



POLITECNICO DI TORINO

DIPARTIMENTO DI INGEGNERIA ENERGETICA

Corso di Laurea Magistrale in Ingegneria Energetica e Nucleare

Il Machine Learning nel Monitoraggio energetico

Relatori:

Prof. Marco Badami

Prof. Armando Portoraro

Candidato:

Riccardo Quaglia

ANNO ACCADEMICO 2018-2019

Indice

INDICE	2
INDICE DELLE FIGURE.....	4
INDICE DELLE TABELLE	7
INTRODUZIONE	9
1 I SISTEMI DI MONITORAGGIO ENERGETICO	10
1.1 INTRODUZIONE AL MONITORAGGIO ENERGETICO	10
1.2 L'ARCHITETTURA TRADIZIONALE DEI SISTEMI DI MONITORAGGIO ENERGETICO	11
1.2.1 <i>Livello di campo</i>	12
1.2.2 <i>Livello di trasmissione</i>	12
1.2.3 <i>Livello di analisi e gestione dei dati</i>	14
1.3 IL FUTURO DEI SISTEMI DI MONITORAGGIO ENERGETICO	14
1.3.1 <i>Il ruolo del monitoraggio energetico nella quarta rivoluzione industriale</i>	14
1.3.2 <i>I nuovi strumenti software per la gestione</i>	20
1.3.3 <i>L'implementazione dei nuovi strumenti negli impianti esistenti</i>	21
2 IL MACHINE LEARNING NEL MONITORAGGIO ENERGETICO.....	22
2.1 IL MACHINE LEARNING E LA DATA ANALYSIS	22
2.2 LE RETI NEURALI ARTIFICIALI.....	24
2.2.1 <i>Funzioni di attivazione</i>	25
2.2.2 <i>Apprendimento</i>	27
2.2.3 <i>Attività di apprendimento</i>	29
2.2.4 <i>Generalizzazione</i>	30
2.2.5 <i>La funzione obiettivo e l'ottimizzatore</i>	32
2.2.6 <i>Prevenzione di Overfitting e Underfitting</i>	32
2.3 CLASSIFICAZIONE DELLE RETI NEURALI	36
2.3.1 <i>Feedforward</i>	36
2.3.2 <i>Convolutionali</i>	37
2.3.3 <i>Recurrent-Long Short Term Memory</i>	38
2.3.4 <i>Reti modulari</i>	39
2.4 MODELLIZZAZIONE DI UNA RETE NEURALE	39
3 ESEMPI DI APPLICAZIONI DEL MACHINE LEARNING IN AMBITO ENERGETICO.....	43
3.1 TIPOLOGIA DEGLI IMPIANTI ANALIZZATI	43
3.2 IMPOSTAZIONE DELLO STUDIO.