e mese di tipo ciclico.....	70
Figura 4.14 Analisi oraria: confronto funzione di perdita Train/Validazione.....	75
Figura 4.15 Analisi oraria: andamento valore prezzo Train/Test/Previsioni.....	76
Figura 4.16 Analisi oraria: confronto valori reali e previsti.....	76
Figura 4.17 Analisi oraria: errore commesso nella previsione (differenza tra valore previsto e valore reale).....	77
Figura 4.18 Analisi oraria: confronto valori reali e trend di prezzo.....	78
Figura 4.19 Analisi oraria: confronto valori reali e trend di prezzo Dicembre 2016.....	78
Figura 4.20 Analisi oraria: distribuzione dell'errore.....	79
Figura 4.21 Analisi oraria: distribuzione dell'errore di trend.....	80
Figura 4.22 Analisi oraria: Prezzo reale – Prezzo predetto.....	81
Figura 4.23 Analisi oraria: Prezzo reale – Trend.....	82
Figura 4.24 Analisi giornaliera: confronto funzione di perdita Train/Validazione.....	86
Figura 4.25 Analisi giornaliera: andamento valore prezzo Train/Test/Previsioni.....	87
Figura 4.26 Analisi giornaliera: confronto valori reali e previsti.....	87
Figura 4.27 Analisi giornaliera: errore commesso nella previsione (differenza tra valore previsto e valore reale).....	88

Figura 4.28 Analisi giornaliera: confronto valori reali e trend di prezzo.....	89
Figura 4.29 Analisi giornaliera: confronto valori reali e trend di prezzo Dicembre 2016.....	89
Figura 4.30 Analisi giornaliera: distribuzione dell'errore.....	90
Figura 4.31 Analisi giornaliera: distribuzione dell'errore di trend.....	91
Figura 4.32 Analisi giornaliera: Prezzo reale – Prezzo predetto.....	92
Figura 4.33 Analisi giornaliera: Prezzo reale – Trend.....	93

Elenco delle tabelle

Tabella 1 Confronto Prezzi Reali e Prezzi Scalati	44
Tabella 2 Analisi effettuate	71
Tabella 3 Caso 1 – Previsione Oraria.....	73
Tabella 4 Caso 2 – Previsione Oraria.....	73
Tabella 5 Caso 3 – Previsione Oraria.....	74
Tabella 6 Caso 4 – Previsione Oraria.....	74
Tabella 7 RMSE Mensili – Previsione Oraria.....	82
Tabella 8 Caso 1 – Previsione Giornaliera.....	84
Tabella 9 Caso 2 – Previsione Giornaliera.....	84
Tabella 10 Caso 3 – Previsione Giornaliera.....	85
Tabella 11 Caso 4 – Previsione Giornaliera.....	85
Tabella 12 RMSE Mensili – Previsione Giornaliera.....	93
Tabella 13 Confronto RMSE analisi oraria e giornaliera (valori predetti)	94
Tabella 14 Confronto RMSE analisi oraria e giornaliera (trend).....	95

Elenco degli script

Script 1 Impostazione data e ora come indice del DataFrame	42
Script 2 Normalizzazione dei dati mediante MinMaxScaler	44
Script 3 Modello LSTM	48
Script 4 Compilazione modello con funzione di perdita MSE.....	54
Script 5 Compilazione modello con funzione di perdita MSLE	54
Script 6 Compilazione modello con funzione di perdita MAE	55
Script 7 Addestramento del modello LSTM	55
Script 8 Applicazione del modello LSTM	58
Script 9 Funzione ‘create_dataset’ per previsione oraria	72
Script 10 Funzione ‘create_dataset’ per previsione giornaliera	83

Introduzione

L'obiettivo che si pone questo lavoro di tesi è l'analisi delle caratteristiche che contraddistinguono il prezzo zonale del mercato elettrico italiano e lo sviluppo di modelli di machine learning atti alla previsione di tale grandezza. Dopo un iniziale approfondimento riguardante la struttura e il funzionamento del mercato elettrico, nel capitolo due si è passati alla descrizione della grandezza oggetto dello studio e delle caratteristiche che ne determinano l'andamento nel tempo. Con l'avvento della liberalizzazione del mercato e la conseguente nascita di nuovi produttori e intermediari, si è assistito ad un incremento della competitività nel mercato stesso, che lo ha reso a tutti gli effetti un settore soggetto alle regole della libera concorrenza. Per affrontare le nuove sfide e i nuovi obiettivi, i diversi operatori necessitano dunque di strumenti sempre più sofisticati e all'avanguardia. Il prezzo dell'energia è, per sua natura, molto volatile e affetto da stagionalità sia su base giornaliera, che settimanale e mensile; appare evidente, quindi, come sia necessario disporre di modelli di previsione adeguati, per rendere consapevoli ed efficaci le scelte di ogni soggetto operante nel mercato. Nei capitoli tre e quattro vengono presentati a tal fine due modelli di machine learning, in particolare due reti neurali LSTM, in grado di svolgere previsioni orarie e giornaliere del prezzo dell'energia elettrica, prendendo in considerazione le diverse grandezze che ne influenzano l'andamento; tali modelli sono poi stati applicati al caso specifico del mercato elettrico in Sardegna. Infine, si è eseguito un confronto tra i due diversi approcci, evidenziando punti deboli, possibili migliorie e futuri sviluppi.

Capitolo 1 - Il mercato elettrico in Italia

1.1 Aspetti storici: dal monopolio alla liberalizzazione

Il settore elettrico in Italia nasce grazie all'iniziativa di operatori locali che ritennero vantaggioso avviare questo tipo di attività in un territorio limitato, spesso urbano, adottando una struttura verticalmente integrata. Le fasi di generazione, trasmissione, distribuzione e vendita erano tutte di competenza di un medesimo operatore, che agiva quindi da monopolista locale. Con il progredire della tecnica e lo sviluppo industriale questi operatori avevano ciascuno rafforzato le proprie linee di distribuzione, evitando però di entrare in conflitto con le aree di altri operatori adiacenti. La ragione alla base di questa evoluzione sta nella condizione di monopolio tipica delle infrastrutture di rete, per le quali un'estensione della struttura di distribuzione porta a benefici, in termini di economie di scala, solo entro una certa soglia. Superata questa, gli operatori andavano incontro a costi medi superiori, che risultavano però comunque inferiori a quelli che avrebbero dovuto sostenere per costruire una rete di distribuzione totalmente nuova e contigua tra le varie zone. Le conseguenze di questa situazione erano sostanzialmente due: innanzitutto, un bene come l'energia elettrica, divenuto ormai di utilità pubblica, era in mano a dei singoli soggetti senza nessuna forma di tutela per i cittadini. In secondo luogo, alcune aree, non ancora elettrificate a causa degli scarsi vantaggi economici per gli operatori, rimanevano prive della fornitura di energia elettrica. Inoltre, nel secondo dopoguerra, il settore elettrico risentì in tutta Europa del nuovo clima politico: l'intero continente fu investito dall'ideologia dello "sviluppo economico" e da sentimenti fortemente antimonopolistici, che in paesi quali Francia e Inghilterra si concretizzarono nella nazionalizzazione del settore. L'obiettivo, infatti, era che lo Stato intervenisse per evitare che le imprese traessero un ingiusto beneficio a causa della loro posizione dominante. In Italia invece il tema della nazionalizzazione rimase limitato al solo dibattito politico fino al 6 dicembre 1962 quando venne istituito l'ENEL (Ente Nazionale per l'Energia Elettrica). Lo stato italiano, ispirandosi al modello della francese EDF (Électricité de France), acquistò e trasferì tutte le imprese precedentemente private in un unico ente, orientando la sua attività al raggiungimento della piena integrazione del sistema elettrico nazionale attraverso il monopolio di ciascuna fase della produzione, della trasmissione e della distribuzione dell'energia. L'obiettivo dell'ENEL era

quello di creare un unico centro per il dispacciamento dei flussi di energia elettrica e costruire una rete di trasmissione, cioè di distribuzione ad altissimo voltaggio, in grado di unificare le varie aree elettrificate sparse nel territorio. Per quanto riguarda gli impianti, nel primo decennio di attività l'ENEL procedette alla loro specializzazione, di base e di punta, allargando l'offerta termoelettrica con grandi impianti destinati a coprire il carico di base e riconvertendo progressivamente alla fornitura di punta gli impianti idroelettrici a serbatoio. L'attenzione per i sistemi integrati si attenuò negli anni Ottanta a causa di fattori quali il successo delle teorie neoliberiste che promuovevano la libera concorrenza e la nascita di nuove tecnologie, come ad esempio le centrali a ciclo combinato e di cogenerazione e lo sviluppo di software informatici in grado di rendere più flessibile la produzione e più efficiente l'utilizzazione dell'energia. In Italia il mercato elettrico, vale a dire la sede delle transazioni aventi per oggetto l'energia elettrica, nasce per effetto del Decreto Legislativo 16 marzo 1999 n. 79 (noto come "Decreto Bersani") [3], nell'ambito del processo di recepimento della direttiva comunitaria sulla creazione di un mercato interno dell'energia (Direttiva 96/92/CE abrogata dalla Direttiva 2003/54/CE). Come si è visto, dalla nazionalizzazione del 1962, il sistema elettrico italiano era stato organizzato in forma di monopolio pubblico verticalmente integrato. Le attività di generazione, trasmissione, distribuzione e vendita di energia elettrica, infatti, erano operate sulla quasi totalità del territorio nazionale da un unico soggetto: l'ENEL. Il decreto Bersani introdusse l'*unbundling* legale delle reti, ossia un regime di separazione giuridica tra la loro gestione e le attività di produzione e vendita di energia elettrica, e aprì alla concorrenza la generazione consentendo ai privati di realizzare nuove centrali. Per ridurre la concentrazione del mercato, da una parte fissò un tetto del 50% a livello nazionale sui volumi di energia elettrica producibili da Enel, dall'altra impose la cessione di 15 GW di capacità. Questa misura dette vita alle cosiddette GenCo (Generation companies): Interpower, Elettrogen, Eurogen, da cui nacquero i primi concorrenti di grandi dimensioni. Di lì a breve seguirono la quotazione in borsa e la cessione del primo pacchetto di azioni dell'Enel (poco più del 30 per cento). Gli obiettivi del decreto Bersani sono quindi così riassumibili: apertura del mercato; terzietà delle reti; riorganizzazione dell'Enel, con la creazione al suo interno di tre distinte società per la generazione (Enel Produzione), trasmissione (Terna) e distribuzione (Enel Distribuzione, oggi E-Distribuzione). Contestualmente, fu istituito anche il Gestore dei mercati energetici (GME) per l'organizzazione e gestione della borsa elettrica. Il decreto Bersani si inserisce nell'ambito di un

disegno dal respiro europeo: l'apertura dei mercati era infatti strumentale alla realizzazione di un mercato interno, che promuovesse lo sviluppo economico e sociale dell'Europa. Il decreto del 1999 fece solo i primi, sebbene imprescindibili, passi verso l'apertura del mercato; servirono infatti altre due direttive ("Pacchetto Energia" del 2003 e 2009) e i relativi decreti di recepimento a completare le regole per il mercato interno dell'energia elettrica. Con il secondo e terzo Pacchetto Energia (recepiti in Italia rispettivamente nel 2007 e 2011) si riuscì ad imporre in modo coordinato in tutti gli stati UE la separazione societaria tra attività di distribuzione e vendita, avviare la piena liberalizzazione della vendita al dettaglio dal 2007 e adottare modelli più incisivi di *unbundling* per la trasmissione, oltre a rafforzare e ampliare obiettivi e funzioni dell'autorità di regolazione di settore. I cambiamenti prodotti dal decreto Bersani e da quelli successivi sono evidenti nella crescita della capacità installata, che dal 1999 al 2007 è cresciuta da 61 GW a 109 GW, con mutamenti altrettanto significativi nella diversificazione delle fonti: l'olio combustibile (che inizialmente aveva rimpiazzato il nucleare) è pressoché scomparso grazie all'avvento dei cicli combinati a gas, mentre eolico e fotovoltaico sono balzati da quasi zero a circa 10 e 18 GW, rispettivamente. Questi cambiamenti si sono poi riflessi nel mix di generazione: il peso del gas è cresciuto da 97 TWh nel 2000 a 140 nel 2017 (con un picco di 173 TWh nel 2008), mentre quello dei prodotti petroliferi è letteralmente crollato da 86 TWh a 4 TWh nello stesso periodo. Le rinnovabili diverse dall'idroelettrico sono cresciute da una percentuale irrisoria a quasi il 20% del totale. Anche dal punto di vista della struttura del mercato si è assistito a mutamenti di vasta portata. La quota di mercato detenuta da Enel nella generazione elettrica si è ridotta dal 71% del 1999 al 24% attuale. Nel mercato finale della vendita sono oggi attivi in Italia quasi 400 operatori, anche se la concentrazione rimane elevata. Gli investimenti effettuati nella distribuzione e trasmissione, invece, hanno prodotto una riduzione della durata e del numero delle interruzioni di oltre due terzi. Quasi l'intera popolazione è dotata di un contatore elettronico, che aumenta la precisione e la tempestività di rilevazione del dato di consumo migliorando la qualità del servizio di fatturazione e rendendo più affidabile ed efficiente il processo di cambio fornitore. L'apertura del mercato è un processo in continuo divenire, date le continue sfide poste dalla transizione energetica e dagli obiettivi ambientali europei: le problematiche emerse riguardano per lo più l'efficiente integrazione delle fonti rinnovabili nei mercati dell'energia, assicurando in ogni momento la sicurezza e il bilanciamento del sistema [1] [2].

1.2 La Borsa Elettrica

Il settore elettrico in Italia, in seguito al processo di liberalizzazione, può essere articolato in quattro attività principali: generazione, trasmissione, distribuzione e vendita ai clienti finali. Secondo i dati statistici riportati da Terna, a fine 2019 il parco di generazione italiano ha visto una sostanziale stabilità per quanto riguarda la generazione termoelettrica rispetto all'anno precedente, mentre l'energia prodotta da fonti rinnovabili ha subito un incremento fino ad arrivare a coprire il 46,5% del totale installato in Italia. Questo trend crescente dovrà accentuarsi ulteriormente nei prossimi anni, per permettere il passaggio dall'utilizzo di fonti energetiche non rinnovabili a fonti rinnovabili e rendere possibile, quindi, lo sviluppo di economie sostenibili. Un ruolo fondamentale in questa transizione è assunto da Terna, che in Italia ricopre il ruolo di TSO (*Transmission System Operator*) e ISO (*Independent System Operator*), in regime di monopolio e sulla base di concessione governativa. Le attività svolte da Terna riguardano la pianificazione, lo sviluppo e la manutenzione della rete di trasmissione nazionale (RTN), nonché la gestione dei flussi di energia elettrica che vi transitano e il bilanciamento costante tra offerta e domanda di energia. Per garantire concorrenza all'interno del mercato, Terna non opera nell'ambito della distribuzione e della vendita dell'energia elettrica: a occuparsi della distribuzione, ovvero della gestione delle cabine primarie e secondarie, delle linee di media e bassa tensione e degli apparecchi che misurano l'energia, è infatti E-Distribuzione, azienda presente sul suolo italiano a partire dall'1 Ottobre 1999. Un'altra azienda importante nel settore energetico e che deve essere qui citata è il GSE (Gestore dei Servizi Elettrici), una società per azioni italiana, interamente controllata dal Ministero dell'Economia e delle Finanze, che ricopre un ruolo centrale nell'incentivazione e nello sviluppo delle fonti rinnovabili in Italia. Essa è, inoltre, responsabile dell'attuazione dei meccanismi di promozione dell'efficienza energetica e svolge attività di informazione per promuovere la cultura dell'uso dell'energia compatibile e sostenibile con le esigenze dell'ambiente. Per quanto riguarda invece il mercato dell'energia elettrica, nel 2004 venne istituita la Borsa elettrica, strumento che permette a produttori, venditori e acquirenti di negoziare i contratti di acquisto e di vendita dell'energia. La Borsa elettrica è organizzata e gestita da un soggetto pubblico creato appositamente, il GME (Gestore dei Mercati Energetici), con la funzione di promuovere la trasparenza e la competitività del mercato all'ingrosso, operando secondo un criterio di neutralità. La presenza di una piattaforma

telematica regolata, accessibile grazie ad apposite credenziali e certificati digitali rilasciati dal GME, garantisce la sicurezza della negoziazione, la minimizzazione dei costi della transazione stessa e una maggiore efficienza produttiva, poiché la copertura del fabbisogno di energia elettrica avviene con le risorse più efficienti in termini di costo. Oltre alla presenza di un mercato spot rimane, come avveniva già prima del 2004, la stipula di contratti bilaterali, che migliorano l'efficienza delle transazioni e rendono più completa l'offerta del mercato. Il mercato elettrico in Italia (IPEX – *Italian Power Exchange*) si suddivide in mercati dell'energia organizzati e gestiti dal GME e Mercato del Servizio di Dispacciamento (MSD). I mercati in gestione al GME si differenziano in Mercato del Giorno Prima (MGP) e Mercato Infragiornaliero (MI) [4].

1.2.1 Il Mercato del Giorno Prima

Il MGP, al quale partecipano i produttori, i grossisti, l'Acquirente Unico (AU)¹ e il Gestore della Rete di Trasmissione Nazionale (GRTN), si apre alle ore 8:00 del nono giorno precedente il giorno di consegna e si chiude alle ore 9:00 del giorno precedente il giorno di consegna. Durante questo periodo gli operatori possono presentare offerte. Il GME seleziona le offerte secondo un meccanismo d'asta implicita non discriminatoria seguendo un criterio di merito economico. In particolare, le offerte di vendita vengono ordinate per prezzo crescente, a partire da quelle con prezzo più basso fino a quelle con prezzo più alto, mentre le offerte di acquisto per prezzo decrescente, a partire da quelle senza indicazione di prezzo fino a quelle con prezzo più basso. Gli obiettivi dell'algorithmo di risoluzione del mercato sono infatti:

- soddisfare la domanda al minimo costo ottenibile;
- massimizzare il valore netto delle transazioni;
- rispettare i vincoli di trasporto tra zone geografiche e/o virtuali.

¹ L'Acquirente Unico S.p.A. è la società per azioni del gruppo Gestore dei servizi energetici (GSE S.p.A.). Esso è il garante della fornitura di energia elettrica per i clienti che ancora non hanno scelto un fornitore sul libero mercato: il compito di Acquirente Unico è quello di acquistare energia elettrica alle condizioni più favorevoli sul mercato e di cederla alle imprese di vendita al dettaglio per rifornire gli utenti domestici e le piccole imprese che non acquistano sul mercato libero. All'AU sono state poi affidate altre competenze, per meglio tutelare i consumatori e per favorire il processo di liberalizzazione del mercato elettrico e del gas [5].

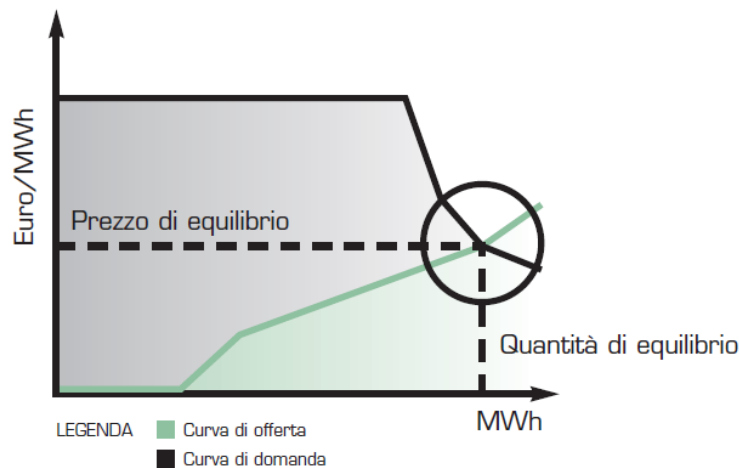


Figura 1.1 Equilibrio tra domanda e offerta per la determinazione del prezzo

Per garantire il rispetto dei vincoli di transito, il GRTN comunica al GME, almeno sessanta minuti prima della chiusura della seduta del MGP, i limiti ammissibili di transiti orari di energia tra le zone geografiche e dalle zone di interconnessione con l'estero, la stima della domanda oraria per zona geografica e i programmi di utilizzo delle unità di produzione CIP6². Le offerte accettate, in presenza di almeno un transito tra zone geografiche saturato, pagano o ricevono un prezzo di equilibrio definito attraverso un algoritmo implementato dal GME che prevede, a fronte di prezzi di vendita differenziati per zona, l'applicazione di un prezzo unico di acquisto su base nazionale (PUN), pari alla media dei prezzi di vendita zonali ponderati per i consumi zonali. Ai fini della determinazione della quantità di equilibrio vengono considerate anche le quantità corrispondenti ai programmi di immissione e prelievo, comunicati dagli operatori in esecuzione di contratti di compravendita conclusi al di fuori del sistema delle offerte (contratti bilaterali) e valutati

² Il CIP6 è una delibera del Comitato interministeriale dei prezzi adottata il 29 aprile 1992 a seguito della legge n. 9 del 1991, con cui sono stabiliti prezzi incentivati per l'energia elettrica prodotta con impianti alimentati da fonti rinnovabili e "assimilate". La dizione "assimilate" fu aggiunta alla previsione originaria in sede di approvazione del provvedimento per includere fonti di vario tipo, non previste espressamente dalla normativa europea in materia. In conseguenza della delibera "CIP6", chi produce energia elettrica da fonti rinnovabili o assimilate ha diritto a rivenderla al Gestore dei Servizi Energetici (GSE) a un prezzo superiore a quello di mercato [6].

rispettivamente alla stregua di offerte di vendita a prezzo zero e di acquisto senza indicazione di prezzo.

1.2.2 Il Mercato Infra-giornaliero e il Mercato dei Servizi di Dispacciamento

Il Mercato Infra-giornaliero (MI), introdotto con la Legge 2/09 del Ministero dello Sviluppo Economico ha sostituito il preesistente Mercato di Aggiustamento e permette a produttori, grossisti e clienti idonei abilitati di modificare i programmi di immissione/prelievo determinati dal MGP. Grazie alla suddivisione in sette sessioni durante l'arco della giornata, il MI permette di scambiare rapidamente quantità di energia elettrica che saranno consumate nello stesso giorno. Le sessioni sono organizzate nella forma di aste implicite di energia (lo stesso meccanismo di funzionamento del MGP) con orari di chiusura diversi ed in successione; tenendo conto di informazioni più aggiornate circa lo stato dei propri impianti di produzione ed il fabbisogno di energia per il giorno successivo, i produttori possono effettuare un miglior controllo degli impianti stessi e gli acquirenti possono aggiornare i programmi di prelievo delle unità di consumo. Se la correzione mediante il MI non dovesse essere sufficiente o se non fosse possibile coprire il carico residuo con prontezza, si deve ricorrere al Mercato dei Servizi di Dispacciamento, che in pochi secondi compensa, se necessario, eventuali squilibri della rete mediante un'energia di regolazione positiva o negativa. Il MSD è lo strumento attraverso il quale Terna si approvvigiona delle risorse necessarie alla gestione e al controllo del sistema. Nello specifico gli obiettivi sono:

- verificare in ciascuna ora e in ciascuna zona la presenza di opportune bande di regolazione per il giorno successivo ed eventualmente modificare gli esiti di MGP e di MI in modo da assicurarle;
- verificare l'assenza previsionale di congestioni intrazonali ed eventualmente modificare gli esiti di MGP così da risolverle;
- garantire il bilanciamento in tempo reale e definire i possibili prezzi dell'energia di bilanciamento a cui ricorrere per contingenze di parco e di rete.

Il Mercato dei Servizi di Dispacciamento si articola in due fasi: una prima fase di programmazione (MSD ex-ante), finalizzata all'acquisto dei servizi per la risoluzione delle congestioni a valle del MI e per l'approvvigionamento di risorse necessarie alla regolazione secondaria di frequenza; la seconda fase è infra-giornaliera e consiste nello scambio, da parte di Terna, dell'energia necessaria per bilanciare immissioni e prelievi a fronte di scostamenti dei produttori e consumatori dai rispettivi programmi [9].

Capitolo 2 - Il prezzo dell'energia elettrica

2.1 Caratteristiche del prezzo dell'energia elettrica

Il prezzo spot dell'energia elettrica presenta delle caratteristiche statistico-finanziarie comuni a tutti i mercati elettrici. Poiché tali caratteristiche sono del tutto particolari e differenti rispetto a quelle di prezzi di altre *commodity*, si rende qui necessaria un'analisi più approfondita. L'energia elettrica, rispetto ad altre materie prime quali acqua, gas, ferro, carbone o petrolio, è un bene che non si può immagazzinare: al giorno d'oggi, infatti, lo sviluppo dei dispositivi di accumulo non ha ancora raggiunto la piena maturità a causa di limiti tecnici e di costo. Le grandi società di produzione e trasmissione di energia elettrica devono dunque regolare la propria attività in modo tale da garantire l'equilibrio tra domanda e offerta, poiché l'energia prodotta in un determinato istante viene trasmessa nell'istante stesso. Il prezzo spot è quindi molto sensibile ai volumi richiesti e bisogna sempre tener presente che la riserva di energia elettrica, da utilizzare nel caso di improvviso aumento della domanda, è limitata. Un'altra caratteristica fondamentale del prezzo dell'energia è la periodicità, collegata strettamente alla variazione della domanda su base giornaliera, settimanale e mensile. In figura 2.1 è riportato a titolo esemplificativo l'andamento del prezzo in un giorno festivo, domenica 4 Ottobre 2020, e in un giorno feriale, lunedì 5 Ottobre 2020. Nei giorni feriali, come si nota nella seconda immagine, il prezzo dell'energia cresce nelle ore diurne raggiungendo un picco tra le ore 9 e 10 del mattino, orario di punta per lo svolgimento delle attività industriali. Nella fascia oraria 12-14, tipicamente dedicata alla pausa pranzo e durante la quale la maggior parte delle attività sono ferme, si ha un abbassamento del prezzo in conseguenza della diminuzione della domanda. Successivamente, il prezzo ricomincia a crescere raggiungendo un picco massimo tra le 18 e le 20, intervallo orario in cui vi è un aumento dei consumi domestici e, a seconda delle stagioni, gli impianti fotovoltaici smettono di produrre energia. Per far fronte al fabbisogno di quel momento è necessario, quindi, ricorrere alle unità di produzione più costose, ad esempio le centrali a gas che hanno dei costi fissi molto elevati; come è facile intuire, un maggiore costo di produzione dell'energia ha come conseguenza un maggior prezzo dell'energia sul mercato. Dopo le ore 21 e fino alle ore 8 del giorno successivo, il prezzo scende, poiché i consumi nelle ore notturne si limitano a quelli delle attività industriali a ciclo continuo. Nei giorni festivi, come si

vede nella prima immagine, il prezzo dell'energia è generalmente più basso. I valori sono simili a quelli dei giorni feriali solo per le fasce notturne, in cui i consumi non cambiano, mentre nelle restanti ore diurne si può notare che il prezzo è più basso e il picco diurno è raggiunto più tardi, verso le 11, poiché i ritmi degli utenti sono diversi. Il picco serale invece viene raggiunto sempre tra le 18 e le 20, ma il valore rimane comunque più basso perché la domanda di energia è inferiore.

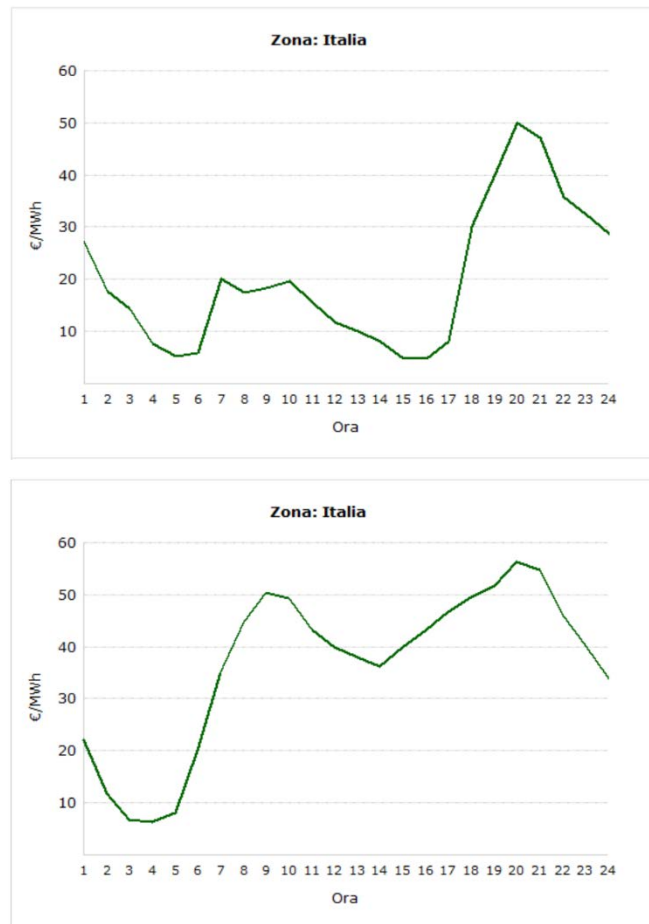


Figura 2.1 Andamento del prezzo dell'energia in Italia nei giorni Domenica 4 Ottobre 2020/ Lunedì 5 Ottobre 2020

Per maggiore completezza in figura 2.2 viene riportato l'andamento della domanda in Italia negli stessi giorni appena analizzati; in particolare viene rappresentato il confronto tra fabbisogno di energia elettrica previsto (linea rossa) e realizzato (linea nera). Dalla comparazione tra curva di prezzo e curva di domanda emerge, come detto in precedenza, che all'aumentare della domanda di energia corrisponde un aumento del prezzo dell'energia stessa.

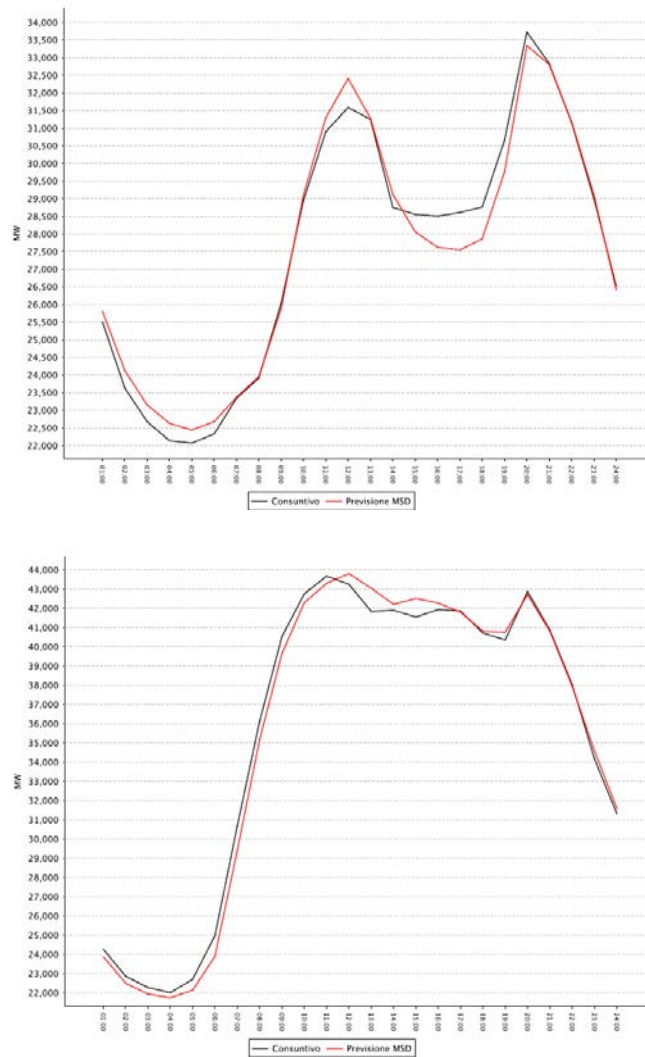


Figura 2.2 Andamento della domanda reale e prevista in Italia nei giorni Domenica 4 Ottobre 2020 e Lunedì 5 Ottobre 2020

La periodicità del prezzo dell'energia si ha, oltre che su base giornaliera, anche su base settimanale: generalmente i consumi di energia elettrica sono molto più elevati nei giorni feriali, per via del funzionamento delle attività industriali, mentre nei giorni festivi la domanda si abbassa poiché gli unici a consumare sono gli utenti domestici e le aziende a ciclo continuo. Questo si traduce in un prezzo spot più elevato durante i giorni feriali e un prezzo inferiore nei giorni festivi. A prova di ciò, in figura 2.3 viene rappresentato l'andamento settimanale del prezzo nelle diverse zone del mercato elettrico italiano, nella settimana del 5 Ottobre 2020.

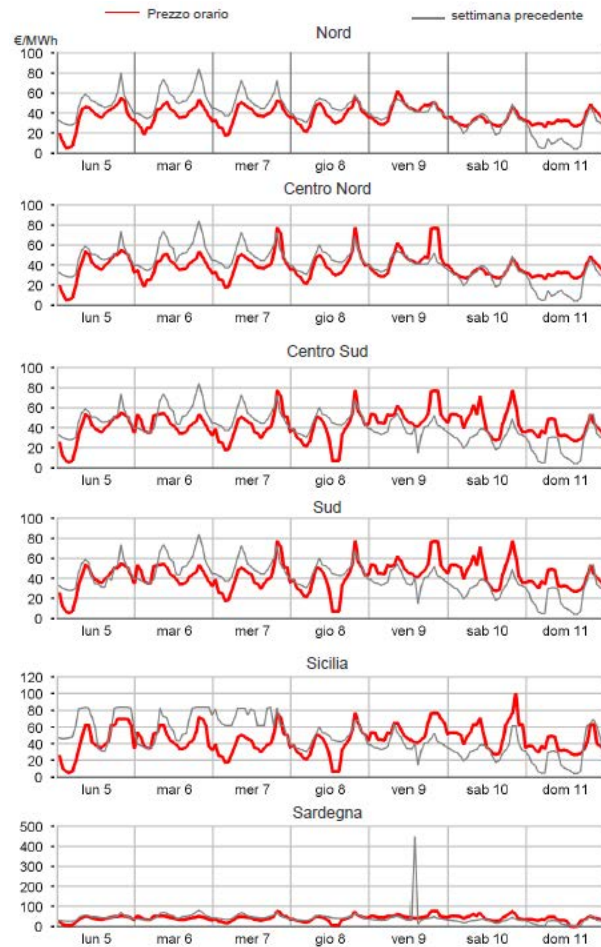


Figura 2.3 Andamento settimanale del prezzo nelle sei zone del mercato elettrico italiano nella settimana del 5 Ottobre 2020

Infine, si può riconoscere una periodicità del prezzo anche a livello annuale: normalmente l'energia elettrica registra un prezzo maggiore nei mesi invernali, sia a causa del numero inferiore di ore di luce che determina un maggior consumo di energia elettrica per l'illuminazione artificiale, sia per il riscaldamento, tramite pompe di calore o stufette elettriche. I picchi più alti di prezzo si registrano però nei mesi estivi, in luglio e agosto, periodi in cui si raggiungono spesso temperature elevate, che rendono indispensabile il ricorso ai condizionatori per raffreddare l'aria. In primavera e nei primi mesi autunnali, quando le giornate sono ancora sufficientemente lunghe e le temperature non sono né troppo calde né troppo fredde, il prezzo scende perché la richiesta di energia elettrica è più bassa. Se si osserva l'andamento del prezzo, ad esempio su base annuale (figura 2.4), si può notare la presenza di quelli che in letteratura vengono definiti “*jumps*” o “*spikes*”. Questi bruschi ed improvvisi cambiamenti nei prezzi sono una delle principali caratteristiche dei mercati elettrici e,

anche se nel linguaggio comune sono spesso utilizzati come sinonimi, in realtà sono differenti: entrambi prevedono un repentino cambiamento nel prezzo, sia verso l'alto che verso il basso, ma i *jumps* sono caratterizzati da un assestamento del prezzo attorno al nuovo livello, mentre gli *spikes* prevedono un ritorno immediato al livello di prezzo precedente. Alla base di queste variazioni possono essere individuate più cause: innanzitutto, come già evidenziato, un elemento di forte impatto sul prezzo dell'energia sono le condizioni meteorologiche e, in particolare, gli improvvisi sbalzi di temperatura che determinano un brusco innalzamento della domanda o una diminuzione della produzione, andando così ad influenzare il prezzo spot; in secondo luogo, un altro fattore che incide sul prezzo sono le crisi operative, ovvero la temporanea indisponibilità degli impianti a causa di malfunzionamenti o guasti improvvisi, con la conseguente necessità di ricorrere ad impianti più costosi o, addirittura, l'impossibilità di fornire la quantità di energia richiesta.

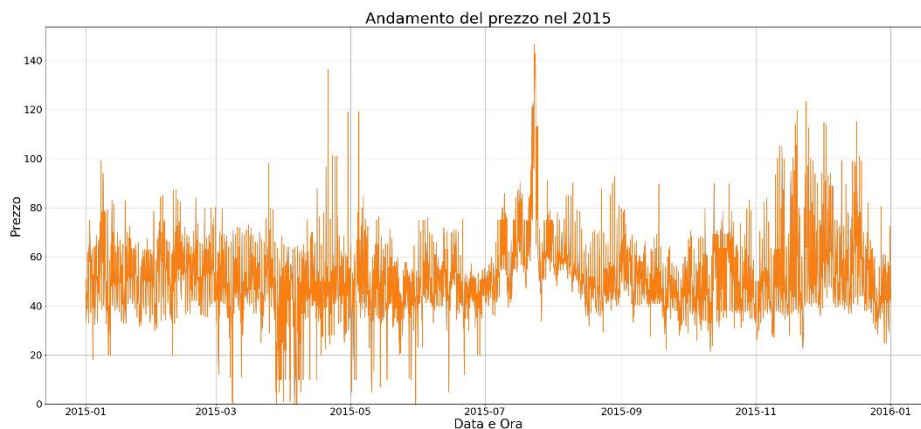


Figura 2.4 Andamento del prezzo dell'energia in Italia nel 2015

Un ulteriore fattore da tenere in considerazione sono i vincoli imposti dalla rete di distribuzione e trasmissione lungo la quale “viaggia” l'energia elettrica. Ogni linea di questa rete è caratterizzata da una capacità o da una quantità massima di elettricità che è in grado di trasportare in un determinato momento. Una volta raggiunto questo limite, il costo marginale di trasmissione, che corrisponde al costo di un'unità di energia trasmessa in più, diventa infinito. Questo implica, ed è ciò che spesso accade, che sezioni del mercato elettrico si isolino dal resto del mercato e che i generatori nella sezione isolata godano di una maggiore influenza sul prezzo dell'energia. Come è

facile intuire, quando ciò accade, il prezzo cresce rapidamente, poiché i produttori sfruttano a loro vantaggio la nuova posizione acquisita. In conclusione, si può quindi affermare che i prezzi dell'energia elettrica siano caratterizzati da una volatilità molto maggiore rispetto a quella di altre attività finanziarie e dei prezzi di altre *commodities*: la dipendenza da così tanti fattori, infatti, rende il prezzo variabile in modo accentuato e improvviso, ed è proprio per questa ragione che fare una previsione di tale grandezza può risultare difficoltoso [10] [11].

2.2 Tecniche di previsione del prezzo dell'energia elettrica: stato dell'arte

La previsione del prezzo dell'energia elettrica gioca un ruolo essenziale nel funzionamento del sistema elettrico. Una previsione accurata è infatti fondamentale sia dal punto di vista dei produttori, che gestiscono i loro impianti al fine di massimizzare i profitti, sia dal punto di vista dei consumatori, il cui obiettivo è di minimizzare i costi dell'energia acquistata. Gli approcci di previsione del prezzo dell'energia elettrica sono molteplici e si possono classificare in diverse categorie [12].

2.2.1 Modelli di simulazione

I modelli di simulazione riproducono le operazioni svolte dai gruppi di generazione e dai sistemi di dispacciamento, tenendo conto delle richieste operative e dei vincoli di trasmissione; l'algoritmo *Market Assessment and Portfolio Strategies* (MAPS), sviluppato dall'azienda americana GE Power³, e il software UPLAN, sviluppato invece dall'azienda LCG Consulting⁴, sono solo due degli esempi che possono essere citati. Tali modelli solitamente non prendono in considerazione le strategie di offerta dei venditori e i comportamenti adottati dagli acquirenti; per considerare anche queste variabili, quindi, è stato proposto il modello SPCM, ovvero *Strategic Production Cost Model*. A differenza dei modelli di equilibrio presentati nel paragrafo successivo, l'SPCM è caratterizzato da una velocità di calcolo maggiore ed è adatto alle previsioni online. I modelli di simulazione possono fornire previsioni accurate solamente basandosi su informazioni precise

³ GE Power (conosciuta formalmente come GE Energy) è un'azienda americana di proprietà di General Electric. Nel 2018 gli impianti posseduti da General Electric hanno prodotto un terzo dell'energia elettrica mondiale.

⁴ LCG Consulting è un'azienda americana, con sede nella Silicon Valley e 20 anni di esperienza nel settore della modellizzazione del mercato competitivo dell'energia elettrica.

riguardanti il sistema di potenza; tali informazioni potrebbero essere difficili da reperire o potrebbero richiedere un'elevata complessità computazionale.

2.2.2 Modelli Multi-Agent

La teoria dei giochi è spesso utilizzata per modellizzare le strategie di offerta dei partecipanti al mercato e trovare quindi la soluzione al “gioco”. Si tratta di una disciplina della matematica applicata che studia e analizza le decisioni individuali di un soggetto, finalizzate al massimo guadagno, in situazioni di conflitto o interazione strategica con altri soggetti rivali (due o più). In tali situazioni le decisioni di un soggetto possono influire sui risultati conseguibili dagli altri e viceversa secondo un meccanismo di retroazione, ricercando soluzioni competitive o cooperative tramite modelli che, in particolare nel contesto economico, si riferiscono al caso in cui due o più aziende interagiscono in concorrenza tra loro [13]. Questo approccio può essere visto come una generalizzazione del modello di costo modificato considerando le strategie di offerta. In tal modo si può prevedere il prezzo dell'energia elettrica senza utilizzare i dati storici e tenendo conto solamente delle informazioni sui costi di approvvigionamento, le concentrazioni di mercato, ovvero il numero di imprese operanti nel mercato dell'energia, la distribuzione delle quote di mercato e le dimensioni delle imprese (quanto maggiore è il grado di concentrazione di un mercato, tanto più è grande il potere di mercato delle imprese leader ossia di quelle che detengono le quote di mercato più grandi). Tra i diversi modelli di equilibrio citati in letteratura, i più noti sono quelli basati sull'equilibrio di Nash: nel 1950 John Nash dimostrò che, sotto certe condizioni, esiste sempre una situazione di equilibrio, che si ottiene quando ciascun individuo (nel caso del mercato dell'energia l'azienda acquirente o venditrice) che partecipa al gioco sceglie la sua mossa strategica in modo da massimizzare il suo guadagno, sotto la congettura che il comportamento dei rivali non varierà a motivo della sua scelta. Questo tipo di modelli sono ampiamente utilizzati per l'analisi del comportamento dei partecipanti al mercato, ma più nell'ottica di un risultato qualitativo che quantitativo; dunque, a causa della scarsa accuratezza del valore previsto, sono utili per lo più per la previsione del *trend* di mercato.

2.2.3 Modelli statistici

Quelli statistici rientrano tra i modelli più largamente utilizzati per la previsione del prezzo dell'energia elettrica; se ne farà ora un breve elenco:

- Il metodo “*Similar day*” stima il prezzo corrente dell'energia sfruttando il valore medio del prezzo di un “giorno simile” precedente. Per valutare la somiglianza tra il giorno oggetto della previsione e il “giorno simile” preso a riferimento, si utilizza la norma euclidea; più piccolo è il valore di tale parametro più i giorni considerati saranno simili tra loro. Questo metodo è caratterizzato da un'implementazione semplice e presenta buone performance quando si considerano giorni con le stesse caratteristiche; dimostra però scarsa accuratezza quando il prezzo dell'energia subisce ampie variazioni.
- Il metodo *ESM*, ovvero *Exponential Smoothing Method*, è uno dei più utilizzati per l'elaborazione di serie storiche di dati. Invece di associare ad ogni dato passato lo stesso peso e di considerare una finestra temporale di osservazione fissa (come nella media mobile semplice), viene utilizzata una particolare funzione per assegnare ad ogni osservazione un peso che decresce esponenzialmente nel tempo. Supponendo che il dato reale sia indicato con x_t , che il suo valore trasformato sia s_t , che la sequenza dei dati inizi al tempo $t=0$ e che $x_0=s_0$, allora la funzione applicata al valore x_t assume la seguente espressione:

$$s_i = \alpha x_i + (1 - \alpha)s_{i-1} = \alpha \sum_{j=0}^i (1 - \alpha)^j x_{i-j} \quad , \quad i > 0$$

Dove α rappresenta il fattore “appianante” ed è compreso tra i valori 0 e 1. Il risultato dell'espressione s_i è dunque una media pesata del dato corrente x_i e del valore trasformato precedente s_{i-1} ; l'equazione precedente mette in evidenza come s_i dipenda da tutti i dati precedenti (x_{i-j}) in formato esponenziale.

- Il modello *GAM* (*Generale Additive Model*), introdotto nel 1990 da Hastie e Tibshirani, è un modello lineare generalizzato in cui l'output Y_t è definito come la somma di funzioni “appiananti” di diverse covariate $X_{t,k}$ ($X_t=(X_{t,1}; X_{t,2}; \dots X_{t,N})$, ovvero variabili che non

rientrano nel vero e proprio meccanismo causale dell'output ma che ne spiegano parte della variabilità. L'output assume dunque la seguente espressione:

$$Y_t = \mu(X_t) + \varepsilon_t$$

Dove μ rappresenta la funzione da stimare e ε_t è una variabile causale caratterizzata da valore medio nullo e appartenente ad una classe di variabili esponenziali (distribuzione di Poisson o Gamma). Una funzione di collegamento, ad esempio la funzione identità o la logaritmica, mette in relazione il valore previsto di Y con le diverse covariate, e presenta la seguente struttura:

$$g(\mu(X_t)) = f_1(X_{t,1}) + f_2(X_{t,2}) + \dots + f_N(X_{t,N})$$

Dove f_i sono le funzioni che determinano una decrescita esponenziale nel tempo delle covariate $X_{t,k}$ ($k=1, 2, \dots, N$). Questo metodo solitamente presenta una migliore accuratezza dei modelli lineari convenzionali, soprattutto se applicato a serie caratterizzate da un *trend* costante e sistematico. La presenza di valori anomali o bruschi cambiamenti nel prezzo dell'energia può portare, infatti, ad un rapido calo dell'accuratezza previsionale.

- Il modello di regressione quantile, ideato da Koenker e Bassett nel 1978, a differenza dei metodi classici di regressione in cui l'unica informazione che si ottiene sulla relazione tra Y e il vettore delle covariate X è il modo in cui la media di Y varia al variare di X, presenta il vantaggio di poter stimare l'intera distribuzione dei quantili condizionati della variabile risposta, così da poter studiare l'influenza delle variabili esplicative sulla forma della distribuzione di Y. In altre parole, la stima di un valore (media condizionata) viene sostituita dalla stima di 99 valori (quantili condizionati). Lo svantaggio di questo modello è rappresentato, invece, dalle limitazioni a cui è soggetto nel caso in cui siano presenti covariate tempo invarianti.
- Il modello *Jump-Diffusion* prende in considerazione diverse caratteristiche del prezzo dell'elettricità, tra cui la stagionalità, la presenza di *spikes* e l'estrema volatilità. Il prezzo viene rappresentato dalla seguente espressione:

$$\log(P_t) = f(t) + X_t$$

Dove P_t è il prezzo spot dell'energia elettrica, $f(t)$ e X_t sono, rispettivamente, la parte deterministica e stocastica del modello. La parte deterministica è data dalla somma di funzioni trigonometriche, mentre quella stocastica è modellizzata mediante il processo di Ornstein-Uhlenbeck. Questo tipo di modello è in grado di simulare *spikes* di prezzo molto ampi ma, allo stesso tempo, per poterlo applicare è necessario stimare accuratamente i parametri che lo caratterizzano.

- Tra i modelli più utilizzati per la previsione del prezzo dell'energia rientrano quelli basati sulle serie temporali, tra cui il modello di auto-regressione (AR), quello di media mobile (in inglese *moving average*, MA), il modello auto-regressivo a media mobile (ARMA), il modello auto-regressivo integrato a media mobile (ARIMA) e infine il modello auto-regressivo generalizzato a eteroschedasticità condizionata (GARCH). Il modello AR stabilisce il valore attuale del prezzo modellizzandolo come la somma tra l'errore stocastico e la combinazione lineare dei prezzi storici. Quello MA, invece, definisce l'output come la somma tra il valore medio e la media mobile del rumore nella serie temporale dei prezzi. Nel modello ARIMA, che rappresenta la generalizzazione del modello ARMA, il valore futuro del prezzo è dato dal risultato di una funzione lineare che ha come input i dati storici e gli errori casuali. Questi modelli, incluso il modello GARCH, vengono applicati con successo in molti ambiti, nonostante presentino dei limiti nell'identificare la presenza di schemi non lineari nelle serie storiche di dati. Essi, infatti, riescono a prevedere i cambiamenti del prezzo solamente se questi rientrano nella sequenza stessa di dati, senza considerare l'influenza del mercato o altri fattori ambientali. Inoltre, tali modelli vengono utilizzati con successo solo quando la frequenza dei dati è bassa, ma possono presentare delle problematiche quando avvengono rapide variazioni nei prezzi.

2.2.4 Modelli di intelligenza computazionale

Tra i modelli di intelligenza computazionale vengono classificate le reti neurali artificiali (ANN), che verranno approfondite nel capitolo successivo, la tecnica *Support Vector Machine* (SVM), molto diffusa nell'ambito del *forecast* dei prezzi dell'energia poiché è caratterizzata da accuratezza

nel fare le previsioni e dalla capacità di ben generalizzare dati mai analizzati in precedenza, e i modelli di *deep learning*, che includono le reti neurali ricorrenti (LSTM, GRU, le reti Elman e Jordan, ecc) e le reti neurali convoluzionali. Quest'ultimo approccio è generalmente caratterizzato da un'accuratezza nell'eseguire le previsioni maggiore rispetto ai metodi statistici tradizionali.

2.2.5 Modelli HIM e previsioni combinate

Generalmente quando si ha a che fare con problemi reali, come quello della previsione del prezzo dell'energia elettrica, non esiste un singolo metodo che presenta ottime performance in tutti i casi. Negli ultimi anni quindi si sono diffusi, soprattutto nell'ambito della ricerca, i modelli ibridi; tali modelli combinano due o più metodi già esistenti per riuscire a sfruttare i punti di forza di ognuno, nonostante questa operazione porti necessariamente ad incrementare la complessità dei modelli stessi. Se si parla invece di previsioni combinate, si fa riferimento ad un approccio differente: i risultati di più metodi di previsione vengono integrati tra loro, in modo tale da ottimizzare le performance generali. I vari metodi, infatti, daranno risultati diversi, nonostante le condizioni iniziali siano le stesse. Dovendo combinare i risultati di più modelli, si avrà un conseguente incremento dell'onere computazionale e della complessità nell'ottenere la previsione finale.

Capitolo 3 - Il Machine Learning e il Modello

LSTM

Il *machine learning*, tradotto in italiano con “apprendimento automatico”, è una branca dell’intelligenza artificiale che permette al sistema di “imparare” automaticamente dall’ambiente in cui è inserito e sfruttare tale conoscenza per svolgere determinate azioni o migliorare il processo decisionale. Gli algoritmi di *machine learning* si basano su tecniche statistiche e sono in grado di indentificare pattern nei dati in ingresso, riuscendo a costruire modelli di sistemi complessi senza conoscerne la natura fisica. Quando si parla di *deep learning* si fa invece riferimento a modelli di apprendimento automatico più sofisticati, che non richiedono alcun intervento umano per arrivare al risultato finale. Alla base di questi algoritmi ci sono le reti neurali artificiali, il cui design è ispirato alla rete neurale biologica del cervello umano: l’obiettivo è infatti quello di replicare la stessa struttura logica con cui il cervello elabora i dati provenienti dall’ambiente esterno. Queste tecniche risultano quindi innovative rispetto agli approcci tradizionali e rappresentano, spesso, l’unico approccio possibile in tutte quelle situazioni in cui la scrittura di un algoritmo esplicito risulterebbe impossibile o estremamente sconveniente. In figura 3.1 è riportata schematicamente la differenza tra un approccio di programmazione di tipo tradizionale e quello utilizzato nel *machine learning*. Nei paragrafi seguenti si analizzeranno le diverse tipologie di algoritmi, focalizzandosi in particolare sulle reti neurali LSTM (*Long Short Term Memory*).

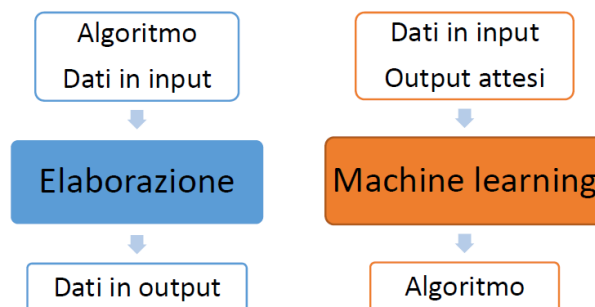


Figura 3.1 Algoritmo tradizionale (sinistra) algoritmo di machine learning (destra)

3.1 Tipologie di algoritmi

Gli algoritmi di machine learning si suddividono in tre categorie, a seconda dei dati che si hanno a disposizione:

- apprendimento supervisionato;
- apprendimento non supervisionato;
- apprendimento rinforzato.

Nel *Supervised Machine Learning* l'obiettivo è costruire una funzione (f) che descriva al meglio la relazione esistente tra i dati di input (x) e i dati di output (y), si conoscono quindi sia i dati in ingresso, sia le corrispondenti uscite. La maggior parte dei dati a disposizione vengono usati per fare il *train* del modello e ricavare una funzione f caratterizzata da un livello di performance adeguato. Tale funzione viene poi applicata a nuovi dati e utilizzata per ottenere le corrispondenti previsioni.

$$y=f(x)$$

Esistono due diverse tipologie di algoritmi supervisionati, quelli che risolvono problemi di classificazione e quelli invece applicabili a problemi di regressione. La differenza tra classificazione e regressione sta nel tipo di variabile in uscita: se la variabile di output è categorica (ad esempio un colore, una specie animale, ...) allora si tratta di un problema di classificazione; se invece l'uscita è un valore reale, ovvero un numero, allora si parla di problema di regressione. Il caso dell'*Unsupervised Machine Learning* si differenzia da quello precedente per la mancanza di dati di output. L'algoritmo deve dunque individuare strutture, distribuzioni o caratteristiche comuni nei dati, per cercare di effettuare ragionamenti e previsioni sugli input successivi. Anche in questo caso i problemi che vengono affrontati con tale tipologia di algoritmi sono di due tipi: i problemi di *clustering*, in cui l'algoritmo riconosce dei gruppi omogenei all'interno dei dati di input, e i problemi di associazione, dove l'obiettivo è riconoscere delle regole che legano tra loro i dati di input. Infine, il *Reinforced Learning* si basa su un algoritmo che esegue un'azione e, se tale azione

viene ritenuta positiva dall'ambiente in cui opera, l'algoritmo viene "ricompensato" in modo tale da incoraggiare i comportamenti positivi e penalizzare invece quelli negativi. Le decisioni vengono prese in modo sequenziale e ad ogni step l'algoritmo viene ricompensato positivamente o negativamente, l'obiettivo è massimizzare la somma delle varie ricompense per determinare il miglior percorso decisionale. Nella figura sottostante viene riportato uno schema riassuntivo dei diversi approcci appena descritti [14].

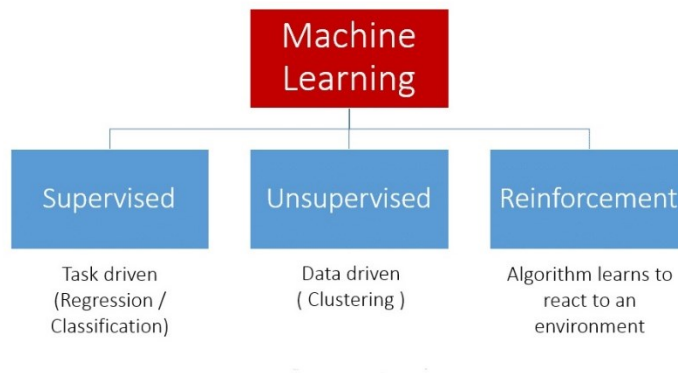


Figura 3.2 Diverse tipologie di machine learning

3.2 Le reti neurali ricorrenti

Al fine di rendere efficace questa trattazione si rende ora necessario un approfondimento delle reti neurali e, in particolare, del modello LSTM, utilizzato in seguito per la previsione del prezzo zonale dell'energia elettrica. Tale modello, infatti, rientra nella categoria delle reti neurali artificiali ricorrenti (RNN, *Recurrent Neural Network*) e viene utilizzato nel campo del *deep learning* per fare previsioni in cui, al fine di ottenere una previsione più accurata, sia necessario tener conto dell'andamento dei dati nel passato. La struttura di una rete neurale artificiale è visivamente schematizzabile con l'immagine riportata in figura 3.3: come già accennato in precedenza, le reti neurali si pongono come obiettivo la riproduzione del comportamento di una rete neurale biologica, sono quindi formate da più strati, uno di input, uno di output e un numero variabile di strati nascosti, ognuno caratterizzato da certo numero di unità, che, idealmente, rappresentano i neuroni. La rete neurale, una volta costruita, viene addestrata: durante questa fase vengono

utilizzati i dati a disposizione per permettere alla rete di stabilire i pesi associati alle unità di ciascuno strato.

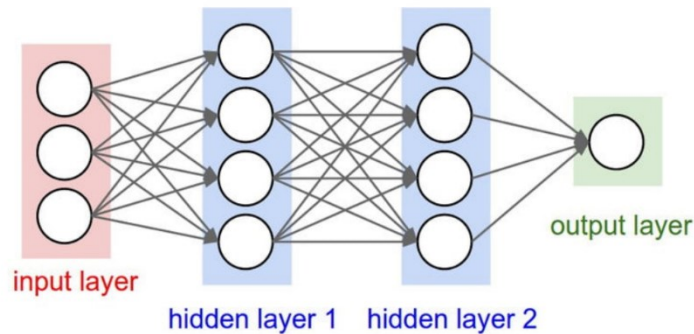


Figura 3.3 Struttura rete neurale artificiale

Quando si parla di reti neurali ricorrenti si fa riferimento ad una particolare tipologia di reti neurali artificiali: esse usano informazioni di tipo sequenziale e vengono definite “ricorrenti” in quanto ripetono la medesima azione su ogni elemento della sequenza; sono inoltre capaci, a differenza delle reti neurali tradizionali, di avere memoria dei dati precedentemente analizzati, poiché l’output in un determinato istante dipende dalle computazioni pregresse. La capacità di una rete neurale ricorrente di tener conto dell’andamento dei dati nel passato è dettata dalla presenza di cicli all’interno della rete stessa. Tale rete può quindi essere pensata come l’insieme di più copie della medesima rete, ognuna delle quali in grado di passare informazioni alla successiva.

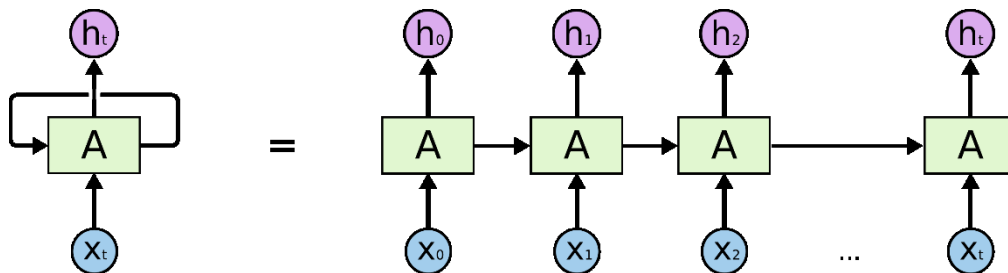


Figura 3.4 Rete Neurale Ricorrente

In figura 3.4 è rappresentata una rete neurale ricorrente “srotolata”, in modo tale da metterne in evidenza il funzionamento ciclico: come si può facilmente intuire, la struttura di questi algoritmi è

adatta all'analisi di serie temporali o dati sequenziali. Le reti neurali ricorrenti riescono a tener conto solo di un numero limitato di step passati, a causa del problema della "scomparsa del gradiente". Questo problema, noto in inglese come *Vanishing Gradients Problem* [15], è legato al modo in cui questi algoritmi vengono addestrati: ad ogni step della fase di addestramento viene calcolato l'errore commesso nella previsione e tale errore viene poi utilizzato per stimare il gradiente, il quale a sua volta serve ad aggiornare il peso associato ad ogni neurone della rete in modo tale da commettere un errore minore nello step successivo. Il gradiente dell'errore viene propagato a ritroso lungo tutta la rete, a partire dallo strato di output fino a raggiungere quello di input. In generale, per aumentare la capacità di una rete di apprendere da una grande varietà di dati e di rappresentare efficientemente relazioni più complesse, si aumenta il numero di strati nascosti; il problema delle reti caratterizzate da un elevato numero di strati nascosti è la drastica diminuzione del gradiente man mano che viene propagato a ritroso nella rete stessa. L'errore, una volta raggiunti gli strati prossimi a quello di input, può essere così piccolo da perdere, in definitiva, qualsiasi effetto correttivo. Lo stesso problema, ma con caratteri opposti, si verifica nel caso in cui ci sia un'esplosione del gradiente (*Exploding Gradient*), che prevede un aumento esponenziale del gradiente man mano che si propaga nella rete. Il problema dell'instabilità del gradiente viene risolto con le reti LSTM (*Long Short Term Memory*).

3.3 Il modello "Long Short Term Memory"

Le reti neurali LSTM, introdotte nel 1997 da Hochreiter & Schmidhuber, sono particolari reti neurali ricorrenti in grado di riconoscere relazioni tra dati anche molto "distanti" l'uno dall'altro e rappresentano, dunque, la soluzione migliore per risolvere problemi caratterizzati da dipendenze temporali sul lungo periodo.

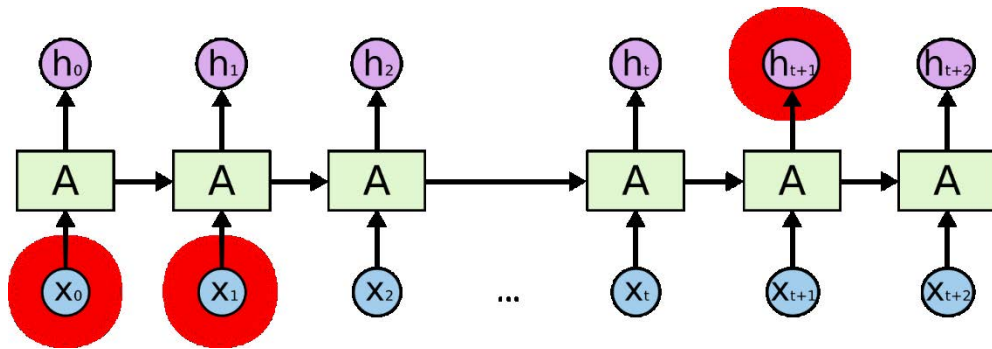


Figura 3.5 Rete Neurale LSTM

Come già accennato in precedenza, tutte le reti neurali ricorrenti sono costituite da più moduli ripetitivi e, nel caso di reti tradizionali, tali moduli hanno una struttura molto semplice. La figura 3.6 riporta il confronto tra una rete neurale tradizionale e una LSTM: nel primo caso si ha una catena formata da più moduli, a loro volta costituiti da un singolo strato che esegue l'operazione di tangente iperbolica. La rete LSTM presenta la stessa struttura a catena, ma in questo caso i singoli moduli sono costituiti da quattro diversi strati che interagiscono tra loro in modo specifico.

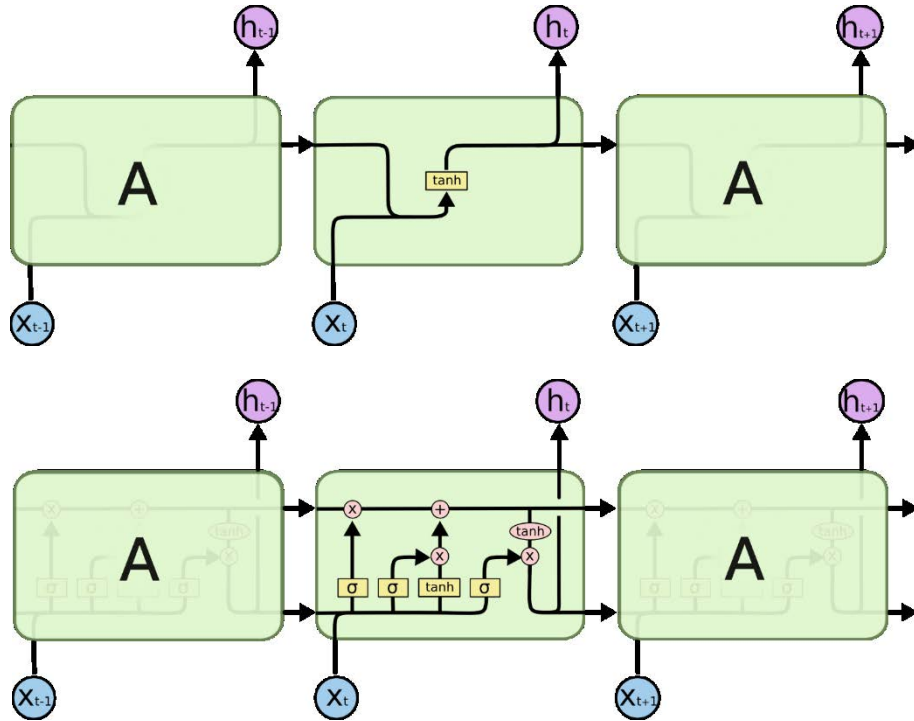


Figura 3.6 Struttura Rete Neurale Tradizionale e Rete Neurale LSTM

L'aspetto fondamentale di una rete LSTM è lo stato della cella, che può essere pensato come un nastro trasportatore che corre lungo l'intera catena; se l'informazione in ingresso alla cella non subisce alcuna modifica da parte della cella stessa, si avrà la coincidenza tra input e output, ed è per questa ragione che lo stato della cella viene rappresentato graficamente con una linea orizzontale posizionata nella parte superiore della cella stessa.

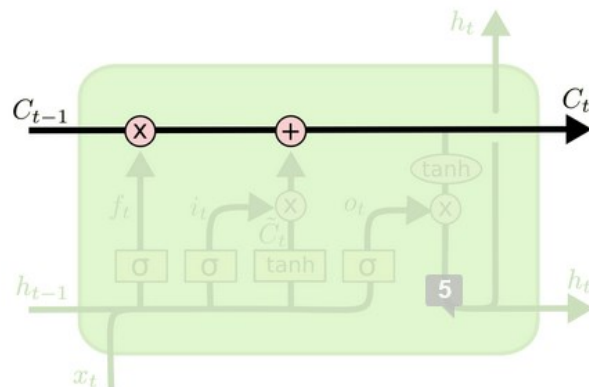


Figura 3.7 Cella di una Rete Neurale LSTM

Questo tipo di rete neurale, sfruttando la presenza di tre distinti ingressi, ha l'abilità di aggiungere o rimuovere informazioni allo stato della cella, riuscendo a farne, quindi, un'attenta regolazione. Gli ingressi presentano la struttura riportata in figura 3.8: uno strato di rete neurale caratterizzato dall'operatore sigma e un'operazione puntuale di moltiplicazione. L'uscita dello strato sigmoideo può variare tra zero e uno, a seconda della componente dell'informazione in input che deve essere lasciata passare (un valore nullo implica a sua volta un valore nullo in uscita, mentre un valore pari ad uno lascia l'input invariato).

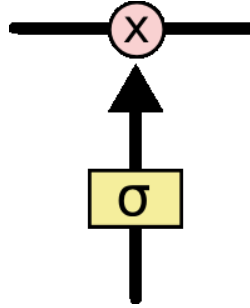
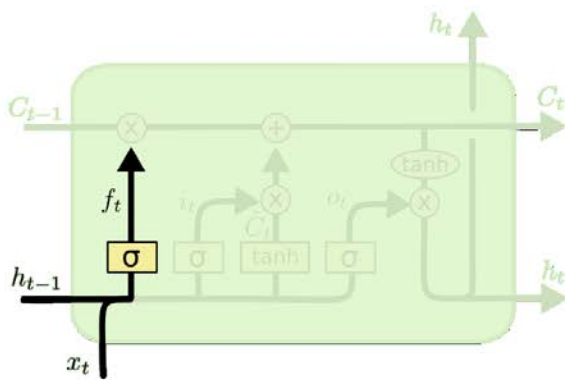


Figura 3.8 Struttura di un "gate" presente in una cella di una Rete Neurale LSTM

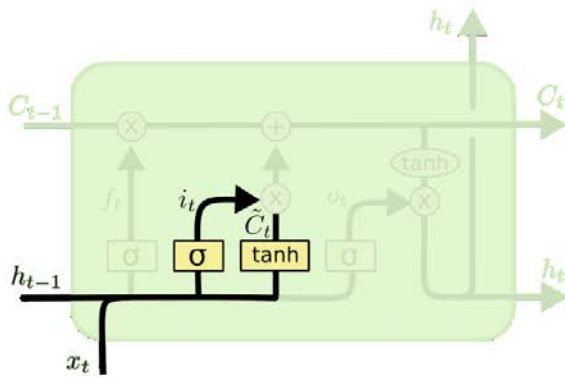
Il primo passo consiste nel decidere quale informazione cancellare dallo stato della cella C_t . Tale decisione viene presa per mezzo di uno strato sigmoideo, chiamato *forget gate layer*, che ha in ingresso le variabili h_{t-1} e x_t (x_t è l'input all'istante t , mentre h_{t-1} è il valore dello stato nascosto all'istante $t-1$) e, in uscita, un numero compreso tra 0 e 1, a seconda di quanto del precedente stato h_{t-1} viene lasciato passare. Tutte le funzioni in uscita dai gate dipendono, inoltre, dai parametri W e b , caratteristici dell'unità LSTM stessa.



$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Figura 3.9 Step 1 Cella LSTM

Lo step successivo, costituito da uno strato con l'operatore sigma e un altro con l'operatore di tangente iperbolica, permette di determinare quale nuova informazione viene immagazzinata nello stato C_t della cella. Il primo strato, chiamato in lingua inglese *input gate layer* decide con quale nuovo valore aggiornare lo stato della cella, mentre il secondo strato crea un vettore di nuovi candidati \tilde{C}_t , che possono essere aggiunti al valore precedentemente determinato.

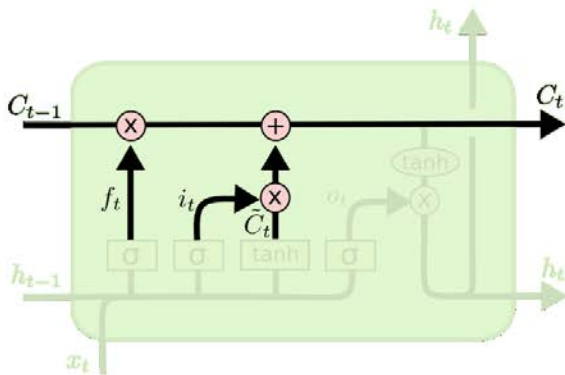


$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Figura 3.10 Step 2 Cella LSTM

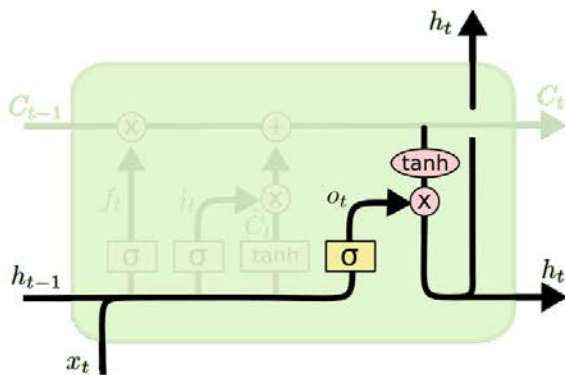
A questo punto è necessario combinare le diverse uscite per aggiornare effettivamente lo stato della cella, passando dal valore C_{t-1} al valore C_t : lo stato C_{t-1} viene moltiplicato per la funzione f_t , in modo tale da eliminare le informazioni non più necessarie, e, a tale valore, si somma il termine dato dalla moltiplicazione tra i_t e \tilde{C}_t . In figura 16 vengono riportati tali passaggi.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Figura 3.11 Step 3 Cella LSTM

Infine, è necessario stabilire quali saranno gli output della cella, i quali si basano sullo stato della cella stessa. Innanzitutto, si utilizza uno strato con un operatore sigma per determinare quale parte dello stato della cella dare in uscita; dopodiché, si fa passare lo stato della cella attraverso uno strato caratterizzato dall'operatore tangente iperbolica (per fare in modo che vari tra i valori -1 e 1) e lo si moltiplica, a sua volta, per l'uscita dell'*output gate layer*, identificato dalla funzione o_t [16] [17].



$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o)$$

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

Figura 3.12 Step 4 Cella LSTM

Nei paragrafi successivi verrà illustrato come tale modello venga implementato attraverso il linguaggio di programmazione Python, andando ad analizzare, in linea generale, le librerie utilizzate e tutte le parti caratteristiche del codice.

3.4 Le librerie Python

Python è un linguaggio di programmazione dalla sintassi semplice e potente, viene utilizzato in svariati ambiti quali lo sviluppo di siti o applicazioni Web, la realizzazione di interfacce grafiche, l'amministrazione di sistema, il calcolo scientifico e numerico, il machine learning, ecc. e, a differenza di altri linguaggi, è completamente gratuito e può essere distribuito senza restrizioni di copyright. Anche se Python è considerato un linguaggio interpretato, i programmi vengono automaticamente compilati in un formato chiamato *bytecode* prima di essere eseguiti. Questo formato è più compatto ed efficiente, e garantisce quindi prestazione elevate. Inoltre, diverse strutture dati, funzioni e moduli di Python sono implementati internamente in C per essere ancora più performanti. Per quanto riguarda la gestione della memoria, Python adotta un meccanismo di *garbage collection*, che si occupa automaticamente dell'allocazione e del rilascio della memoria. Questo consente al programmatore di usare variabili liberamente, senza doversi preoccupare di dichiararle e di allocare e rilasciare spazi di memoria manualmente (cosa che è invece necessaria in linguaggi di più basso livello come il C o il C++). Un'altra caratteristica fondamentale è la ricchezza di librerie: ogni installazione di Python include la *standard library*, cioè una collezione di oltre 200 moduli per svolgere i compiti più disparati, come ad esempio l'interazione con il sistema operativo e il *filesystem*, o la gestione di diversi protocolli. Inoltre, il *Python Package Index*

consente di scaricare e installare migliaia di moduli aggiuntivi creati e mantenuti dalla comunità. Nel modello presentato nel capitolo quattro vengono utilizzate diverse librerie Python, che verranno ora brevemente descritte:

- **Keras.** Keras è una libreria *open source* per l'apprendimento automatico e le reti neurali. È progettata come un'interfaccia a un livello di astrazione superiore di altre librerie simili di più basso livello e supporta come *back-end*⁵ le librerie TensorFlow, Microsoft Cognitive Toolkit (CNTK) e Theano. Venne sviluppata come parte del progetto di ricerca ONEIROS (*Open-ended Neuro-Electronic Intelligent Robot Operating System*), e il suo autore principale è François Chollet, di Google. Nel 2017 il team di Tensorflow decise di supportare questa libreria ufficialmente, infatti, a detta del suo ideatore, Keras era stata pensata come un'interfaccia e non come una libreria *stand-alone*. Essa offre una serie di moduli che permettono di sviluppare reti neurali profonde indipendentemente dal *back-end* utilizzato, con un linguaggio comune e intuitivo [18].
- **Pandas.** Questa libreria viene utilizzata per l'analisi e la manipolazione di dati. Venne creata nel 2008 da Wes McKinney per risolvere problemi di natura finanziaria, ma successivamente si diffuse anche in molti altri ambiti, poiché, in particolare, permette di gestire in modo facile e flessibile serie storiche di dati. Pandas si basa sulle librerie standard del pacchetto SciPy, infatti utilizza Numpy per una gestione più rapida dei vettori, StatsModels per semplificare l'uso di determinate operazioni statistiche e Matplotlib per creare grafici. La libreria Pandas basa tutte le sue funzioni sul Dataframe, ovvero una struttura tabulare bidimensionale costituita da righe e colonne indicizzate (figura 3.13). Alcune delle più importanti funzioni che possono essere svolte da tale libreria sono la manipolazione di dati (gestione di colonne, divisione e organizzazione gerarchica dei dati, *reshaping*, ecc.), le operazioni su serie storiche di dati, la gestione di dati mancanti (eliminazione, *replacing*, interpolazione, ecc.) e visualizzazione semplificata

⁵ Molti programmi per calcolatore sono concettualmente suddivisi in *front end* e *back end*. Generalmente nascosto all'utente, il *back end* è talvolta un programma completo e indipendente che può essere controllato attraverso un altro programma, detto *front end*. Spesso il *front end* è un'interfaccia grafica (GUI) che genera comandi per una più semplice interfaccia a riga di comando. Tutte le attività richieste attraverso l'interazione con gli elementi dell'interfaccia grafica vengono tradotte in linee di comando e indirizzate al sottostante programma di *back end* [19].

dell'andamento dei dati e delle loro caratteristiche mediante istogrammi, grafici o matrici di dispersione [20].

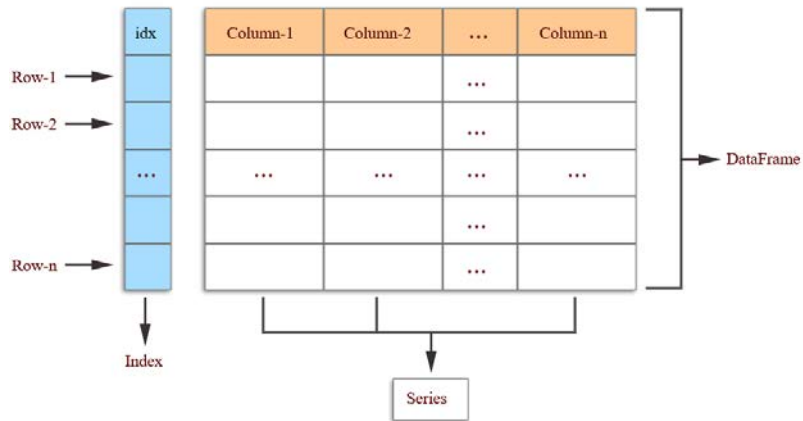


Figura 3.13 Struttura di un DataFrame

- **Numpy.** Il nome di questa libreria è, per esteso, *Numerical Python*; venne sviluppata nel 2005 da Travis Oliphant e permette di svolgere computazioni numeriche in modo facile ed efficiente. Numpy fornisce numerosi vantaggi nella gestione di vettori caratterizzati da un elevato numero di elementi: non si potrebbe infatti raggiungere la stessa velocità operativa se quegli stessi elementi fossero gestiti mediante liste Python. La libreria Numpy offre, inoltre, una grande varietà di funzioni matematiche, che permettono di svolgere computazioni complesse in modo rapido e con poche righe di codice, rendendo quindi il programma più facilmente comprensibile. Alla base di Numpy ci sono gli “ndarray”, ovvero i vettori n-dimensionali costituiti da elementi dello stesso tipo (numeri interi, stringhe, ecc.) [21].
- **Scikit-learn.** Questa libreria *open source* contiene una serie di implementazioni per diversi algoritmi popolari di machine learning; essa si basa su altre librerie quali Scipy, Numpy e Matplotlib.
- **Matplotlib.** Matplotlib è una libreria Python introdotta nel 2002 da John Hunter e utilizzata per la creazione e visualizzazione di grafici 2D. Essa si basa sui vettori Numpy, permettendo di accedere visivamente ad una grande quantità di dati per mezzo di

istogrammi, grafici di dispersione, grafici a barre, ecc. Si tratta di uno strumento molto utile per capire i *trend*, gli schemi e le correlazioni presenti nei dati [22].

3.5 La struttura e il codice del modello LSTM

L'implementazione di un modello LSTM mediante un codice Python prevede più step, che verranno qui di seguito analizzati.

3.5.1 La preparazione dei dati

Alla base di un qualsiasi modello di machine learning ci sono i dati. La natura di quest'ultimi dipende dal tipo di progetto, ma si tratta in genere di serie storiche, testi, immagini, video o audio. Il primo step consiste nella raccolta dei dati stessi, si tratta di un processo fondamentale poiché le performance del modello dipendono per lo più dalla bontà dei dati con cui viene addestrato. Nel caso di una serie storica di dati, una volta reperiti, in formato Excel o csv, essi vengono caricati in Python mediante la funzione Pandas `read_excel('file_name')` o `read_csv('file_name')`, che genera automaticamente un DataFrame. Su tali dati si eseguono poi alcune operazioni: innanzitutto, si imposta la colonna riportante la data e l'ora come indice del DataFrame; prima di eseguire tale operazione è necessario modificare il formato della data e dell'ora in modo tale che sia riconoscibile dal programma [23]. Le funzioni Pandas utilizzate per svolgere tali operazioni sono `to_datetime()` e `set_index()`, come riportato nello script Python n.1. Vengono poi riportate a titolo esemplificativo in figura 3.14 le prime colonne del DataFrame utilizzato nel modello LSTM sviluppato per la previsione del prezzo zonale dell'energia elettrica.

```
1 # importazione dati
2 data_ora = pd.read_excel(r:'C:\Documents\Dati\Dati.xlsx',
                           sheet_name='Foglio1', usecols='A')
3 # impostazione del formato corretto
4 data_ora = pd.to_datetime(data_ora['Data Ora'],
                           format='d%m%Y%\ H%M%S:')
5 # impostazione della colonna data_ora come indice del DataFrame 'dati'
```

```
6 dati.set_index(['Data Ora'], inplace=True)
```

Script 1 Impostazione data e ora come indice del DataFrame

Data Ora	Prezzo	Total Load (MW)	Prezzo Gas	Global Irradiance (W/m2)	Wind Speed (h:10m, m/s)
2015-01-01 00:00:00	51,1	26074,83575	25,056	0	4,118456641
2015-01-01 01:00:00	49	24684,25	25,056	0	3,9611405
2015-01-01 02:00:00	39,1	22995,9995	25,056	0	3,817987424
2015-01-01 03:00:00	35,87	21631	25,056	0	3,7761876
2015-01-01 04:00:00	33,4	20863	25,056	0	3,243402868
2015-01-01 05:00:00	36,57	20916,4995	25,056	0	2,862199932
2015-01-01 06:00:00	39,1	21816,2505	25,056	0	2,541905571
2015-01-01 07:00:00	45,21	22136,50025	25,056	8,404701599	1,909053366
2015-01-01 08:00:00	40	22671	25,056	47,7625933	1,888007607
2015-01-01 09:00:00	40,8	24858,9995	25,056	79,09251242	1,891524824
2015-01-01 10:00:00	38,8	26858,74975	25,056	99,07026576	1,93097194
2015-01-01 11:00:00	38,8	28649,5	25,056	107,527734	2,0968677
2015-01-01 12:00:00	38,85	28883,50075	25,056	106,1986385	2,249470981
2015-01-01 13:00:00	38,8	27036	25,056	93,69401489	2,262313211
2015-01-01 14:00:00	38,85	25842,74925	25,056	70,16826101	2,311415375
2015-01-01 15:00:00	49,95	25738	25,056	35,52908338	2,391361444
2015-01-01 16:00:00	54,69	27249,25075	25,056	0	2,397644735
2015-01-01 17:00:00	61,9	31673	25,056	0	2,440577665
2015-01-01 18:00:00	58,2	32761,74925	25,056	0	2,436346399

Figura 3.14 DataFrame in formato Excel

In secondo luogo, è necessario gestire i dati mancanti. Nella maggior parte dei casi i dati che vengono utilizzati presentano dei valori mancanti, rappresentati da Python con il termine “Nan” (abbreviazione di *Not a number*). Affinché il modello funzioni, è necessario eliminare tali valori e, per farlo, esistono diversi approcci. Il primo consiste nell’eliminare dal DataFrame le colonne che contengono valori mancanti; utilizzare questo metodo è una scelta drastica e bisogna prestare attenzione poiché si potrebbero perdere dati essenziali alla definizione del modello. Il secondo approccio prevede la sostituzione del dato mancante con un altro valore; scegliere questa strada porta solitamente ad avere un modello più accurato, nonostante il dato sostituito non sia mai esattamente uguale a quello reale. Per realizzare la sostituzione può essere utilizzata la funzione *SimpleImputer()* della libreria Scikit-learn, essa permette di inserire al posto di tutti i valori mancanti della colonna un valore corrispondente alla media o alla mediana dei dati presenti nella colonna stessa, oppure sostituirli con il valore del dato più frequente o un valore costante, a seconda della strategia scelta. Un’alternativa consiste nel selezionare la colonna di interesse e utilizzare la

funzione Pandas *fillna()*, specificando tra parentesi quale metodo utilizzare (con il metodo *bfill* si utilizza il valore successivo a quello mancante, mentre con il metodo *ffill* il valore precedente) [24].

3.5.2 La normalizzazione dei dati

Il processo di normalizzazione consiste nello scalare i dati in modo tale da farli variare all'interno di un intervallo prestabilito. Si tratta di un passaggio fondamentale, poiché lo svolgimento di calcoli con numeri variabili in un range limitato previene alcune imprecisioni numeriche e limita la potenza di calcolo richiesta; inoltre si è riscontrata una migliore performance di alcuni algoritmi di machine learning se, questi ultimi, vengono addestrati con dati normalizzati. La necessità di normalizzazione deriva dal fatto che spesso i dati in input ad un modello di machine learning presentano range di variazione molto diversi gli uni dagli altri o unità di misura differenti (Euro, Km, °C, ecc.); di conseguenza i dati con ordine di grandezza maggiore finirebbero per governare le scelte dell'algoritmo, alterando in questo modo i risultati. Esistono diversi metodi per scalare i dati [25]:

- La normalizzazione Min-Max consiste nello scalare ogni dato appartenente ad una colonna in modo tale che rientri nel range [0,1] o il range [-1,1], a seconda della natura dei dati stessi; la formula che viene applicata ad ogni dato è la seguente:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Dove x' è il valore normalizzato, x è il valore originale e $\min(x)$ e $\max(x)$ sono, rispettivamente, il valore minimo e massimo tra i dati presenti nella colonna. Dal punto di vista pratico tale trasformazione viene eseguita mediante la funzione *MinMaxScaler()* della libreria Scikit-learn. Nello script sottostante, a titolo esemplificativo, viene riportata tale trasformazione applicata ai dati relativi al prezzo dell'energia.

```
1 # importazione valori del prezzo dell'energia
2 p_energia = pd.read_excel(r'C:\Documents\Dati\Dati.xlsx',
                           sheet_name='PrezzoEnergia', usecols='A')
```

```

3 # trasformazione del DataFrame p_energia in un Numpy array,
  #poichè MinMaxScaler opera con array
4 p_energia = asarray(p_energia)
5 # definizione dello scaler e del range di variazione
6 scaler = MinMaxScaler(feature_range=(0, 1))
7 # applicazione dello scaler ai dati
8 scaled_p_energia = scaler.fit_transform(p_energia)

```

Script 2 Normalizzazione dei dati mediante MinMaxScaler

Nella tabella successiva viene riportato il confronto tra i dati reali e quelli scalati.

Tabella 1 Confronto Prezzi Reali e Prezzi Scalati

	Prezzo Reale	Prezzo Scalato
0	51.10	0.340621
1	49.00	0.326623
2	39.10	0.260632
3	35.87	0.239101
4	33.40	0.222637
...
17539	62.17	0.414411
17540	57.00	0.379949
17541	54.50	0.363236
17542	52.10	0.347287
17543	51.22	0.341421

Ne viene data, inoltre, una rappresentazione grafica in figura 3.15, facendo prima il confronto tra i dati reali e scalati, e poi uno zoom esclusivamente sui dati scalati.

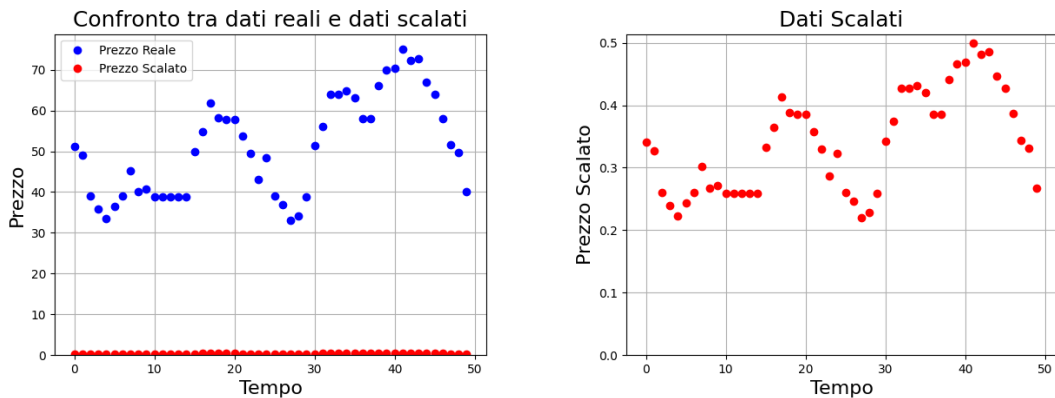


Figura 3.15 Dati Reali e Dati Scalati con MinMaxScaler

- Normalizzazione standard. Questo tipo di normalizzazione permette di scalare i dati in modo tale che il valore medio del dataset così ottenuto sia pari a 0 e la deviazione standard⁶ sia uguale a 1. Dal punto di vista matematico, la formula che viene applicata ad ogni dato è la seguente:

$$x' = \frac{x - \text{mean}(x)}{\text{standard deviation}}$$

$$\text{standard deviation} = \sqrt{\frac{\sum(x - \text{mean}(x))^2}{\text{count}(x)}}$$

Dove x' è il valore del dato scalato, x il valore reale, $\text{mean}(x)$ il valore medio dei dati e $\text{count}(x)$ il numero totale di dati presenti nella colonna. Per applicare tale operazione in Python può essere utilizzata la funzione `StandardScaler()` della libreria Scikit-learn; il procedimento è analogo a quello riportato per la trasformazione Min-Max.

⁶ La deviazione standard (o scarto quadratico medio) è un indice di dispersione statistico, vale a dire una stima della variabilità di una popolazione di dati o di una variabile casuale. È uno dei modi per esprimere la dispersione dei dati intorno ad un indice di posizione, quale può essere, ad esempio, la media aritmetica o una sua stima. Ha pertanto la stessa unità di misura dei valori. In statistica la precisione si può esprimere come lo scarto quadratico medio.

- Nel caso in cui tra i dati di input siano presenti valori anomali, ovvero valori estremi che sono molto al di fuori del range di variazione degli altri dati, potrebbe risultare difficoltoso applicare la funzione `StandardScaler`, poiché il calcolo della media e della deviazione standard risulterebbe errato proprio a causa della presenza di tali valori. In questo caso, quindi, viene utilizzata un altro tipo di normalizzazione che prevede l'utilizzo della seguente formula:

$$x' = \frac{x - \text{median}(x)}{IQR}$$

Dove x' è il valore del dato scalato, x il valore reale, $\text{median}(x)$ il valore corrispondente alla mediana dei dati presenti nella colonna e IQR è lo scarto interquartile (in inglese *Interquartile Range*), che è dato dalla differenza tra il 75° e il 25° quartile⁷ (ovvero l'ampiezza della fascia di valori che contiene la metà “centrale” dei valori osservati). I risultati ottenuti avranno, in questo modo, valore medio nullo e mediana e deviazione standard pari ad 1; i valori anomali, nonostante siano ancora presenti nei dati, non vanno ad influire sui valori delle grandezze statistiche. In Python questa operazione viene eseguita mediante la funzione `RobustScaler()` della libreria Scikit-learn, utilizzando lo stesso approccio descritto per la normalizzazione con `MinMaxScaler` [26].

3.5.3 La suddivisione dei dati

Il modello LSTM, una volta costruito, deve essere addestrato e testato: i dati a disposizione vengono quindi suddivisi in *train set*, *validation set* e *test set* (figura 3.16). Come visto in precedenza, i dati di *train* (x_{train} , y_{train}) vengono utilizzati per addestrare il modello e determinare il peso associato a ciascun neurone appartenente alla rete. Particolare importanza va posta, invece, sul *validation set*: esso è costituito da una percentuale dei dati di *train* stessi e viene utilizzato per regolare i parametri che caratterizzano il modello (come ad esempio il numero di strati nascosti presenti nella rete neurale). I dati di *test* (x_{test} , y_{test}), mai analizzati in precedenza

⁷ In statistica data una distribuzione di un carattere quantitativo oppure qualitativo ordinabile (ovvero le cui modalità possano essere ordinate in base a qualche criterio), i quartili sono quei valori/modalità che ripartiscono la popolazione in quattro parti di uguale numerosità.

dal modello, vengono infine dati in input al modello addestrato per verificarne l'accuratezza delle previsioni. Il processo di suddivisione dei dati è un passaggio delicato, poiché tale suddivisione non è univoca e varia al variare del problema analizzato [27]. Se, per esempio, l'analisi si basa su una serie storica di dati la cui estensione temporale è di 2 anni, appare logico suddividere i dati a metà (un anno per il *train* e un anno per il *test*), in questo modo ogni mese viene sottoposto alla fase di *train* una volta sola e avrà lo stesso "peso" nel determinare il modello. Se invece si hanno a disposizione più anni di dati, può risultare conveniente utilizzare un solo anno per il test.

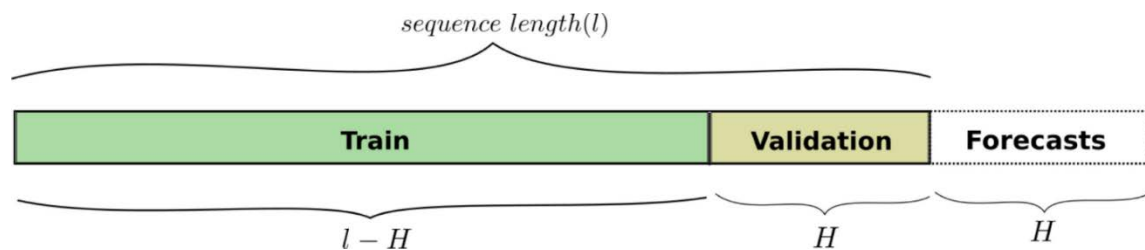


Figura 3.16 Suddivisione dei Dati

3.5.4 La definizione del modello e la funzione di attivazione

La definizione del modello vero e proprio richiede poche righe di codice. Qui di seguito viene riportato l'esempio di una rete neurale LSTM caratterizzata da due strati nascosti, rispettivamente di 80 e 45 unità, e da un'unità nello strato in uscita. Le funzioni utilizzate appartengono alla libreria Keras: la funzione *Sequential()* permette di definire il tipo di modello, ovvero un modello di tipo sequenziale costituito da più strati che vengono aggiunti man mano utilizzando la funzione *add()*. Nel caso del primo strato è necessario specificare la dimensione in ingresso (*input shape*); nell'esempio riportato `x_train.shape[1]` è uguale a 24, poiché si danno in ingresso i dati relativi alle 24 ore precedenti all'ora di cui si vuole ottenere la previsione, mentre `x_train.shape[2]` è pari a 2, ovvero le *features* considerate. Sono presenti, inoltre, due strati di *Dropout* (`layers.Dropout()`), posti tra i vari strati che costituiscono la rete. La loro presenza fa sì che durante ogni step della fase di addestramento alcuni neuroni, selezionati casualmente, vengano esclusi dalla rete; ciò implica la rimozione temporanea del loro contributo all'attivazione dei neuroni a valle e il mancato aggiornamento del loro stesso peso. Tale processo costringe le altre unità ad intervenire e ad

assumersi anche l'onere computazionale dei neuroni mancanti; in questo modo la rete risulta essere meno fragile, perché meno sensibile al peso di ogni singolo neurone [28].

```
1 LSTM_model = keras.Sequential()
2 LSTM_model.add(keras.layers.LSTM(units=80, activation='relu',
    return_sequences=True,
    input_shape=(x_train.shape[1], x_train.shape[2])))
3 LSTM_model.add(keras.layers.Dropout(rate=0.2))
4 LSTM_model.add(keras.layers.LSTM(units=45, activation='relu',
    return_sequences=False))
5 LSTM_model.add(keras.layers.Dropout(rate=0.2))
6 LSTM_model.add(keras.layers.Dense(units=1))
```

Script 3 Modello LSTM

Un particolare attenzione va ora posta sulla funzione di attivazione, indicata nel codice con il termine *activation*. Come già visto in precedenza, una rete neurale è costituita da più neuroni interconnessi tra loro, ognuno caratterizzato da un certo peso; utilizzando il peso a lui associato, il neurone esegue una trasformazione lineare sul dato in input. Il valore così ottenuto non viene passato subito allo strato successivo, ma subisce un'ulteriore elaborazione per mezzo della funzione di attivazione. A questo punto il dato è pronto per passare allo strato successivo e subire una trasformazione analoga; questo processo è noto in letteratura con il termine *Forward Propagation*, ovvero “Propagazione in avanti”. Il risultato ottenuto per mezzo di questa propagazione viene utilizzato per calcolare l'errore (errore quadratico medio, errore assoluto, ecc.) e, basandosi sull'errore stesso, vengono aggiornati i pesi dei singoli neuroni; tale meccanismo è conosciuto come *Back Propagation*, ovvero “Propagazione all'indietro”. Appare quindi evidente come la funzione di attivazione giochi un ruolo chiave nel processo di addestramento della rete, poiché, nonostante determini un incremento della complessità del modello, permette alla rete stessa

di riconoscere relazioni più complesse tra i dati. Esistono diverse funzioni di attivazione, ognuna caratterizzata da una specifica abilità computazionale [29]:

- La funzione binaria. Questa funzione permette di “disattivare” il neurone nel caso in cui il suo input sia maggiore di un valore prestabilito, in questo modo l’output di tale neurone non avrà alcun effetto sullo strato successivo. Dal punto di vista matematico può essere così rappresentata:

$$f(x) = \begin{cases} 1, & x \geq t \\ 0, & x < t \end{cases}$$

Dove t rappresenta il valore di soglia. La funzione binaria è la più semplice tra le funzioni di attivazione e può risultare particolarmente utile nei problemi di classificazione binari in cui, cioè, sono presenti solamente due classi; non avrà alcuna utilità invece nei problemi multi-classe. Un’altra limitazione è rappresentata dal calcolo del gradiente: durante il processo di propagazione all’indietro, il gradiente della funzione $f(x)$ viene utilizzato per aggiornare i pesi associati ai neuroni; in questo caso il gradiente della funzione è costantemente pari a zero e non potrà, dunque, essere utilizzato per l’aggiornamento.

- La funzione lineare. A differenza della funzione binaria, quella lineare presenta una dipendenza da x :

$$f(x) = tx$$

Dove t può essere un qualsiasi valore costante; l’output sarà, quindi, proporzionale all’input. Anche in questo caso l’aggiornamento dei pesi associati ai neuroni non è molto efficace, essendo il gradiente della funzione $f(x)$ costante e sempre uguale a t . L’utilizzo di questa funzione di attivazione può risultare, dunque, adatto solo nel caso di problemi semplici, dove non sia richiesto il riconoscimento di relazioni troppo complesse tra i dati.

- La funzione sigmoidea. Questa funzione, una delle più utilizzate, permette di trasformare l’output del neurone in un valore compreso tra 0 e 1. A differenza delle precedenti funzioni

di attivazione, quella sigmoidea è non lineare, di conseguenza sarà non lineare anche il suo output. L'espressione matematica ad essa associata è la seguente:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Per quanto riguarda il gradiente, facendo la derivata di $f(x)$ si ottiene il seguente risultato:

$$f'(x) = \frac{e^{-x}}{(1 + e^{-x})^2}$$

Esso presenta un valore significativo nel range $(-3,3)$ mentre, per altri valori di x , tende a zero; ciò significa che, per valori minori di -3 e maggiori di 3 , l'abilità di apprendimento della rete è molto scarsa. Inoltre, la funzione sigmoidea non è simmetrica in un intorno di zero e questo implica che gli output di tutti i neuroni saranno dello stesso segno.

- La funzione tangente iperbolica. Il comportamento di tale funzione è molto simile a quello della funzione sigmoidea; l'unica differenza è che la funzione $\tanh(x)$ è simmetrica in un intorno dello zero, di conseguenza gli input dello strato successivo non avranno tutti lo stesso segno. Il range di variazione inoltre si riduce, passando a $(-1,1)$. L'espressione matematica che caratterizza questa funzione di attivazione è la seguente:

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

Nonostante le caratteristiche di questa funzione siano molto simili a quelle della funzione sigmoidea, spesso si preferisce utilizzare la tangente iperbolica poiché è centrata rispetto allo zero e presenta un andamento del gradiente variabile (cioè non strettamente crescente come nel caso della sigmoide).

- La funzione ReLu. Il termine ReLu sta per *Rectified Linear Unit*, e si tratta di un'altra funzione non lineare molto diffusa nel campo del *deep learning*. Il principale vantaggio nell'utilizzare questa funzione di attivazione sta nel fatto che non attiva contemporaneamente tutti i neuroni; un neurone, infatti, viene attivato solamente quando il

risultato della trasformazione lineare è maggiore di zero. Tale concetto è facilmente intuibile osservando l'espressione matematica alla base di tale funzione e la sua rappresentazione grafica in figura 3.17.

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

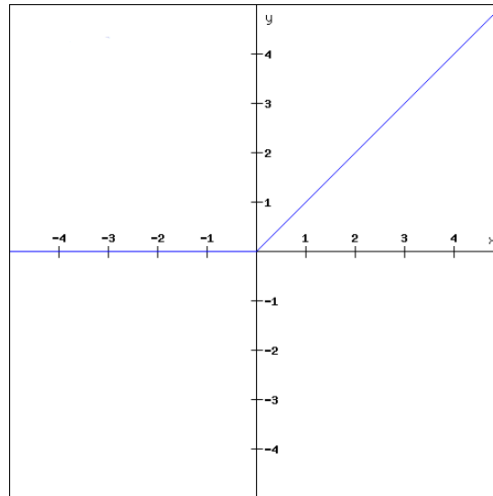


Figura 3.17 Funzione di Attivazione ReLu

Il gradiente di tale funzione è nullo per x minore di zero e uguale a 1 per x maggiore di zero; ciò significa che, durante la propagazione all'indietro, il peso associato ad alcuni neuroni non viene aggiornato e questo può determinare la presenza di neuroni spenti, che non vengono mai attivati.

- La funzione ELU (*Exponential Linear Unit*). Si tratta di una variante della funzione ReLu poiché, a differenza di quest'ultima, introduce una modifica per i valori di x minori di zero. La funzione ELU, infatti, utilizza una curva logaritmica per definire tali valori. L'espressione matematica che la identifica è la seguente:

$$f(x) = \begin{cases} x, & x \geq 0 \\ a(e^x - 1), & x < 0 \end{cases}$$

3.5.6 La compilazione del modello

Una volta definita la rete neurale, è necessario compilarla. Questa fase è molto importante, poiché permette di rendere efficiente il modello: essa trasforma la semplice sequenza di strati, precedentemente definita, in un serie altamente efficiente di matrici di trasformazione, con un formato che può essere eseguito sulla GPU o CPU⁸ (a seconda della configurazione di Keras). Per svolgere tale azione è necessario definire determinati parametri; in particolare, deve essere identificato l'algoritmo di ottimizzazione, utilizzato durante la fase di addestramento per regolare i parametri interni del modello, e la funzione di perdita (in inglese *loss function*), che viene minimizzata dall'algoritmo di ottimizzazione stesso al fine di valutare le performance del modello. Al fine di rendere efficace questa trattazione si preferisce concentrarsi sul problema della regressione, l'algoritmo di ottimizzazione utilizzato nel modello successivamente descritto e le diverse funzioni di perdita che possono essere ad esso associate. Quando si affronta un problema di regressione, ovvero un problema caratterizzato in uscita da un valore reale o continuo, l'algoritmo di ottimizzazione più largamente impiegato è la Discesa Stocastica del Gradiente (in inglese *Stochastic gradient descent*). In termini matematici, il gradiente è definito come la pendenza di una curva o superficie; quindi, "Discesa del Gradiente" significa letteralmente "muoversi lungo una curva fino a raggiungerne il punto più basso", ovvero quello caratterizzato da valore del gradiente nullo. Come appena accennato, durante la fase di compilazione viene definita anche la funzione di perdita; l'algoritmo si occuperà, dunque, di trovare il minimo di tale funzione, ovvero i valori dei pesi associati ad ogni unità della rete che permettono di minimizzarla. Il processo di minimizzazione può essere riassunto in alcuni step:

- Determinare la pendenza della funzione obiettivo rispetto ad ogni parametro/*feature*, in altre parole calcolarne il gradiente;

⁸ La CPU (in inglese *Central Processing Unit*) indica in elettronica e informatica l'unità o sottosistema logico e fisico che sovrintende alle funzionalità logiche di elaborazione principali del computer; in ciò essa si contrappone a tutte le altre unità di elaborazione secondarie presenti nelle architetture hardware dei computer, ovvero le varie schede elettroniche (scheda audio, scheda video, scheda di rete, coprocessore e processore di segnale digitale). Invece la GPU (dall'inglese *Graphics Processing Unit*) o processore grafico, è un tipo particolare di coprocessore che si contraddistingue per essere specializzato nella resa grafica d'immagini (grafica computerizzata). Il suo tipico utilizzo è come coprocessore della CPU, e da alcuni anni viene anche utilizzata in generiche elaborazioni dati [30] [31].

- Scegliere per i parametri un valore iniziale casuale;
- Aggiornare la funzione del gradiente inserendo i valori dei parametri;
- Calcolare la dimensione del passo per ogni *feature* con la formula:

$$\text{dimensione del passo} = \text{gradiente} \times \text{velocità di apprendimento}$$

Dove la “velocità di apprendimento” rappresenta un parametro flessibile, che influenza pesantemente la convergenza dell’algoritmo. Se la velocità di apprendimento risultasse troppo elevata, la dimensione del passo compiuto dall’algoritmo nel discendere lungo la curva potrebbe risultare così ampia da “mancare” il punto di minimo; è buona norma, quindi, settare tale parametro ad un valore basso, ad esempio 0,01.

- Calcolare il nuovo parametro nel seguente modo:

$$\text{nuovo parametro} = \text{vecchio parametro} - \text{dimensione del passo}$$

- Ripetere i tre step precedenti finché il gradiente è quasi nullo.

Uno degli aspetti negativi di tale algoritmo è il numero di computazioni che vengono fatte ad ogni iterazione: questo tipo di problemi sono solitamente caratterizzati da una quantità elevata di dati, tale per cui il numero di iterazioni potrebbe raggiungere l’ordine delle decine di milioni. In questo frangente rientra, dunque, il carattere stocastico dell’algoritmo. Ad ogni step, infatti, seleziona casualmente dei dati per i quali calcolare il valore delle derivate, riducendo in questo modo lo sforzo computazionale. Nel modello presentato nel capitolo quattro, viene impiegato l’ottimizzatore Adam. Si tratta di un’estensione della discesa stocastica del gradiente, introdotto da Diederik Kingma e Jimmy Ba nel 2015. Nell’approccio classico, la velocità di apprendimento (indicata con la lettera α) viene mantenuta costante durante tutta la fase di addestramento e uguale per tutti i parametri; utilizzando l’ottimizzatore Adam, invece, viene definita una velocità di apprendimento diversa per ogni parametro, adattata man mano durante l’addestramento. Negli ultimi anni questo algoritmo ha visto un’ampia diffusione, risultando molto efficace nelle applicazioni di *deep learning*; la scelta dell’algoritmo di ottimizzazione può, infatti, fare la

differenza tra ottenere i risultati in alcuni minuti, ore o, addirittura, giorni [32]. Per quanto riguarda la funzione di perdita, necessaria per il calcolo dell'errore commesso ad ogni step dal modello e il conseguente aggiornamento dei pesi per ridurre tale l'errore, nei problemi di regressione ne possono essere utilizzati di diversi tipi [33]:

- Errore quadratico medio (MSE). Viene calcolato come la media delle differenze al quadrato tra i valori predetti e i valori reali; il risultato è sempre un valore positivo e si ha convergenza perfetta solo quando è nullo. Elevare al quadrato gli errori permette di accentuare gli errori più grandi, a discapito di quelli più piccoli. A livello pratico, nel codice Python, per definire questo tipo di errore viene utilizzata la linea di codice qui riportata.

```
1 model.compile(loss='mean_squared_error', optimizer='adam')
```

Script 4 Compilazione modello con funzione di perdita MSE

- Errore logaritmico quadratico medio (MSLE). Questa funzione di perdita esegue il logaritmo della differenza tra valore predetto e valore reale prima di eseguire l'elevamento al quadrato e la media; in questo modo, a differenza del caso precedente, la presenza di errori elevati non viene accentuata. Viene utilizzata soprattutto nel caso di modelli caratterizzati da dati molto sparsi o in cui, per la previsione, vengono utilizzati dati non scalati. Dal punto di vista della programmazione, si avrà:

```
1 model.compile(loss='mean_squared_logarithmic_error',  
                optimizer='adam')
```

Script 5 Compilazione modello con funzione di perdita MSLE

- Errore assoluto medio (MAE). L'uso di tale funzione è appropriato nel caso in cui la distribuzione della variabile obiettivo non sia gaussiana ma presenti valori anomali lontani dal valore medio. L'errore assoluto medio calcola la media delle differenze tra valori predetti e valori reali. In questo caso la compilazione viene effettuata nel modo seguente:

```
1 model.compile(loss='mean_absolute_error', optimizer='adam')
```

Script 6 Compilazione modello con funzione di perdita MAE

3.5.7 L'addestramento del modello

L'ultima fase da eseguire per completare il modello e utilizzarlo poi per avere delle previsioni, è la fase di addestramento vero e proprio. Durante questo processo i dati di *train* (x_{train} , y_{train}) vengono dati in ingresso alla rete (precedentemente costruita e compilata), affinché riconosca le relazioni presenti tra input e output [34]. Anche in questo caso si scrivono solo poche righe di codice:

```
1 early_stop = keras.callbacks.EarlyStopping(monitor='val_loss',
                                             patience=5)
2 history = model.fit(X_train, y_train, epochs=100, batch_size=24,
                    shuffle=False, validation_split=0.1,
                    callbacks=[early_stop])
```

Script 7 Addestramento del modello LSTM

I parametri specificati nella funzione *fit()* determinano il tipo di addestramento a cui sarà sottoposta la rete neurale:

- *Epochs*. Rappresenta il numero di volte in cui l'intero set dei dati di *train* verrà sottoposto all'algoritmo di apprendimento; nell'esempio qui riportato tale parametro è uguale a 100, ciò significa che ogni dato all'interno del dataset di addestramento avrà 100 possibilità per influire sull'aggiornamento dei pesi associati a ciascuna unità della rete;
- *Batch_size*. Definisce il numero di dati che devono essere presi in considerazione prima di procedere con l'aggiornamento dei pesi; sempre facendo riferimento allo script precedente, se il valore di tale parametro è 24 si può pensare, per semplicità, che l'algoritmo operi un

ciclo for iterando su un numero di dati pari a 24 e che, alla fine di tale processo, le previsioni vengano confrontate con i valori reali per calcolare l'errore commesso. A partire dall'errore calcolato, viene poi eseguito l'aggiornamento del modello. Se il valore di questo parametro coincide con la dimensione del *dataset* di *train*, l'algoritmo è chiamato *Batch gradient descent*, se invece è uguale a 1, prende il nome di *Stochastic gradient descent* [35].

- *Shuffel*. Indicare come “Falso” questo comando fa sì che i dati di *train* non vengano rimescolati in modo casuale prima di essere dati in input alla rete. Questo aspetto è particolarmente importante nel caso di analisi di serie storiche poiché, in questi problemi, è rilevante la successione temporale dei valori analizzati.
- *Validation_split*. Identifica la quota di dati di *train* che viene utilizzata per la validazione del modello.

Risulta ora necessario un approfondimento di questo ultimo processo e della funzione *early_stop*, richiamata nella fase di addestramento dal parametro *callbacks*. I dati appartenenti al *validation set* vengono utilizzati alla fine di ogni iterazione durante l'addestramento per valutare l'accuratezza del modello; questo permette di monitorare l'andamento del rendimento del modello stesso durante tutta questa prima fase. Quando si addestra un algoritmo di apprendimento in modo iterativo, infatti, si ottiene un miglioramento del modello solo fino ad un certo numero di iterazioni; dopo quel punto, la capacità del modello di generalizzare può indebolirsi iniziando a sovra-utilizzare i dati di *train*. Questo fenomeno è noto in letteratura come *overfitting*. Esso si verifica quando un modello nella fase di addestramento apprende, oltre alle relazioni specifiche, anche il rumore e le fluttuazioni casuali presenti nei dati di *train*, rendendoli parte del modello stesso; il problema si presenta successivamente, durante la fase di *test*, poiché tali relazioni non si possono applicare ai nuovi dati, che saranno caratterizzati da rumore e fluttuazioni del tutto diverse. Un modello soggetto ad *overfitting* presenterà, dunque, ottime prestazioni sul *train set* e pessime prestazioni sul *test set*. Tale problema può presentarsi anche in forma inversa, quando l'algoritmo non è in grado di modellare i dati di *train*; in tal caso si parla, allora, di *underfitting*. Nella figura sottostante viene riportata una sintesi grafica delle problematiche appena discusse.

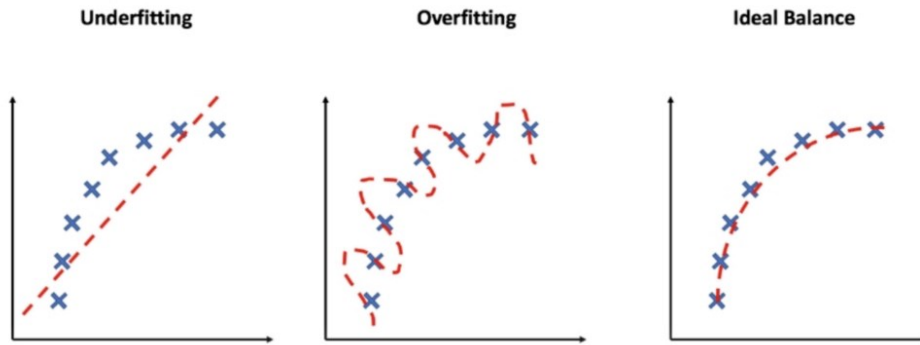


Figura 3.18 Overfitting e Underfitting

Come accennato in precedenza, il problema dell'*overfitting* può essere risolto interrompendo l'addestramento del modello nel momento in cui non si ha più alcun incremento del suo rendimento; ovvero quando l'accuratezza calcolata sui dati di validazione si stabilizza o, addirittura, inizia a diminuire. Per eseguire l'arresto anticipato viene definita la funzione *early_stop*, che monitora l'andamento della funzione di perdita sul *validation set* e interrompe l'addestramento se non registra alcun miglioramento dopo un numero di iterazioni pari al valore indicato dal parametro *patience*. La presenza di *overfitting* o *underfitting*, quindi, può essere riconosciuta visivamente osservando il grafico che plotta l'andamento della funzione di perdita sia sui dati di addestramento che sui dati di validazione. In figura 3.19 vengono riportati in successione il caso di modello in *underfitting*, *overfitting* e il caso di convergenza tra le due funzioni, sinonimo di modello ben addestrato [36] [37].

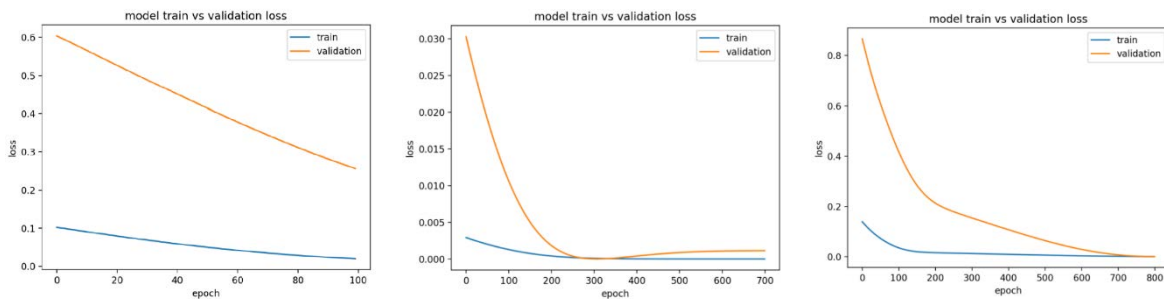


Figura 3.19 Confronto tra la funzione di perdita valutata sul train set e validation set nei tre diversi casi

3.5.8 La valutazione del modello

Una volta terminata la fase di addestramento il modello è completo e può essere applicato ai dati di *test* per valutarne l'efficacia; i dati presenti nel *test set* sono, infatti, del tutto nuovi per il modello. Attraverso il comando riportato nello script 8, vengono dati in ingresso i valori di *test* delle *features* (`x_test`) e si ottengono in uscita le previsioni effettuate dal modello. Tali valori vengono successivamente confrontati con i valori di output reali (`y_test`) e se ne calcola l'errore associato (RMSE, *Root Mean Square Error*) [34].

```
1 y_pred = model.predict(x_test)
2 error = y_pred - y_test
3 RMSE = np.sqrt(np.mean(error**2))
4 print(RMSE)
```

Script 8 Applicazione del modello LSTM

Capitolo 4 - Previsione del Prezzo zonale dell'energia elettrica

Nei capitoli precedenti sono state discusse le caratteristiche che contraddistinguono il prezzo dell'energia ed è stato proposto un possibile approccio per fare una stima di tale grandezza. Nei paragrafi successivi verrà presentato un modello di machine learning, in particolare una rete neurale LSTM, realizzato in ambiente Python e finalizzato alla previsione del prezzo dell'energia elettrica in Sardegna, una delle sei zone geografiche in cui si articola il mercato elettrico italiano.

4.1 Le tipologie di analisi e la features considerate

La procedura seguita per realizzare il modello LSTM è quella descritta nel capitolo 3, si ha quindi una fase iniziale di reperimento ed elaborazione dei dati, a cui segue la costruzione e l'addestramento della rete neurale e infine l'applicazione del modello ai dati di test, per valutarne l'efficacia nell'effettuare le previsioni. In questo studio sono state realizzate due tipologie di analisi, con conseguente realizzazione di due diversi modelli LSTM. La prima rete neurale è costruita in modo tale da realizzare previsioni del prezzo su base oraria, la seconda invece su base giornaliera; nel primo caso per ottenere tali risultati è necessario dare in input al modello i dati relativi alle 23 ore precedenti all'ora di cui si vuole ottenere la previsione e quelli relativi all'ora stessa, mentre nel secondo caso per ottenere i prezzi previsti di un'intera giornata vengono dati in input i dati relativi al giorno precedente. In figura 4.1 e 4.2 viene data una rappresentazione grafica dei due diversi approcci, dove con $i(t)$ si indicano i dati di input relativi alle diverse *features*, mentre con $o(t)$ l'output, ovvero il prezzo stimato dell'energia. Per la costruzione e la valutazione del modello sono stati utilizzati dati su base oraria relativi agli anni 2015 e 2016, utilizzando il primo per eseguire l'addestramento del modello e il secondo per valutarne l'efficacia predittiva.

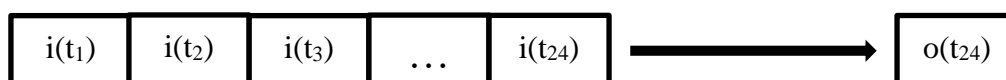


Figura 4.1 Previsione oraria

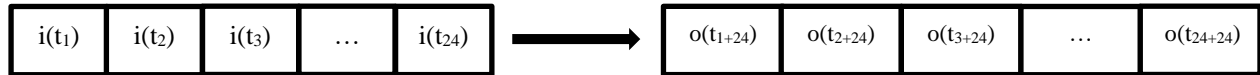


Figura 4.2 Previsione giornaliera

I dati relativi al prezzo dell'energia elettrica, ovvero della grandezza di cui si vuole effettuare la previsione, sono scaricabili dal sito del GME nella sezione “Esiti dei Mercati e Statistiche” (<https://www.mercatoelettrico.org/it/Esiti/MGP/EsitiMGP.aspx>); se ne riportano alcuni esempi in figura 4.3.

	A	B
1	Data Ora	Prezzo
2	2015-01-01 00:00:00	51,1
3	2015-01-01 01:00:00	49
4	2015-01-01 02:00:00	39,1
5	2015-01-01 03:00:00	35,87
6	2015-01-01 04:00:00	33,4
7	2015-01-01 05:00:00	36,57
8	2015-01-01 06:00:00	39,1
9	2015-01-01 07:00:00	45,21
10	2015-01-01 08:00:00	40
11	2015-01-01 09:00:00	40,8
12	2015-01-01 10:00:00	38,8
13	2015-01-01 11:00:00	38,8
14	2015-01-01 12:00:00	38,85
15	2015-01-01 13:00:00	38,8
16	2015-01-01 14:00:00	38,85
17	2015-01-01 15:00:00	49,95

Figura 4.3 Prezzo dell'energia elettrica (Euro/MWh)

In figura 4.4, invece, viene rappresentato l'andamento di tale grandezza negli anni 2015 e 2016.

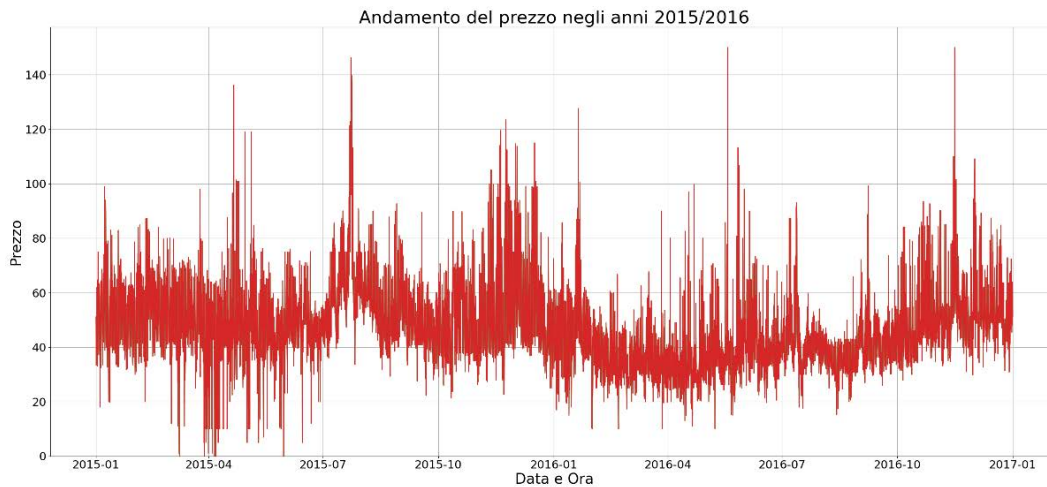


Figura 4.4 Andamento del prezzo dell'energia elettrica negli anni 2015/2016

Il valore del prezzo dell'energia, come visto in precedenza, è influenzato da numerosi fattori, ma solo alcuni di questi sono stati considerati nello sviluppo dei modelli qui descritti. Si presenteranno ora le diverse *features*, riportando la fonte e l'elaborazione eseguita per ognuna di esse.

4.1.1 La potenza assorbita

Nel capitolo due si è evidenziata la stretta correlazione tra gli andamenti del prezzo e della domanda di energia, ovvero la potenza assorbita in un certo arco temporale. Tale informazione può essere reperita sul sito di Terna, nella sezione "Download Center". I dati scaricati fanno riferimento alla sola regione Sardegna e riportano la potenza assorbita in MW ogni quarto d'ora. Per ottenere, quindi, la potenza assorbita in una determinata ora si è eseguita la media dei quattro valori (uno ogni 15 minuti) riferiti all'ora considerata.

	A	B	C
1	Data Ora	Prezzo	Total Load (MW)
2	2015-01-01 00:00:00	51,1	1000,25975
3	2015-01-01 01:00:00	49	945,4935
4	2015-01-01 02:00:00	39,1	902,282
5	2015-01-01 03:00:00	35,87	865,112
6	2015-01-01 04:00:00	33,4	841,468
7	2015-01-01 05:00:00	36,57	825,2005
8	2015-01-01 06:00:00	39,1	845,2095
9	2015-01-01 07:00:00	45,21	872,57575
10	2015-01-01 08:00:00	40	917,54525
11	2015-01-01 09:00:00	40,8	1001,6975
12	2015-01-01 10:00:00	38,8	1073,7455
13	2015-01-01 11:00:00	38,8	1138,248
14	2015-01-01 12:00:00	38,85	1140,11125
15	2015-01-01 13:00:00	38,8	1054,81625
16	2015-01-01 14:00:00	38,85	965,79425
17	2015-01-01 15:00:00	49,95	965,3215

Figura 4.5 Potenza assorbita (MW)

4.1.2 Il prezzo del gas

Il prezzo del gas e il prezzo dell'energia sono tra loro correlati e ciò è dovuto al fatto che il gas naturale viene utilizzato sia per il riscaldamento residenziale sia per la generazione di energia elettrica per uso residenziale. La liberalizzazione del mercato del gas ha rappresentato, in Italia, uno dei momenti topici nelle privatizzazioni; dall'1 Gennaio 2003 è infatti possibile per ogni cliente scegliere il proprio fornitore di gas naturale. Il Codice di Rete per il servizio di distribuzione gas (CRDG), approvato dall'Autorità di Regolazione Per Energia Reti e Ambiente (ARERA), è lo strumento contrattuale con cui vengono regolati i rapporti tra i diversi soggetti presenti nel mercato del gas:

- Trasportatore, il quale possiede e gestisce le infrastrutture per il trasporto del gas in alta pressione dal luogo di estrazione fino ai punti di immissione nella rete in bassa pressione;
- Grossista, ovvero il proprietario del gas all'interno dei gasdotti gestiti dal trasportatore;

- Distributore, che possiede o più spesso gestisce in concessione le reti di distribuzione in media e bassa pressione (reti urbane);
- Società di vendita, la quale possiede il gas nelle reti in bassa pressione gestite dal distributore e si interfaccia con i clienti finali per la vendita.

Con la liberalizzazione, dunque, si è creata una netta distinzione tra distributore e società di vendita, con conseguente modifica della tariffa applicata ai clienti finali, la quale è ora composta da due quote: la prima relativa all'acquisto del gas e la seconda relativa al costo sostenuto dalla società di vendita per i servizi del distributore. Analizzando gli andamenti del prezzo del gas e del prezzo dell'energia si può notare una certa similitudine tra i due *trend*, di conseguenza nella previsione del prezzo dell'energia elettrica è necessario tener conto anche del prezzo del gas. Tale considerazione può assumere poca rilevanza nel caso della previsione del prezzo in Sardegna, poiché il consumo di gas in questa regione è molto minore se paragonato a quello delle altre regioni italiane. Nel 2016, ad esempio, si è registrato un consumo annuo di 8,1 GWh, ovvero solo lo 0,006% del consumo di gas nazionale. Nei vari casi presi in considerazione successivamente, quindi, si è fatto il confronto tra considerare o non considerare come *feature* il prezzo del gas, per valutare effettivamente quale sia l'impatto di tale grandezza sul prezzo dell'energia elettrica. I dati relativi al prezzo del gas sono scaricabili dal sito del GME, sempre nella sezione "Esiti dei Mercati e Statistiche".

	A	B	C	D
1	Data Ora	Prezzo	Total Load (MW)	Prezzo Gas
2	2015-01-01 00:00:00	51,1	1000,25975	25,056
3	2015-01-01 01:00:00	49	945,4935	25,056
4	2015-01-01 02:00:00	39,1	902,282	25,056
5	2015-01-01 03:00:00	35,87	865,112	25,056
6	2015-01-01 04:00:00	33,4	841,468	25,056
7	2015-01-01 05:00:00	36,57	825,2005	25,056
8	2015-01-01 06:00:00	39,1	845,2095	25,056
9	2015-01-01 07:00:00	45,21	872,57575	25,056
10	2015-01-01 08:00:00	40	917,54525	25,056
11	2015-01-01 09:00:00	40,8	1001,6975	25,056
12	2015-01-01 10:00:00	38,8	1073,7455	25,056
13	2015-01-01 11:00:00	38,8	1138,248	25,056
14	2015-01-01 12:00:00	38,85	1140,11125	25,056
15	2015-01-01 13:00:00	38,8	1054,81625	25,056
16	2015-01-01 14:00:00	38,85	965,79425	25,056
17	2015-01-01 15:00:00	49,95	965,3215	25,056

Figura 4.6 Prezzo del gas (Euro/MWh)

4.1.3 La radiazione solare

L'informazione relativa alla radiazione solare, reperibile per gli anni 2015 e 2016 sul sito PVGIS, è fondamentale per tener conto della quantità di energia elettrica prodotta dagli impianti fotovoltaici presenti sul territorio sardo. La parte di energia rinnovabile non auto-consumata viene immessa in rete e arriva sul mercato elettrico, offerta in gran parte dal GSE (ente che gestisce il ritiro della produzione rinnovabile per la maggioranza di piccoli e medi impianti in Italia); tale energia, essendo prodotta da un impianto caratterizzato da un costo marginale⁹ molto basso, viene proposta sul mercato a prezzi praticamente nulli. In questo modo, secondo la disposizione per ordine di merito, tendono ad essere selezionate le fonti di energia rinnovabili, tra cui il fotovoltaico, a discapito delle fonti caratterizzate invece da costi marginali più alti, come le centrali termoelettriche. Per dare evidenza di questo fatto, poiché influenza molto il prezzo dell'energia, si

⁹ Rappresenta il costo che è necessario sostenere per produrre una unità di prodotto in più (nel caso di centrali elettriche, un kWh o suoi multipli). Dato che tra tutti i costi solo alcuni variano con la produzione (detti appunto costi variabili), come ad esempio il costo per i combustibili (le materie prime), ne deriva che il costo marginale dipende solo dall'incremento di tali costi.

considerano i dati relativi alla radiazione solare delle otto province sarde (poiché tale parametro è variabile a seconda della zona) e li si scala mediante il coefficiente c , dato dalla seguente espressione:

$$c = \frac{\text{produzione annua da fotovoltaico per provincia (GWh)}}{\text{produzione annua da fotovoltaico della Sardegna (GWh)}}$$

I valori relativi alla produzione annua sono disponibili sui Report Statistici del GSE; nel caso del 2014/2015, ad esempio, i valori riportati sono i seguenti:

	Produzione GWh	
	2014	2015
Sardegna	952,5	916,7
Cagliari	217,0	192,8
Carbonia-Iglesias	70,3	71,0
Medio Campidano	82,1	81,7
Nuoro	135,8	104,3
Ogliastra	28,9	27,4
Olbia-Tempio	40,2	40,5
Oristano	164,1	182,2
Sassari	214,0	216,9

Figura 4.7 Produzione da fotovoltaico

La relazione che lega radiazione solare e producibilità di un impianto fotovoltaico è di tipo lineare, di conseguenza, una volta calcolato il parametro c , si esegue il prodotto tra il dato orario relativo alla radiazione solare e il coefficiente c stesso. Tale operazione viene eseguita per ogni provincia, dopodiché si calcola la media dei valori ottenuti.

	A	B	C	D	E
1	Data Ora	Prezzo	Total Load (MW)	Prezzo Gas	Solar Radiation
2	2015-01-01 00:00:00	51,1	1000,25975	25,056	0
3	2015-01-01 01:00:00	49	945,4935	25,056	0
4	2015-01-01 02:00:00	39,1	902,282	25,056	0
5	2015-01-01 03:00:00	35,87	865,112	25,056	0
6	2015-01-01 04:00:00	33,4	841,468	25,056	0
7	2015-01-01 05:00:00	36,57	825,2005	25,056	0
8	2015-01-01 06:00:00	39,1	845,2095	25,056	0
9	2015-01-01 07:00:00	45,21	872,57575	25,056	8,404701599
10	2015-01-01 08:00:00	40	917,54525	25,056	47,7625933
11	2015-01-01 09:00:00	40,8	1001,6975	25,056	79,09251242
12	2015-01-01 10:00:00	38,8	1073,7455	25,056	99,07026576
13	2015-01-01 11:00:00	38,8	1138,248	25,056	107,527734
14	2015-01-01 12:00:00	38,85	1140,11125	25,056	106,1986385
15	2015-01-01 13:00:00	38,8	1054,81625	25,056	93,69401489
16	2015-01-01 14:00:00	38,85	965,79425	25,056	70,16826101
17	2015-01-01 15:00:00	49,95	965,3215	25,056	35,52908338

Figura 4.8 Radiazione solare (W/m²)

4.1.4 La velocità del vento

La velocità del vento, come la radiazione solare, è un parametro che permette di considerare la produzione da fonti energetiche rinnovabili, in particolare da eolico. Alla fine del 2016 la Sardegna si classificava quarta tra le regioni italiane per numerosità di impianti (354) e quinta per potenza installata (1.011,5 MW); appare evidente, quindi, come considerare tale grandezza sia importante per stimare correttamente il prezzo dell'energia. L'elaborazione eseguita sui dati relativi alla velocità del vento, scaricati anch'essi dal sito PVGIS, è la stessa applicata alla radiazione solare: si calcola, quindi, il parametro c e lo si moltiplica per la velocità oraria del vento, tenendo conto che la producibilità di un impianto eolico dipende dal cubo di questa grandezza. La produzione totale da eolico in Sardegna negli anni 2015 e 2016 ammonta a 1.535,8 GWh mentre, per quanto riguarda la produzione di ogni singola provincia sarda, si è fatto riferimento ai dati riportati nei report statistici del GSE, in cui compaiono le mappe riportate in figura 4.9 e 4.10.

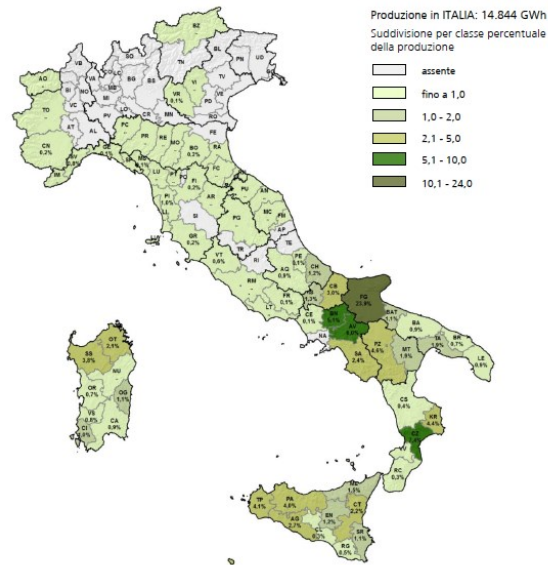


Figura 4.9 Distribuzione provinciale della produzione eolica nel 2015

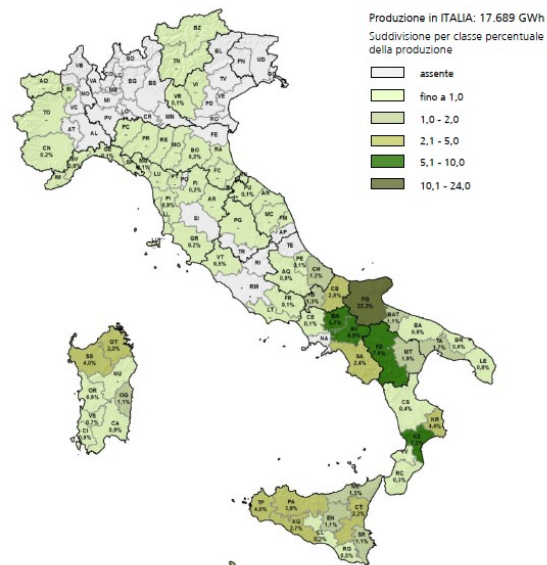


Figura 4.10 Distribuzione provinciale della produzione eolica nel 2016

Prima di eseguire la moltiplicazione tra produzione relativa e velocità del vento è necessario apportare una modifica ai dati, ponendo uguale a 0 m/s tutti i valori di velocità inferiori ai 3 m/s e uguale a 15 m/s tutti i dati maggiori o uguali a tale valore. Infatti, per velocità del vento minori di 3 m/s l'aerogeneratore non entra in funzione, mentre una velocità del vento superiore ai 15 m/s

risulta dannosa per il funzionamento stesso della macchina; questi valori, dunque, vengono considerati come limiti di funzionamento. Al termine di questi passaggi si esegue la media oraria dei valori ottenuti per ogni provincia sarda, in modo tale da considerare il comportamento medio dell'intera regione.

	A	B	C	D	E	F
1	Data Ora	Prezzo	Total Load (MW)	Prezzo Gas	Solar Radiation	Wind Speed (h:10m, m/s)
2	2015-01-01 00:00:00	51,1	1000,25975	25,056	0	27,0851063
3	2015-01-01 01:00:00	49	945,4935	25,056	0	25,58794874
4	2015-01-01 02:00:00	39,1	902,282	25,056	0	24,28337999
5	2015-01-01 03:00:00	35,87	865,112	25,056	0	23,43482514
6	2015-01-01 04:00:00	33,4	841,468	25,056	0	19,53264867
7	2015-01-01 05:00:00	36,57	825,2005	25,056	0	16,87551447
8	2015-01-01 06:00:00	39,1	845,2095	25,056	0	15,0548481
9	2015-01-01 07:00:00	45,21	872,57575	25,056	8,404701599	12,3853235
10	2015-01-01 08:00:00	40	917,54525	25,056	47,7625933	11,82345162
11	2015-01-01 09:00:00	40,8	1001,6975	25,056	79,09251242	11,5167209
12	2015-01-01 10:00:00	38,8	1073,7455	25,056	99,07026576	10,90923622
13	2015-01-01 11:00:00	38,8	1138,248	25,056	107,527734	11,22610985
14	2015-01-01 12:00:00	38,85	1140,11125	25,056	106,1986385	12,1404663
15	2015-01-01 13:00:00	38,8	1054,81625	25,056	93,69401489	12,59738304
16	2015-01-01 14:00:00	38,85	965,79425	25,056	70,16826101	13,65990617
17	2015-01-01 15:00:00	49,95	965,3215	25,056	35,52908338	15,2973937

Figura 4.11 Velocità del vento (m/s)

4.1.5 Giorno feriale o festivo

Nel capitolo due è stata messa in evidenza la stretta correlazione tra prezzo e fabbisogno di energia, ponendo particolare attenzione sulla differenza tra giorno feriale e festivo; come detto in precedenza, infatti, nei giorni lavorativi si ha una domanda, e quindi un prezzo, di energia più alto rispetto ai giorni festivi. Per mettere in evidenza questa relazione si è creata un'ulteriore *feature* in cui si identificano i giorni feriali con il numero 1 e i giorni festivi con il numero 0; questo procedimento viene eseguito per ciascun giorno degli anni 2015 e 2016. In figura 4.12 viene riportata la colonna relativa a questa *feature* in cui, a titolo esemplificativo viene riportato l'1 Gennaio 2015, ovvero un giorno festivo.

	A	B	C	D	E	F	G
1	Data Ora	Prezzo	Total Load (MW)	Prezzo Gas	Solar Radiation	Wind Speed (h:10m, m/s)	Giorno Feriale o Festivo
2	2015-01-01 00:00:00	51,1	1000,25975	25,056	0	27,0851063	0
3	2015-01-01 01:00:00	49	945,4935	25,056	0	25,58794874	0
4	2015-01-01 02:00:00	39,1	902,282	25,056	0	24,28337999	0
5	2015-01-01 03:00:00	35,87	865,112	25,056	0	23,43482514	0
6	2015-01-01 04:00:00	33,4	841,468	25,056	0	19,53264867	0
7	2015-01-01 05:00:00	36,57	825,2005	25,056	0	16,87551447	0
8	2015-01-01 06:00:00	39,1	845,2095	25,056	0	15,0548481	0
9	2015-01-01 07:00:00	45,21	872,57575	25,056	8,404701599	12,3853235	0
10	2015-01-01 08:00:00	40	917,54525	25,056	47,7625933	11,82345162	0
11	2015-01-01 09:00:00	40,8	1001,6975	25,056	79,09251242	11,5167209	0
12	2015-01-01 10:00:00	38,8	1073,7455	25,056	99,07026576	10,90923622	0
13	2015-01-01 11:00:00	38,8	1138,248	25,056	107,527734	11,22610985	0
14	2015-01-01 12:00:00	38,85	1140,11125	25,056	106,1986385	12,1404663	0
15	2015-01-01 13:00:00	38,8	1054,81625	25,056	93,69401489	12,59738304	0
16	2015-01-01 14:00:00	38,85	965,79425	25,056	70,16826101	13,65990617	0
17	2015-01-01 15:00:00	49,95	965,3215	25,056	35,52908338	15,2973937	0

Figura 4.12 Giorno feriale o festivo

4.1.6 Ora e mese di tipo ciclico

Nelle *features* considerate finora non si è fatto alcun cenno all'informazione riguardante l'ora e il mese; tali informazioni sono però molto rilevanti, poiché ci si appresta a fare un'analisi oraria in cui la successione temporale dei dati ha molta importanza. In particolare, è bene mettere in evidenza la correlazione tra le ore 22.00 e 23.00 con le ore 1.00 e 2.00, essendo tra loro vicine dal punto di vista temporale ma distanti numericamente. L'idea è quindi quella di creare una variabile ciclica che sia in grado di evidenziare la vicinanza presente tra i primi e gli ultimi valori. Si riportano di seguito le espressioni utilizzate per ottenere questo risultato:

$$hourSin = \sin\left(2\pi * \frac{hour}{24}\right)$$

Utilizzando questa variabile però si associa lo stesso valore alla mezzanotte (ora 0) e a mezzogiorno (ora 12); quindi, per differenziare i due momenti, è necessario introdurre una seconda variabile in coseno (cioè sfalsata rispetto alla funzione seno di $\pi/2$) in modo che la loro combinazione porti ad un singolo valore, unico per ogni momento della giornata.

$$hourCos = \cos\left(2\pi * \frac{hour}{24}\right)$$

Lo stesso ragionamento si applica all'informazione relativa al mese, per tener conto della vicinanza temporale tra i mesi invernali di Novembre e Dicembre con Gennaio e Febbraio. In questo caso,

per rendere più fluido l'avanzamento del tempo, viene prima definita la variabile *monthFloat*, ottenuta aggiungendo a *month* il termine di ordine inferiore.

$$monthFloat = month + \frac{day}{24}$$

$$monthSin = \sin\left(2\pi * \frac{monthFloat}{12}\right)$$

$$monthCos = \cos\left(2\pi * \frac{monthFloat}{12}\right)$$

In figura 4.13 è riportata una rappresentazione completa di tutti i dati utilizzati nei modelli successivamente descritti.

	A	B	C	D	E	F	G	H	I	J	K
1	Data Ora	Prezzo	Total Load (MW)	Prezzo Gas	Solar Radiation	Wind Speed (h:10m, m/s)	Giorno Feriale o Festivo	hourSin	hourCos	monthSin	monthCos
2	2015-01-01 00:00:00	51,1	1000,25975	25,056	0	27,0851063	0	0	1	0,515038	0,857167
3	2015-01-01 01:00:00	49	945,4935	25,056	0	25,58794874	0	0,258819	0,965926	0,515038	0,857167
4	2015-01-01 02:00:00	39,1	902,282	25,056	0	24,28337999	0	0,5	0,866025	0,515038	0,857167
5	2015-01-01 03:00:00	35,87	865,112	25,056	0	23,43482514	0	0,707107	0,707107	0,515038	0,857167
6	2015-01-01 04:00:00	33,4	841,468	25,056	0	19,53264867	0	0,866025	0,5	0,515038	0,857167
7	2015-01-01 05:00:00	36,57	825,2005	25,056	0	16,87551447	0	0,965926	0,258819	0,515038	0,857167
8	2015-01-01 06:00:00	39,1	845,2095	25,056	0	15,0548481	0	1	6,12E-17	0,515038	0,857167
9	2015-01-01 07:00:00	45,21	872,57575	25,056	8,404701599	12,3853235	0	0,965926	-0,25882	0,515038	0,857167
10	2015-01-01 08:00:00	40	917,54525	25,056	47,7625933	11,82345162	0	0,866025	-0,5	0,515038	0,857167
11	2015-01-01 09:00:00	40,8	1001,6975	25,056	79,09251242	11,5167209	0	0,707107	-0,70711	0,515038	0,857167
12	2015-01-01 10:00:00	38,8	1073,7455	25,056	99,07026576	10,90923622	0	0,5	-0,86603	0,515038	0,857167
13	2015-01-01 11:00:00	38,8	1138,248	25,056	107,527734	11,22610985	0	0,258819	-0,96593	0,515038	0,857167
14	2015-01-01 12:00:00	38,85	1140,11125	25,056	106,1986385	12,1404663	0	1,22E-16	-1	0,515038	0,857167
15	2015-01-01 13:00:00	38,8	1054,81625	25,056	93,69401489	12,59738304	0	-0,25882	-0,96593	0,515038	0,857167
16	2015-01-01 14:00:00	38,85	965,79425	25,056	70,16826101	13,65990617	0	-0,5	-0,86603	0,515038	0,857167
17	2015-01-01 15:00:00	49,95	965,3215	25,056	35,52908338	15,2973937	0	-0,70711	-0,70711	0,515038	0,857167

Figura 4.13 Ora e mese di tipo ciclico

4.2 Il modello LSTM per la previsione oraria e giornaliera

La rete neurale LSTM utilizzata per effettuare la previsione del prezzo dell'energia elettrica è la stessa sia per la previsione oraria sia per la previsione giornaliera; si tratta di una rete costituita da due strati nascosti, il cui numero di neuroni viene fatto variare al fine di trovare la configurazione con performance migliori. Ogni strato nascosto è caratterizzato dalla funzione di attivazione ReLU e, dopo ognuno di essi, è presente uno strato di *Dropout*. L'unica differenza tra i due casi sta, necessariamente, nel numero di unità costituenti lo strato di output, si avrà infatti una singola unità per la previsione oraria e 24 unità per quella giornaliera. Nei due modelli, inoltre, viene applicata una funzione diversa per creare il *train* e il *test set*. Nella fase di compilazione viene specificata

come funzione di perdita l'errore quadratico medio (MSE) e come ottimizzatore l'algoritmo Adam. Al fine di valutare le prestazioni del modello al variare della sua complessità, sono stati considerati quattro diversi casi, ognuno dei quali prende in considerazione *features* differenti: il primo caso presenta come dati di input la potenza assorbita in MW e il prezzo giornaliero del gas in Euro/MWh; nel secondo caso vengono aggiunti ai precedenti anche i dati relativi al tempo ciclico e alla distinzione tra giorno feriale e festivo; il terzo caso prende in considerazione, oltre a tutte le precedenti, anche le informazioni riguardanti la radiazione solare e la velocità del vento, scalate opportunamente considerando la producibilità degli impianti fotovoltaici ed eolici presenti sul territorio; infine, il caso quattro, utilizza tutti i dati presenti nel caso tre a meno del prezzo del gas, per valutare come tale grandezza incida sulla previsione del prezzo dell'energia in Sardegna. Le analisi svolte sono schematizzabili come riportato nella tabella sottostante, dove nell'ultima colonna viene indicata la struttura della rete, ovvero il numero di unità costituenti lo strato di input, i due strati nascosti e lo strato di output.

Tabella 2 Analisi effettuate

	CASO	STRUTTURA
IPREVISIONE ORARIA	1	24-10-4-1
		24-25-12-1
		24-50-25-1
		24-80-40-1
	2	24-10-4-1
		24-25-12-1
		24-50-25-1
		24-80-40-1
	3	24-10-4-1
		24-25-12-1
		24-50-25-1
		24-80-40-1
	4	24-10-4-1
		24-25-12-1
		24-50-25-1
		24-80-40-1
PREVISIONE GIORNALIERA	1	24-25-12-24
		24-50-25-24
		24-80-40-24
		24-100-55-24
	2	24-25-12-24
		24-50-25-24
		24-80-40-24
		24-100-55-24
	3	24-25-12-24
		24-50-25-24
		24-80-40-24
		24-100-55-24
	4	24-25-12-24
		24-50-25-24
		24-80-40-24
		24-100-55-24

4.3 Previsione oraria del prezzo zonale dell'energia elettrica

La previsione oraria permette di ottenere il valore del prezzo dell'energia della 24-esima ora, dando in ingresso i dati relativi alle *features* delle 23 ore precedenti e dell'ora stessa di cui si vuole ottenere la previsione. La funzione utilizzata per creare il *train* e il *test set* è quella riportata nello script 9.

```
1 def create_dataset(x, y, time_steps=1):
2     xs, ys = [], []
3     for i in range(len(x) - time_steps):
4         v = x.iloc[i:(i + time_steps)].values
5         xs.append(v)
6         ys.append(y.iloc[i + time_steps])
7     return np.array(xs), np.array(ys)

8 TIME_STEPS = 24
9 X_train, y_train = create_dataset(scaled_train_ft, scaled_train_prezzo,
                                  Time_steps=TIME_STEPS)
```

Script 9 Funzione 'create_dataset' per previsione oraria

Come accennato in precedenza, i casi analizzati sono quattro. Per ogni caso sono state provate diverse configurazioni della rete neurale, andando a variare il numero di unità costituenti gli strati nascosti della rete (“struttura”). Il *train* del modello, con la medesima configurazione, viene effettuato più volte e, come si può notare, il risultato che si ottiene varia di volta in volta a causa della natura stocastica del problema. Durante la fase di addestramento e di validazione viene calcolata alla fine di ogni epoca la funzione di perdita, ovvero l'errore quadratico medio (MSE), per valutare l'andamento e l'efficacia del processo; viene data, inoltre, anche una rappresentazione grafica di tale valore in funzione delle epoche, al fine di valutare se il modello è soggetto ad *overfitting* o *underfitting*. Una volta terminata la fase di addestramento si è applicato il modello sia ai dati di *test* che ai dati di *train*, calcolando successivamente la radice dell'errore quadratico medio (RMSE) in modo tale da poter confrontare le prestazioni del modello nei vari casi e con le diverse configurazioni. Nelle seguenti tabelle vengono riportati i risultati ottenuti dalle analisi, sia durante la fase di addestramento che di test del modello.

- Caso 1:

Tabella 3 Caso 1 – Previsione Oraria

Struttura	Training		Validazione		RMSE TRAIN [Euro/MWh]	RMSE TEST [Euro/MWh]	Epoche
	MSE	MAE	MSE	MAE			
10-4	0,0096	0,0712	0,0163	0,1131	15,2007	24,2029	6
10-4	0,0090	0,0688	0,0166	0,1146	15,1517	25,4431	6
10-4	0,0093	0,0701	0,0182	0,1207	15,2631	25,7814	6
25-12	0,0095	0,0708	0,0153	0,1092	15,5694	25,3896	6
25-12	0,0092	0,0704	0,0183	0,1208	15,5091	26,6164	6
25-12	0,0092	0,0699	0,0168	0,1151	15,2172	25,5967	6
50-25	0,0086	0,0675	0,0154	0,1106	15,3299	26,6701	8
50-25	0,0087	0,0677	0,0153	0,1102	15,2072	26,2493	8
50-25	0,0086	0,0675	0,0171	0,1174	15,099	26,7391	8
80-45	0,0083	0,0656	0,0126	0,0992	14,1760	25,3183	8
80-45	0,0085	0,0668	0,0148	0,1082	14,6079	25,9450	8
80-45	0,0088	0,0684	0,0143	0,1057	15,5400	26,1161	8

- Caso 2:

Tabella 4 Caso 2 – Previsione Oraria

Struttura	Training		Validazione		RMSE TRAIN [Euro/MWh]	RMSE TEST [Euro/MWh]	Epoche
	MSE	MAE	MSE	MAE			
10-4	0,0099	0,0716	0,0150	0,1075	15,1509	22,8584	8
10-4	0,0054	0,0509	0,0062	0,0674	10,0868	22,1034	53
10-4	0,0066	0,0554	0,0083	0,0803	11,8857	21,1096	17
25-12	0,0057	0,0518	0,0055	0,0620	10,6592	19,4855	17
25-12	0,0059	0,0531	0,0060	0,0665	10,8429	19,126	14
25-12	0,0061	0,0541	0,0056	0,0631	11,0655	19,7725	11
50-25	0,0056	0,0513	0,0048	0,0575	10,7520	19,3767	13
50-25	0,0057	0,0517	0,0044	0,0542	10,6751	19,1655	14
50-25	0,0050	0,0483	0,0045	0,0507	10,2361	19,9469	29
80-45	0,0056	0,0511	0,0053	0,0608	10,8500	18,6162	14
80-45	0,0055	0,0508	0,0046	0,0558	10,7732	18,9493	14
80-45	0,0053	0,0498	0,0053	0,0606	10,6529	19,5516	17

- Caso 3:

Tabella 5 Caso 3 – Previsione Oraria

Struttura	Training		Validazione		RMSE TRAIN [Euro/MWh]	RMSE TEST [Euro/MWh]	Epoche
	MSE	MAE	MSE	MAE			
10-4	0,0061	0,0538	0,0047	0,0489	10,6498	17,994	30
10-4	0,0062	0,0547	0,0058	0,0615	10,8989	18,6258	21
10-4	0,0065	0,0560	0,0056	0,0628	10,9237	19,8300	12
25-12	0,0052	0,0501	0,0043	0,0488	9,8783	17,1214	22
25-12	0,0053	0,0497	0,0049	0,0481	10,6633	17,7327	22
25-12	0,0057	0,0516	0,0115	0,0688	11,6813	17,9324	14
50-25	0,0054	0,0501	0,0034	0,0450	10,2985	17,732	16
50-25	0,0055	0,0507	0,0039	0,0467	10,384	17,2617	13
50-25	0,0052	0,0494	0,0042	0,0487	10,3407	17,4542	16
80-45	0,0051	0,0487	0,0037	0,0445	10,3616	17,5003	19
80-45	0,0052	0,0491	0,0036	0,0470	10,2145	18,1298	16
80-45	0,0052	0,0489	0,0033	0,0432	10,3651	17,8831	18

- Caso 4:

Tabella 6 Caso 4 – Previsione Oraria

Struttura	Training		Validazione		RMSE TRAIN [Euro/MWh]	RMSE TEST [Euro/MWh]	Epoche
	MSE	MAE	MSE	MAE			
10-4	0,0067	0,0565	0,0073	0,0749	11,5279	19,1154	13
10-4	0,0070	0,0583	0,0089	0,0826	11,9012	20,3472	12
10-4	0,0059	0,0532	0,0053	0,0607	10,5122	19,0079	25
25-12	0,0055	0,0510	0,0046	0,0560	10,1971	17,9289	19
25-12	0,0057	0,0519	0,0053	0,0599	10,4214	18,4179	14
25-12	0,0058	0,0514	0,0057	0,0636	10,7161	18,8055	16
50-25	0,0055	0,0507	0,0054	0,0603	10,5483	18,9741	16
50-25	0,0055	0,0505	0,0066	0,0699	10,6427	19,6050	15
50-25	0,0055	0,0506	0,0064	0,0674	10,7296	19,8246	11
80-45	0,0053	0,0494	0,0053	0,0614	10,4743	18,9298	15
80-45	0,0052	0,0490	0,0059	0,0641	10,4072	19,3318	20
80-45	0,0053	0,0492	0,0062	0,0664	10,613	19,7758	16

Osservando l'andamento dell'RMSE calcolato sui dati di test, si nota come le performance del modello migliorino all'aumentare della complessità del modello stesso, ovvero maggiore è il

numero di *features* che vengono considerate; la rete neurale che presenta risultati migliori, infatti, è quella del caso 3, in cui vengono dati in input i dati relativi a tutte le grandezze sopra descritte, con struttura 50-25. Di seguito verrà dunque analizzato solamente questo caso, aiutandosi con l'ausilio di diverse rappresentazioni grafiche. Si riporta, innanzitutto, il confronto tra l'andamento della funzione di perdita durante la fase di *train* e di validazione (figura 4.14): con il progredire del processo di addestramento, ovvero all'aumentare del numero delle epoche, si ha una diminuzione dell'errore quadratico medio, ciò significa che la rete neurale sta "imparando" correttamente le relazioni presenti nei dati che le vengono forniti in input; la convergenza tra le due curve, inoltre, suggerisce che il modello sarà in grado di riconoscere tali relazioni anche nei dati di test.

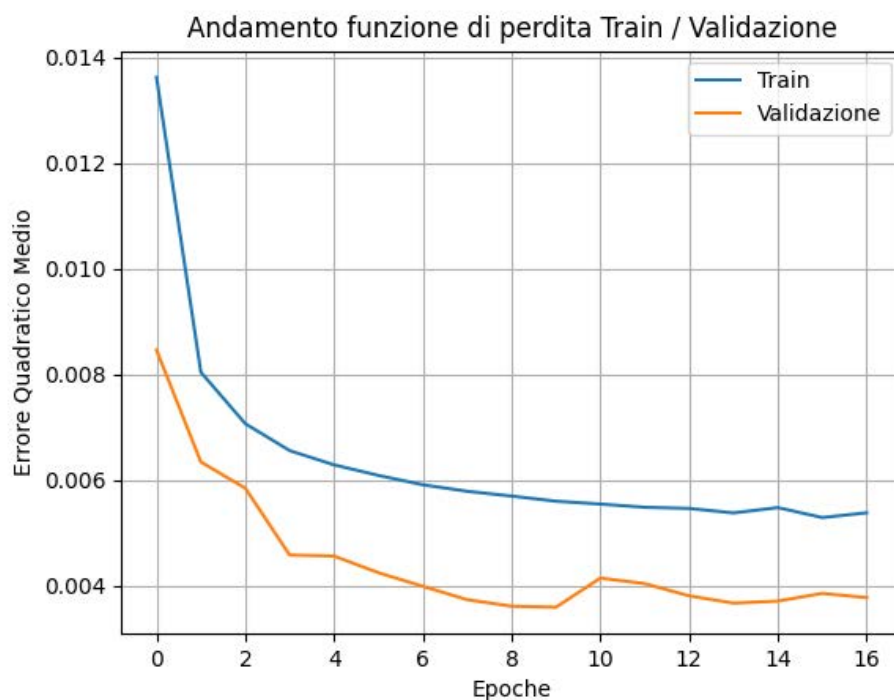


Figura 4.14 Analisi oraria: confronto funzione di perdita Train/Validazione

In figura 4.15 vengono invece rappresentati i valori del prezzo di *train* (relativi all'anno 2015) e i valori di *test* (relativi all'anno 2016), a cui si sovrappongono i valori previsti mediante il modello LSTM.

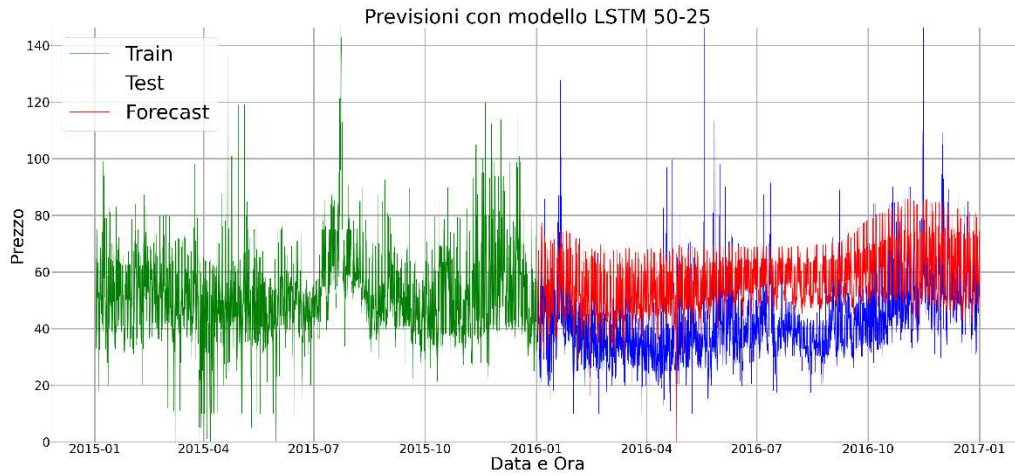


Figura 4.15 Analisi oraria: andamento valore prezzo Train/Test/Previsioni

Dato l'elevato numero di dati utilizzati, si è effettuato un grafico con i soli dati di test, per avere una migliore percezione della differenza rispetto alle previsioni ottenute. In figura 4.16 si riporta, inoltre, tale differenza, ovvero l'errore commesso.

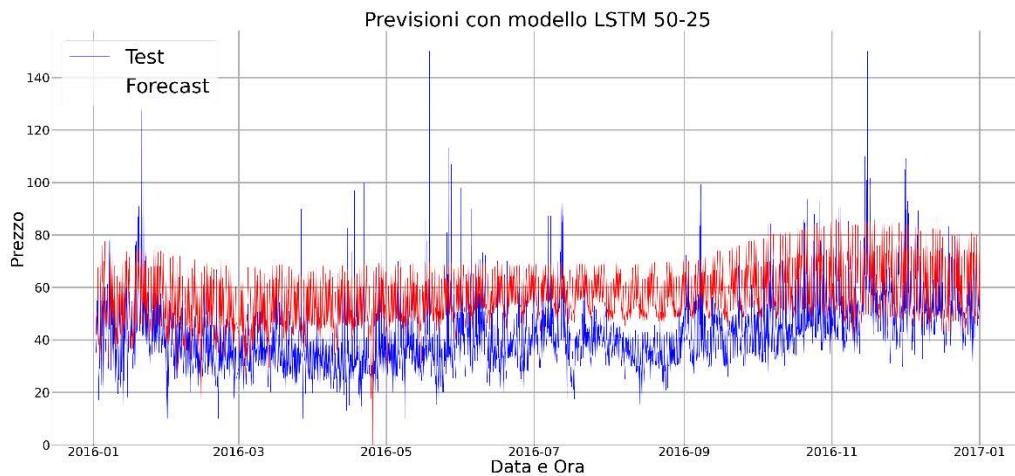


Figura 4.16 Analisi oraria: confronto valori reali e previsti

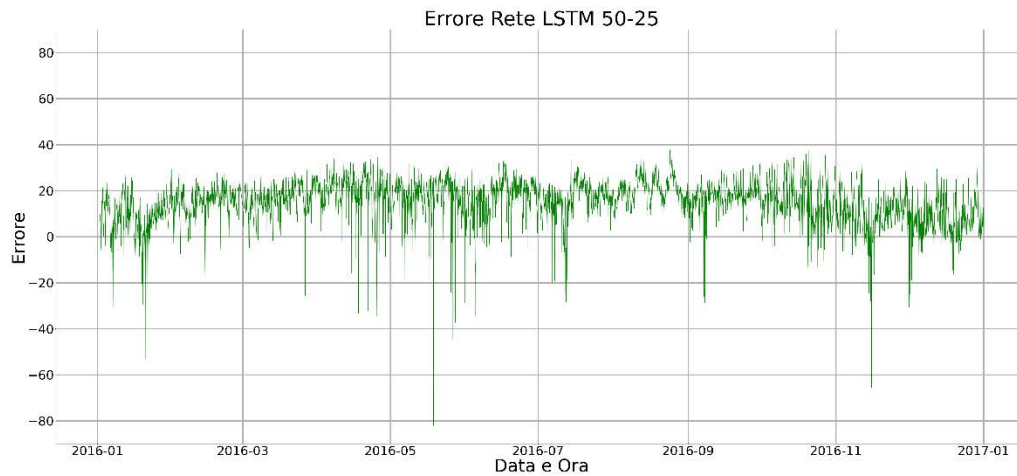


Figura 4.17 Analisi oraria: errore commesso nella previsione (differenza tra valore previsto e valore reale)

In generale, osservando il grafico di figura 4.16, si può notare come il modello sovrastimi il prezzo dell'energia elettrica; si è quindi calcolato il valore medio dell'errore, che in tal caso è pari a 14,9193 Euro/MWh, e si è supposto che il modello compia tale errore di previsione per ogni prezzo stimato. Si è ricavato, dunque, il valore del *trend* di prezzo eseguendo la sottrazione tra i valori previsti e l'errore medio; dopo aver compiuto tale operazione si è calcolato l'RMSE sui valori di *trend*, ottenendo un risultato pari a 9,0622 Euro/MWh, valore sensibilmente minore rispetto ai 17,446 Euro/MWh dell'RMSE di *test*. In figura 4.18 viene riportato l'andamento del *trend*, ottenuto slittando verso il basso il grafico di figura 4.16 di un valore pari all'errore medio.

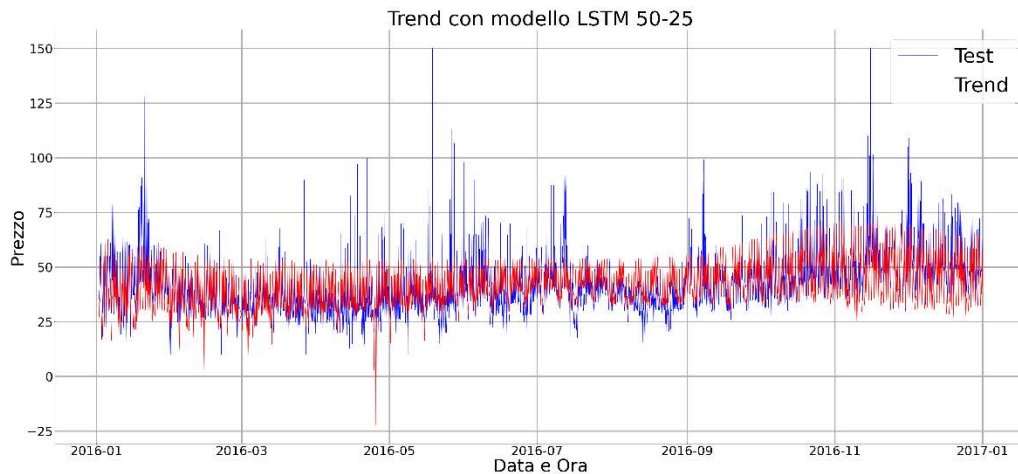


Figura 4.18 Analisi oraria: confronto valori reali e trend di prezzo

Per poter apprezzare maggiormente i risultati ottenuti ed evidenziare come le previsioni seguano effettivamente l'andamento dei dati reali, si è realizzato nuovamente il grafico precedente prendendo in considerazione solamente il mese di Dicembre 2016.

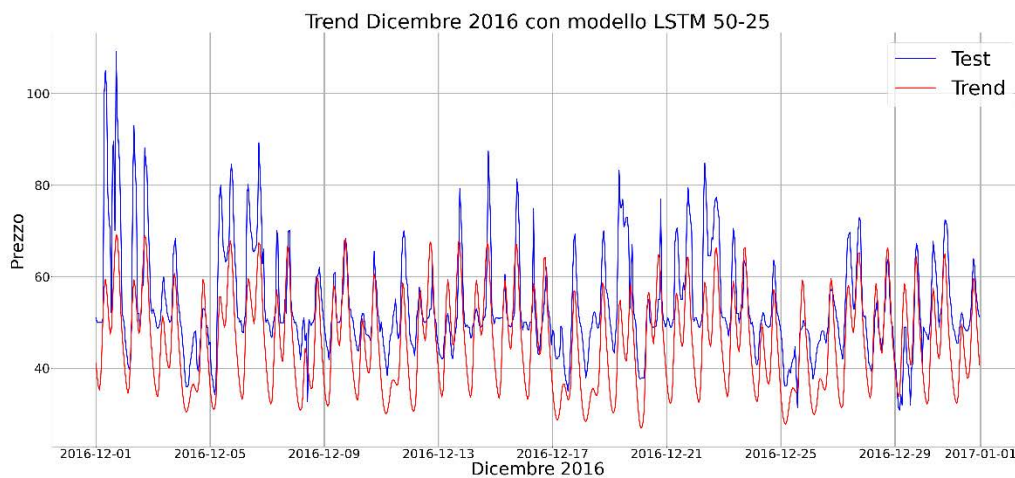


Figura 4.19 Analisi oraria: confronto valori reali e trend di prezzo Dicembre 2016

Un'analisi più approfondita dell'errore commesso, ovvero della differenza tra i valori di prezzo reali e i valori predetti, può essere fatta osservando le seguenti figure. Dalla figura 4.20, in cui viene rappresentata la distribuzione dell'errore mediante un istogramma, si ottiene la conferma che il

modello sovrastimi il prezzo dell'energia, poiché la maggior parte degli errori è superiore allo zero. Tale rappresentazione viene effettuata anche per il *trend* (figura 4.21) e, osservando quest'ultimo grafico, si evince come la distribuzione dell'errore cambi solo leggermente e si sposti a sinistra di un valore pari all'errore medio.

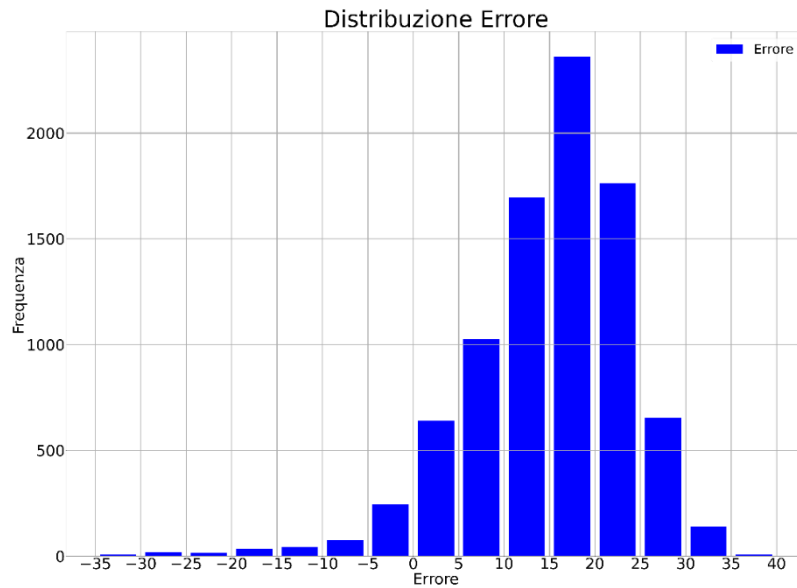


Figura 4.20 Analisi oraria: distribuzione dell'errore

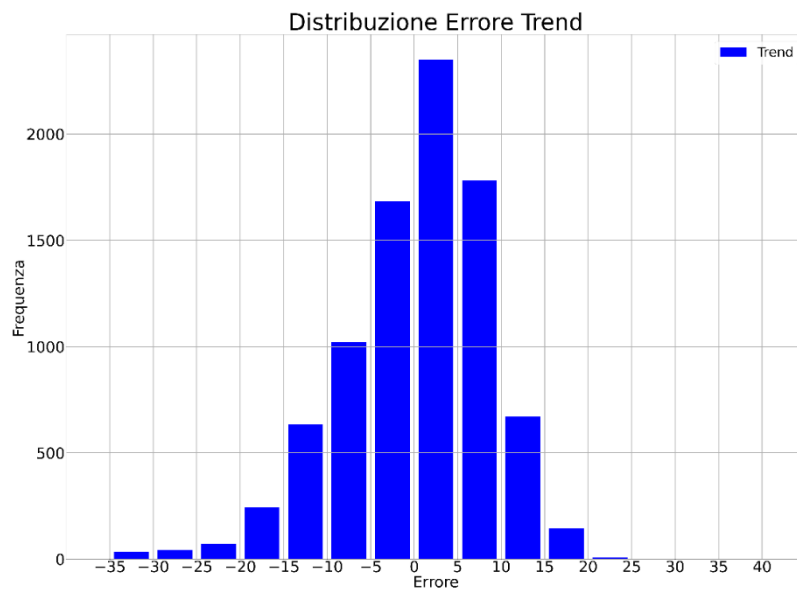


Figura 4.21 Analisi oraria: distribuzione dell'errore di trend

La figura 4.22, invece, esegue una comparazione tra i valori reali (asse x) e i valori predetti (asse y) del prezzo dell'energia; se le previsioni del modello fossero prive di errore, effettuando questo confronto, si otterrebbe una retta coincidente con la bisettrice del grafico, ma, come si evince dalla figura riportata, ciò non accade. È interessante notare però come i diversi punti si addensino per lo più parallelamente alla bisettrice, ciò permette nuovamente di concludere che le previsioni effettuate dalla rete neurale siano affette da un errore caratterizzato da una componente costante (errore medio) e da una componente casuale, dipendente da aspetti non presi in considerazione nel modello. In figura 4.23, viene eseguita la stessa rappresentazione considerando il *trend* del prezzo; in questo caso si è eliminata la componente costante dell'errore commesso dal modello e si avrà, quindi, maggiore densità in corrispondenza della bisettrice del grafico, nonostante rimanga elevata la componente casuale dell'errore.

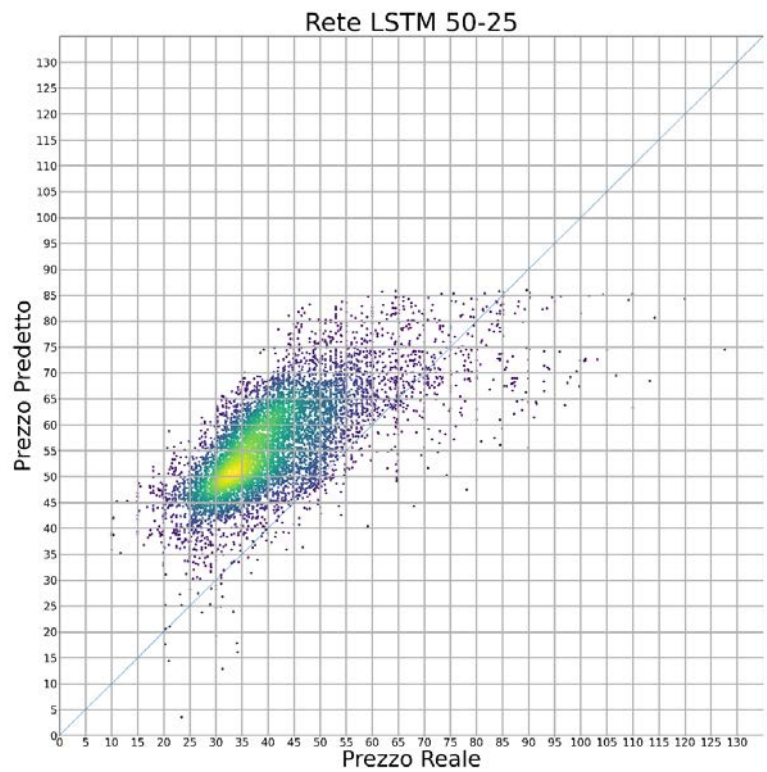


Figura 4.22 Analisi oraria: Prezzo reale - Prezzo predetto

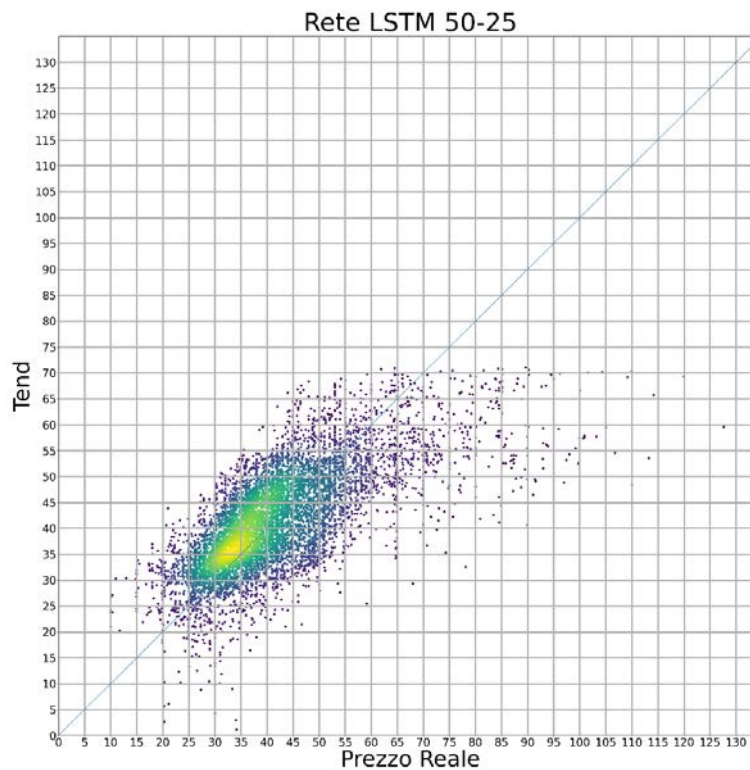


Figura 4.23 Analisi oraria: Prezzo reale - Trend

Vengono riportati, infine, i valori dell'RMSE calcolati su base mensile sia per le previsioni ottenute dal modello sia per i valori del *trend*; osservando questi dati si può notare come, una volta sottratto il valore medio, si abbia per ogni mese un generale miglioramento del risultato ottenuto.

Tabella 7 RMSE Mensili – Previsione Oraria

MESE	RMSE VALORI PREDETTI ORARI [Euro/MWh]	RMSE VALORI TREND ORARI [Euro/MWh]
Gennaio	12,4182	11,1006
Febbraio	16,2489	6,1047
Marzo	17,4522	6,3769
Aprile	21,3276	9,8538

Maggio	19,4351	10,4555
Giugno	18,5653	8,6481
Luglio	17,9300	8,3235
Agosto	21,2721	7,8733
Settembre	18,9042	7,4636
Ottobre	18,1837	8,8777
Novembre	13,2797	10,3612
Dicembre	10,6277	11,2752
VALORE ANNUO	14,9193	9,0622

4.4 Previsione giornaliera del prezzo zonale dell'energia elettrica

La previsione giornaliera permette di ottenere il valore del prezzo dell'energia di un'intera giornata, dando in ingresso i dati relativi alle *features* delle 24 ore precedenti. La funzione utilizzata per creare il *train* e il *test set* è quella riportata nello script 10.

```

1 def create_dataset(x, y):
2     xs, ys = list(), list()
3     k = np.arange(0, len(x), 24)
4     for j in k:
5         xs.append(x.iloc[j:(j+24), :9])
6         ys.append(y.iloc[j:(j+24), -1])
7     return array(xs), array(ys)

8 X_train, y_train = create_dataset(scaled_train_ft,
                                   scaled_train_prezzo)

```

Script 10 Funzione 'create_dataset' per previsione giornaliera

L'analisi svolta in questo caso è la stessa eseguita per il caso orario: si sono analizzati, infatti, i quattro diversi casi, calcolando i medesimi parametri e rappresentando graficamente le stesse grandezze. Verranno di seguito riportate le tabelle frutto di questa analisi.

- Caso 1:

Tabella 8 Caso 1 – Previsione Giornaliera

Struttura	Training		Validazione		RMSE TRAIN [Euro/MWh]	RMSE TEST [Euro/MWh]	Epoche
	MSE	MAE	MSE	MAE			
25-12	0,0067	0,0568	0,0131	0,1024	13,5301	23,8997	8
25-12	0,0062	0,0546	0,0150	0,1103	13,7443	25,3081	8
25-12	0,0064	0,0552	0,0156	0,1123	13,8313	25,2238	8
50-25	0,0065	0,0558	0,0163	0,1142	14,7204	25,6349	8
50-25	0,0067	0,0566	0,0158	0,1128	14,5866	25,7871	8
50-25	0,0065	0,0555	0,0160	0,1139	14,2866	25,6867	8
80-40	0,0065	0,0559	0,0091	0,0817	12,6671	21,1109	8
80-40	0,0062	0,0552	0,0141	0,1057	14,4192	25,1593	8
80-40	0,0065	0,0559	0,0150	0,1094	14,4289	25,1855	8
100-55	0,0063	0,0550	0,0151	0,1095	14,4707	25,3248	8
100-55	0,0066	0,0558	0,0144	0,1062	14,3257	24,7588	8
100-55	0,0064	0,0559	0,0138	0,1039	14,3006	24,6734	8

- Caso 2:

Tabella 9 Caso 2 – Previsione Giornaliera

Struttura	Training		Validazione		RMSE TRAIN [Euro/MWh]	RMSE TEST [Euro/MWh]	Epoche
	MSE	MAE	MSE	MAE			
25-12	0,0055	0,0507	0,0073	0,0732	10,9416	19,313	12
25-12	0,0059	0,0528	0,0068	0,0701	11,2693	20,6813	8
25-12	0,0056	0,0514	0,0059	0,0634	10,7781	18,8202	10
50-25	0,0058	0,0520	0,0085	0,0801	11,5201	20,5448	8
50-25	0,0058	0,0530	0,0068	0,0700	11,4249	19,4025	8
50-25	0,0053	0,0498	0,0069	0,0713	10,9325	19,8261	13
80-40	0,0055	0,0505	0,0068	0,0697	11,2541	19,8223	11
80-40	0,0055	0,0505	0,0074	0,0736	11,4973	20,7591	10
80-40	0,0058	0,0524	0,0073	0,0741	11,22	19,5839	8
100-55	0,0055	0,0511	0,0080	0,0779	11,3802	20,5652	10
100-55	0,0053	0,0498	0,0089	0,0701	11,4229	19,9522	18
100-55	0,0057	0,0528	0,0071	0,0722	11,3609	20,1509	8

- Caso 3:

Tabella 10 Caso 3 – Previsione Giornaliera

Struttura	Training		Validazione		RMSE TRAIN [Euro/MWh]	RMSE TEST [Euro/MWh]	Epoche
	MSE	MAE	MSE	MAE			
25-12	0,0058	0,0527	0,0059	0,0644	11,0144	19,3932	8
25-12	0,0061	0,0539	0,0079	0,0781	11,4622	19,9172	8
25-12	0,0059	0,0524	0,0052	0,0592	10,9431	18,8333	8
50-25	0,0056	0,0513	0,0057	0,0636	10,8777	18,7671	10
50-25	0,0054	0,0501	0,0069	0,0709	11,5006	20,2805	11
50-25	0,0056	0,0519	0,0059	0,0625	10,6242	18,7409	13
80-40	0,0056	0,0522	0,0079	0,0758	11,5395	20,1432	9
80-40	0,0056	0,0513	0,0070	0,0727	11,1613	19,7384	10
80-40	0,0054	0,0500	0,0063	0,0665	10,9854	18,8727	13
100-55	0,0056	0,0515	0,0054	0,0611	11,1986	19,4048	9
100-55	0,0056	0,0510	0,0067	0,0687	11,1411	19,4463	10
100-55	0,0057	0,0520	0,0070	0,0719	11,3471	20,1184	8

- Caso 4:

Tabella 11 Caso 4 – Previsione Giornaliera

Struttura	Training		Validazione		RMSE TRAIN [Euro/MWh]	RMSE TEST [Euro/MWh]	Epoche
	MSE	MAE	MSE	MAE			
25-12	0,0052	0,0497	0,0101	0,0891	11,2302	19,9106	13
25-12	0,0057	0,0515	0,0082	0,0800	10,9562	19,6947	10
25-12	0,0055	0,0516	0,0089	0,0836	10,9792	19,3314	13
50-25	0,0059	0,0527	0,0068	0,0717	11,2819	20,3696	8
50-25	0,0056	0,0518	0,0077	0,0765	11,2423	19,8260	9
50-25	0,0059	0,0528	0,0092	0,0848	11,9182	22,1055	8
80-40	0,0049	0,0479	0,0092	0,0853	11,0552	20,3221	22
80-40	0,0053	0,0496	0,0081	0,0782	11,1551	20,0105	14
80-40	0,0059	0,0531	0,0083	0,0791	11,6242	20,6512	8
100-55	0,0055	0,0511	0,0083	0,0795	11,4963	20,3931	9
100-55	0,0056	0,0509	0,0093	0,0848	11,6341	20,9368	11
100-55	0,0054	0,0503	0,0076	0,0760	11,1692	20,2005	12

Anche per questa analisi le performance migliori si ottengono nel caso 3, quando vengono considerate tutte le *features* in input, ma con configurazione 25-12. Si prosegue dunque

considerando solamente questo caso, riportando innanzitutto l'andamento della funzione di perdita durante la fase di *train* e di validazione (figura 4.24): con il progredire del processo di addestramento, ovvero all'aumentare del numero delle epoche, si ha una diminuzione dell'errore quadratico medio, ciò significa che la rete neurale sta "imparando" correttamente le relazioni presenti nei dati che le vengono forniti in input; la convergenza tra le due curve, inoltre, suggerisce che il modello sarà in grado di riconoscere tali relazioni anche nei dati di *test*.

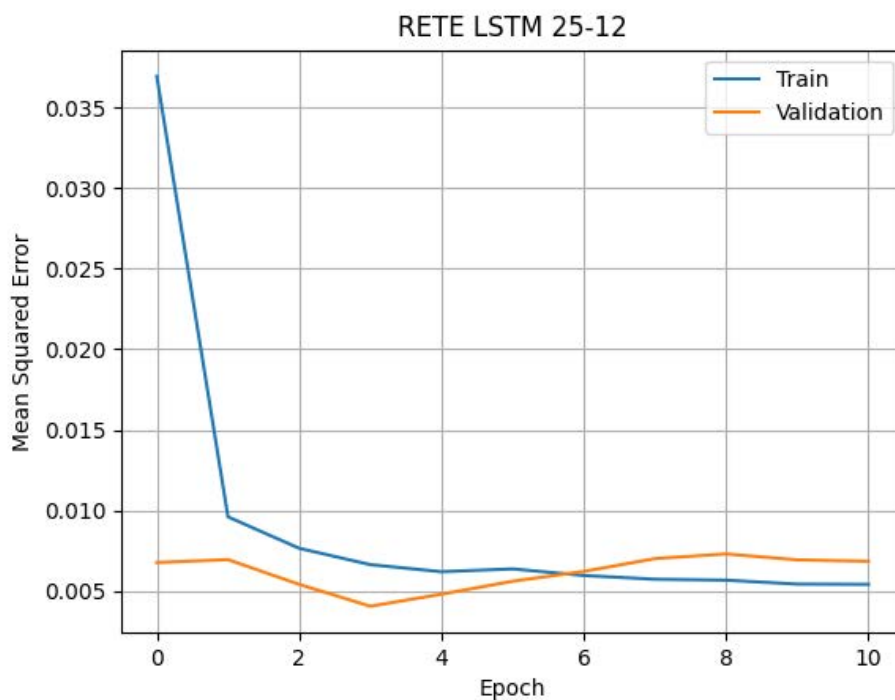


Figura 4.24 Analisi giornaliera: confronto funzione di perdita Train/Validazione

In figura 4.25 vengono invece rappresentati i valori del prezzo di *train* (relativi all'anno 2015) e i valori di *test* (relativi all'anno 2016), a cui si sovrappongono i valori previsti mediante il modello LSTM.

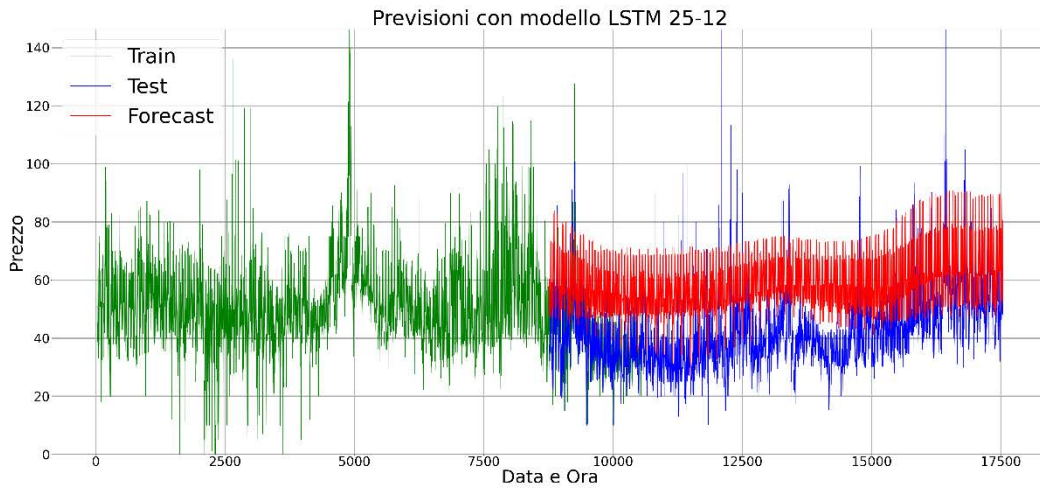


Figura 4.25 Analisi giornaliera: andamento valore prezzo Train/Test/Previsioni

Dato l'elevato numero di dati utilizzati, si è effettuato un grafico con i soli dati di test, per avere una migliore percezione della differenza rispetto alle previsioni ottenute. In figura 4.27 si riporta, inoltre, tale differenza, ovvero l'errore commesso.

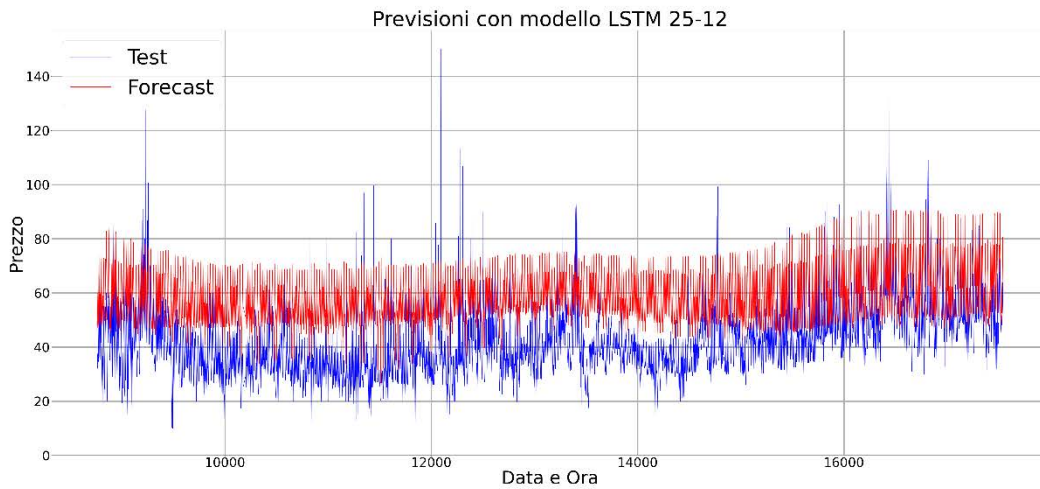


Figura 4.26 Analisi giornaliera: confronto valori reali e previsti

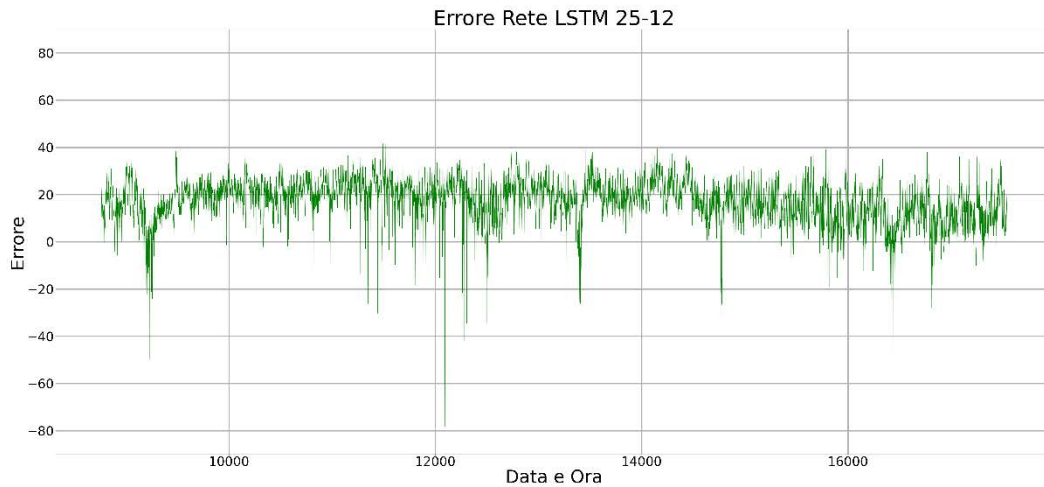


Figura 4.27 Analisi giornaliera: errore commesso nella previsione (differenza tra valore previsto e valore reale)

In generale, osservando il grafico di figura 4.26, si può notare come il modello sovrastimi il prezzo dell'energia elettrica; si è quindi calcolato il valore medio dell'errore, che in tal caso è pari a 17,1131 Euro/MWh, e si è supposto che il modello compia tale errore di previsione per ogni prezzo stimato. Si è ricavato, dunque, il valore del *trend* di prezzo eseguendo la sottrazione tra i valori previsti e l'errore medio; dopo aver compiuto tale operazione si è calcolato l'RMSE sui valori di *trend*, ottenendo un risultato pari a 9,0038 Euro/MWh, valore sensibilmente minore rispetto ai 19,1519 Euro/MWh dell'RMSE di *test*. In figura 4.28 viene riportato l'andamento del *trend*, ottenuto slittando verso il basso il grafico di figura 4.26 di un valore pari all'errore medio.

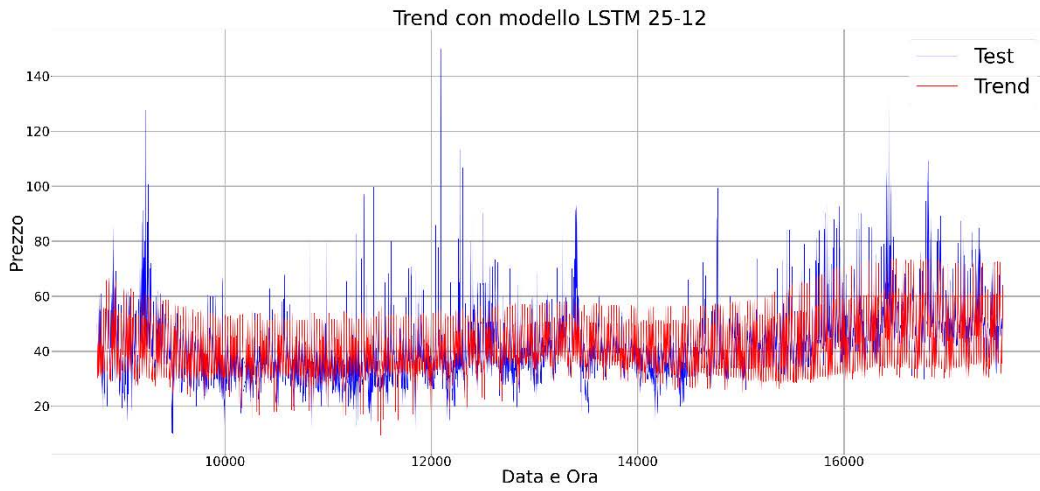


Figura 4.28 Analisi giornaliera: confronto valori reali e trend di prezzo

Per poter apprezzare maggiormente i risultati ottenuti ed evidenziare come le previsioni seguano effettivamente l'andamento dei dati reali, si è realizzato nuovamente il grafico precedente prendendo in considerazione solamente il mese di Dicembre 2016.

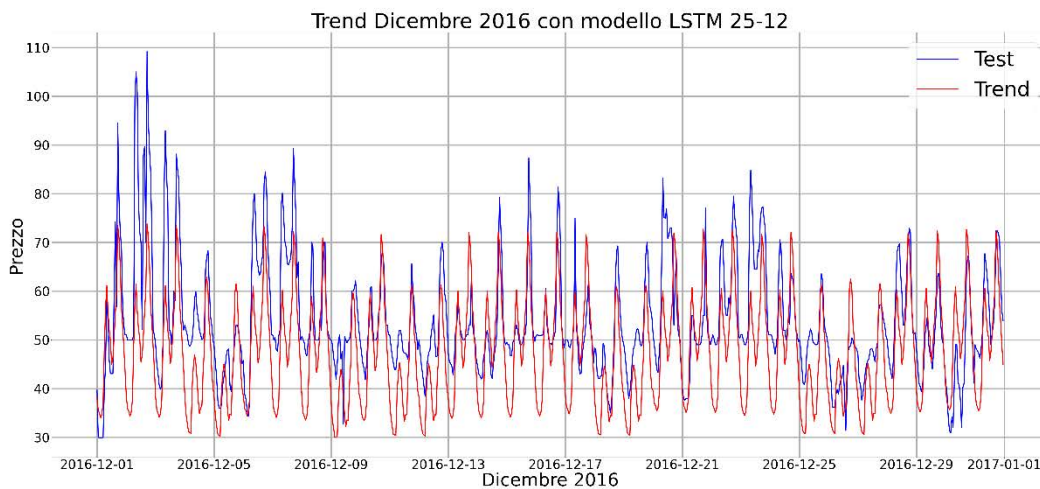


Figura 4.29 Analisi giornaliera: confronto valori reali e trend di prezzo Dicembre 2016

Un'analisi più approfondita dell'errore commesso, ovvero della differenza tra i valori di prezzo reali e i valori predetti, può essere fatta osservando le seguenti figure. Dalla figura 4.30, in cui viene rappresentata la distribuzione dell'errore mediante un istogramma, si ottiene la conferma che il

modello sovrastimi il prezzo dell'energia, poiché la maggior parte degli errori è superiore allo zero. Tale rappresentazione viene effettuata anche per il *trend* (figura 4.31) e, osservando quest'ultimo grafico, si evince come la distribuzione dell'errore cambi solo leggermente e si sposti a sinistra di un valore pari all'errore medio.

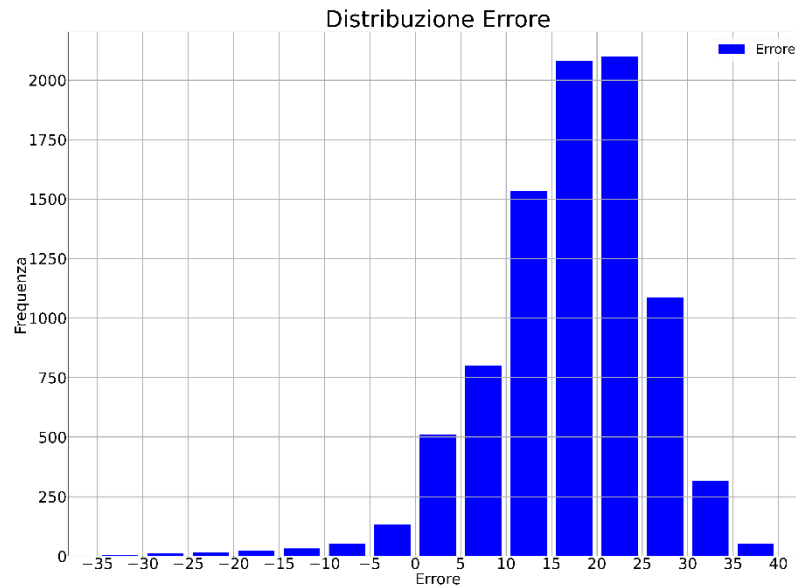


Figura 4.30 Analisi giornaliera: distribuzione dell'errore

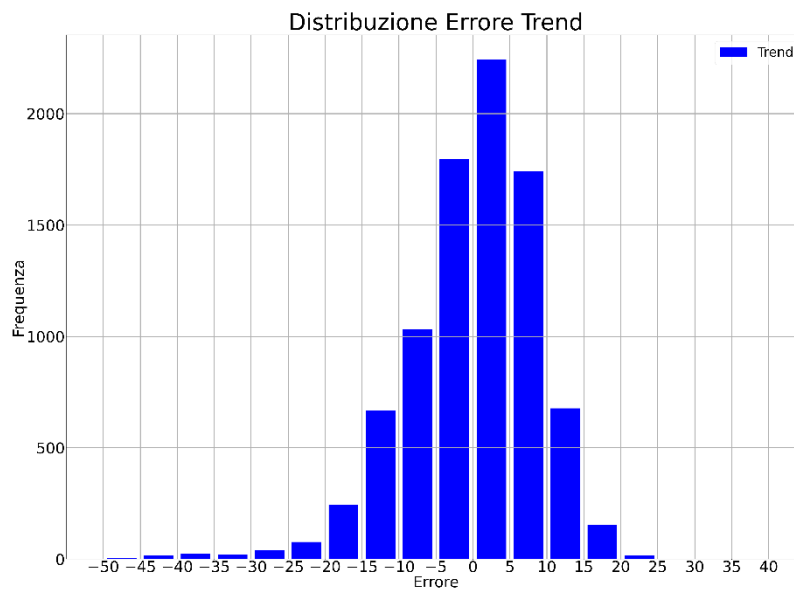


Figura 4.31 Analisi giornaliera: distribuzione dell'errore di trend

La figura 4.32, invece, esegue una comparazione tra i valori reali (asse x) e i valori predetti (asse y) del prezzo dell'energia; se le previsioni del modello fossero prive di errore, effettuando questo confronto, si otterrebbe una retta coincidente con la bisettrice del grafico, ma, come si evince dalla figura riportata, ciò non accade. È interessante notare però come i diversi punti si addensino per lo più parallelamente alla bisettrice, ciò permette nuovamente di concludere che le previsioni effettuate dalla rete neurale siano affette da un errore caratterizzato da una componente costante (errore medio) e da una componente casuale, dipendente da aspetti non presi in considerazione nel modello. In figura 4.33, viene eseguita la stessa rappresentazione considerando il *trend* del prezzo; in questo caso si è eliminata la componente costante dell'errore commesso dal modello e si avrà, quindi, maggiore densità in corrispondenza della bisettrice del grafico, nonostante rimanga elevata la componente casuale dell'errore.

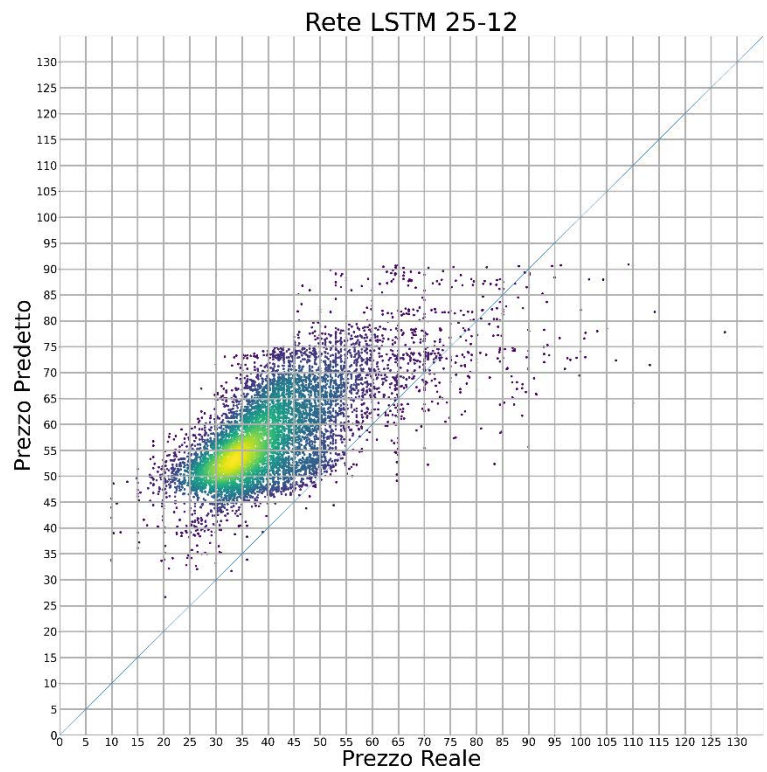


Figura 4.32 Analisi giornaliera: Prezzo reale - Prezzo predetto

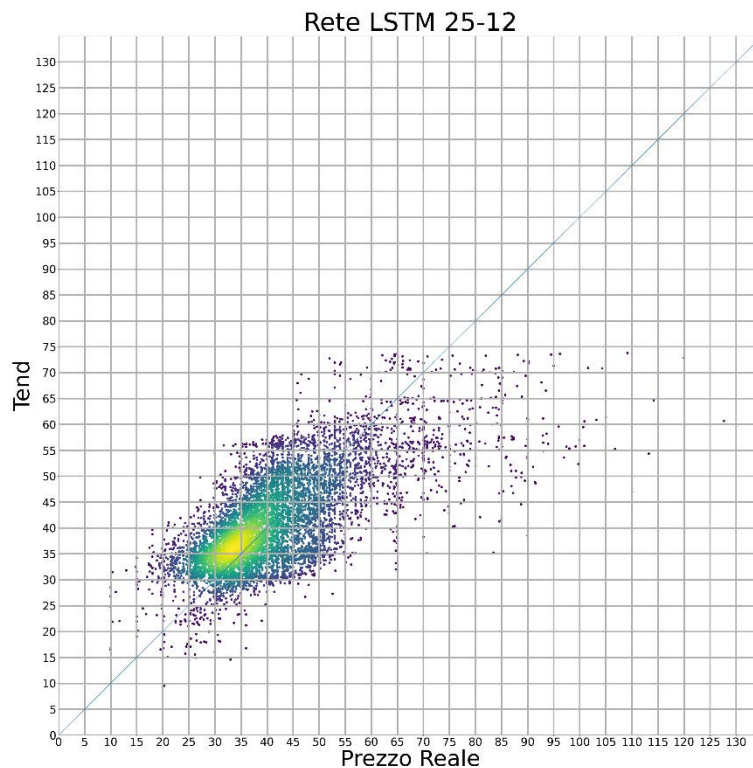


Figura 4.33 Analisi giornaliera: Prezzo reale - Trend

Vengono riportati, infine, i valori dell'RMSE calcolati su base mensile sia per le previsioni ottenute dal modello sia per i valori del *trend*; osservando questi dati si può notare come, una volta sottratto il valore medio, si abbia per ogni mese un generale miglioramento del risultato ottenuto.

Tabella 12 RMSE Mensili – Previsione Giornaliera

MESE	RMSE VALORI PREDETTI GIORNALIERI [Euro/MWh]	RMSE VALORI TREND ORARI [Euro/MWh]
Gennaio	16,7976	10,0115
Febbraio	20,8298	5,8315
Marzo	21,2981	6,6638
Aprile	23,6373	9,3473
Maggio	21,3722	10,4831
Giugno	20,5286	9,0333

Luglio	20,0737	8,6511
Agosto	22,7481	7,5327
Settembre	17,4652	7,6817
Ottobre	15,6856	9,0729
Novembre	14,5364	11,0379
Dicembre	13,6003	10,8068
VALORE ANNUO	19,2957	9,0038

Capitolo 5 – Conclusioni

5.1 Confronto tra analisi oraria e giornaliera

Le analisi svolte finora hanno permesso di concludere che, in generale, le performance dei modelli migliorano all'aumentare del numero delle *features* considerate. Nel riportare i risultati, infatti, si è preso in considerazione il caso tre, nel quale si tiene conto dell'influenza sul prezzo dell'energia di tutti i parametri descritti nel capitolo precedente. Per quanto riguarda il prezzo del gas e la sua capacità di influire sul prezzo dell'energia, da entrambi i risultati del caso quattro si può notare come, anche se in modo lieve, la relazione tra le due grandezze incida sull'abilità predittiva del modello. Successivamente si è eseguito un confronto tra i valori mensili dell'RMSE, per valutare effettivamente quale dei due modelli presenti performance migliori. In tabella 13 si riporta la differenza tra l'RMSE mensile dei valori predetti nell'analisi giornaliera e oraria; nella tabella successiva (tabella 14) si esegue la stessa operazione per i valori dell'RMSE calcolato sul *trend*.

Tabella 13 Confronto RMSE analisi oraria e giornaliera (valori predetti)

MESE	RMSE VALORI PREDETTI ORARI [Euro/MWh]	RMSE VALORI PREDETTI GIORNALIERI [Euro/MWh]	DIFFERENZA (VALORI GIORNALIERI - VALORI ORARI)
Gennaio	12,4182	16,7976	4,3794
Febbraio	16,2489	20,8298	4,5809
Marzo	17,4522	21,2981	3,8459
Aprile	21,3276	23,6373	2,3097
Maggio	19,4351	21,3722	1,9371
Giugno	18,5653	20,5286	1,9633

Luglio	17,9300	20,0737	2,1437
Agosto	21,2721	22,7481	1,476
Settembre	18,9042	17,4652	-1,439
Ottobre	18,1837	15,6856	-2,4981
Novembre	13,2797	14,5364	1,2567
Dicembre	10,6277	13,6003	2,9726
VALORE ANNUO	17,446	19,2957	1,8497

Tabella 14 Confronto RMSE analisi oraria e giornaliera (trend)

MESE	RMSE VALORI TREND ORARI [Euro/MWh]	RMSE VALORI TREND GIORNALIERI [Euro/MWh]	DIFFERENZA (VALORI GIORNALIERI - VALORI ORARI)
Gennaio	11,1006	10,0115	-1,0891
Febbraio	6,1047	5,8315	-0,2732
Marzo	6,3769	6,6638	0,2869
Aprile	9,8538	9,3473	-0,5065
Maggio	10,4555	10,4831	0,0276
Giugno	8,6481	9,0333	0,3852
Luglio	8,3235	8,6511	0,3276
Agosto	7,8733	7,5327	-0,3406
Settembre	7,4636	7,6817	0,2181
Ottobre	8,8777	9,0729	0,1952
Novembre	10,3612	11,0379	0,6767
Dicembre	11,2752	10,8068	-0,4684
VALORE ANNUO	9,0622	9,0038	-0,0584

Osservando i risultati della differenza nella prima tabella si è portati ad affermare che, globalmente, il modello costruito per la previsione oraria sia più efficiente di quello per la previsione giornaliera; in quest'ultimo caso, infatti, l'RMSE presenta valori maggiori e ciò sta a significare che la capacità predittiva del modello è minore. Concentrandosi invece sulla seconda tabella, si può concludere che, a meno di un errore costante, i due modelli presentino performance molto simili, poiché la differenza assoluta dell'RMSE di *trend* non supera quasi mai l'unità.

5.2 Considerazioni finali e possibili sviluppi futuri

I modelli sviluppati e descritti finora acquistano un ruolo importante se inseriti nell'ottica di una programmazione della produzione: per impianti come, ad esempio, quelli termoelettrici o idroelettrici ad accumulo, in cui la produzione può essere pianificata e diversificata nell'arco della giornata, può essere fondamentale conoscere l'andamento del prezzo dell'energia elettrica al fine di massimizzare il ritorno economico derivante dalla vendita dell'energia prodotta. Risulterà infatti conveniente concentrare una maggior produzione durante le ore in cui si registrano i valori più alti del prezzo e ridurla al minimo quando il prezzo si abbassa; in tal senso, inoltre, è necessario tener conto dei vincoli di produzione, che possono essere legati sia all'immissione di potenza in rete sia al tipo di impianto e all'ambiente in cui si inserisce. Tali ragionamenti sono necessariamente applicabili solo ad impianti che prendono parte al mercato libero dell'energia vendendo la loro produzione a prezzo variabile, poiché, in caso di vendita a prezzo fisso, la programmazione della produzione non porterebbe alcun beneficio. Il prezzo dell'energia elettrica è una grandezza, come si è visto, influenzata da molti fattori, alcuni dei quali non sono stati presi in considerazione in questa trattazione; in questo caso, per ottenere un modello più preciso e affidabile, si potrebbe aggiungere, oltre alle *features* utilizzate, anche la saturazione della connessione elettrica tra Sardegna e penisola. L'introduzione di questo parametro permetterebbe, infatti, di considerare l'influenza che il mercato elettrico della zona centro e centro-nord dell'Italia esercita su quello sardo. In tal senso, si riuscirebbe ad esprimere con maggior precisione la relazione esistente tra prezzo del gas e prezzo dell'energia nella sezione del mercato elettrico della Sardegna: l'andamento del prezzo del gas, infatti, ha un'influenza diretta e consistente solamente sul prezzo dell'energia nelle zone del nord, centro e sud Italia, mentre influenza quello delle isole solo indirettamente, attraverso le interconnessioni elettriche esistenti. Effettuare una stima del prezzo dell'energia non affetta da alcun errore risulta in ogni caso molto complicato, poiché, come si è spiegato nel capitolo due, si tratta di una grandezza caratterizzata da un'elevata volatilità e influenzata da fattori il più delle volte imprevedibili; i modelli realizzati, seppur affetti da errori, possono risultare utili per determinare l'andamento generale di tale grandezza e offrono, in conclusione, uno strumento utile per effettuare scelte più consapevoli e convenienti.

Bibliografia

- [1] R. Giannetti, “Il servizio elettrico dai sistemi regionali alla liberalizzazione”, *Treccani*, 2013, https://www.treccani.it/enciclopedia/il-servizio-elettrico-dai-sistemi-regionali-alla-liberalizzazione_%28Il-Contributo-italiano-alla-storia-del-Pensiero:-Tecnica%29/
- [2] A. Mandro, “Analisi e previsione del Prezzo Unico Nazionale dell’energia elettrica”, 2014/2015
- [3] S. Benedettini, C. Stagnaro, “Quando l’Italia era capace di aprirsi al mercato e di correre insieme all’Europa”, *Il Foglio quotidiano*, 16 Marzo 2019
- [4] “Vademecum della Borsa Elettrica Italiana”, GME, 28 Ottobre 2009, <https://www.google.com/search?client=firefox-b-d&q=gme+vademecum+borsa+elettrica>
- [5] “Acquirente Unico”, *Wikipedia - The Free Encyclopedia*, 8 Novembre 2020, https://it.wikipedia.org/wiki/Acquirente_Unico
- [6] “CIP6”, *Wikipedia - The Free Encyclopedia*, 22 Maggio 2019, <https://it.wikipedia.org/wiki/CIP6>
- [7] “E-Distribuzione”, *Wikipedia - The Free Encyclopedia*, 8 Novembre 2020, <https://it.wikipedia.org/wiki/E-distribuzione>
- [8] “C’è una rete che unisce le nostre energie”, *Brochure E-Distribuzione*, <https://www.e-distribuzione.it/Azienda/chi-siamo.html>
- [9] “Come funziona il mercato dei servizi di dispacciamento”, *NEXT*, <https://www.centrali-next.it/hub-della-conoscenza/mercato-dei-servizi-di-dispacciamento-msd/>
- [10] A. Escribano, J. I. Peña, P. Villaplana, “Modeling electricity price: international evidence”, *Working Paper 02-27, Economic Series 08*, pp. 1-32, Departamento de Economía Universidad Carlos III de Madrid, 2002

- [11] C. R. Knittel, M. R. Roberts, “An empirical examination of restructured electricity prices”, *Energy Economics* 27, pp. 791– 817, 2005
- [12] L. Jiang, G. Hu, “A Review on Short-Term Electricity Price Forecasting Techniques for Energy Markets”, Novembre 2018
- [13] “Teoria dei giochi”, *Wikipedia - The Free Encyclopedia*, 13 Settembre 2020, https://it.wikipedia.org/wiki/Teoria_dei_giochi
- [14] J. Wu, “AI, Machine Learning, Deep Learning Explained Simply Supervised ML, Unsupervised ML, Reinforcement Learning”, *Towards Data Science*, 1 Luglio 2019
- [15] J. Brownlee, “How to Fix the Vanishing Gradients Problem Using the ReLU”, *Machine Learning Mastery*, 25 Agosto 2020, <https://machinelearningmastery.com/how-to-fix-vanishing-gradients-using-the-rectified-linear-activation-function/>
- [16] C. Olah, “Understanding LSTM Networks”, *Colah’s Blog*, 27 Agosto 2015, <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- [17] S. Bouktif, A. Fiaz, A. Ouni, M. A. Serhani, “Optimal Deep Learning LSTM Model for Electric Load Forecasting using Feature Selection and Genetic Algorithm: Comparison with Machine Learning Approaches”, *Energies* 2018, 22 Giugno 2018, <https://www.mdpi.com/1996-1073/11/7/1636/htm>
- [18] A. Grossi, “Keras, cos’è, a cosa serve, come si usa nel deep learning”, *AI4business*, 16 Luglio 2020, <https://www.ai4business.it/intelligenza-artificiale/keras-cose-a-cosa-serve-come-si-usa-nel-deep-learning/>
- [19] “Front-End e Back-End”, *Wikipedia - The Free Encyclopedia*, 30 Settembre 2020, https://it.wikipedia.org/wiki/Front-end_e_back-end
- [20] J. Brownlee, “Prepare Data for Machine Learning in Python with Pandas”, *Machine Learning Mastery*, 23 Giugno 2014, <https://machinelearningmastery.com/prepare-data-for-machine-learning-in-python-with-pandas/>

- [21] Kaggle, “Pandas, Numpy, Python Cheatsheet”,
<https://www.kaggle.com/lavanyashukla01/pandas-numpy-python-cheatsheet>
- [22] “Python, introduction to Matplotlib”, *Geeks for Geeks*, 14 Maggio 2018,
<https://www.geeksforgeeks.org/python-introduction-matplotlib/>
- [23] B. Romero, “Cleaning Data”, *KDnuggets*, <https://www.kdnuggets.com/2016/06/doing-data-science-kaggle-walkthrough-data-cleaning.html>
- [24] Kaggle, “Handling Missing values”, <https://www.kaggle.com/dansbecker/handling-missing-values>
- [25] J. Brownlee, “How to Scale Data for Long Short-Term Memory networks in Python”, *Machine Learning Mastery*, 7 Luglio 2017, <https://machinelearningmastery.com/how-to-scale-data-for-long-short-term-memory-networks-in-python/>
- [26] J. Brownlee, “How to Scale Data with Outliers for Machine Learning”, *Machine Learning Mastery*, 27 Maggio 2020, <https://machinelearningmastery.com/robust-scaler-transforms-for-machine-learning/>
- [27] J. Brownlee, “What is the Difference Between Test and Validation Dataset”, *Machine Learning Mastery*, 14 Luglio 2017, <https://machinelearningmastery.com/difference-test-validation-datasets/>
- [28] J. Brownlee, “Dropout Regularization in Deep Learning Models with Keras”, *Machine Learning Mastery*, 20 Giugno 2016, <https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/>
- [29] D. Gupta, “Fundamentals of Deep Learning – Activation Functions and When to Use Them”, *Analytics Vidhya*, 30 Gennaio 2020,
<https://www.analyticsvidhya.com/blog/2020/01/fundamentals-deep-learning-activation-functions-when-to-use-them/>

- [30] “CPU”, *Wikipedia - The Free Encyclopedia*, 27 Ottobre 2020, <https://it.wikipedia.org/wiki/CPU>
- [31] “GPU”, *Wikipedia - The Free Encyclopedia*, 5 Maggio 2020, https://it.wikipedia.org/wiki/Graphics_processing_unit
- [32] J. Brownlee, “Gentle introduction to the Adam Optimization Algorithm for Deep Learning”, *Machine Learning Mastery*, 3 Luglio 2017, <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>
- [33] J. Brownlee, “How to Choose Loss Functions When Training Deep Learning Neural Networks”, *Machine Learning Mastery*, 30 Gennaio 2019, <https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/>
- [34] J. Brownlee, “The 5 Step Life-Cycle for long Short-Term Memory Models in Keras”, *Machine Learning Mastery*, 7 Giugno 2017, <https://machinelearningmastery.com/5-step-life-cycle-long-short-term-memory-models-keras/>
- [35] J. Brownlee, “Difference Between a Batch and an Epoch in a Neural Network”, *Machine Learning Mastery*, 20 Luglio 2018, <https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/>
- [36] L. Govoni, “L’Overfitting e l’Underfitting nel machine learning”, <https://lorenzogovoni.com/overfitting-e-underfitting-machine-learning/>
- [37] J. Brownlee, “Overfitting and Underfitting With Machine Learning Algorithms”, *Machine Learning Mastery*, 21 Marzo 2016, <https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/>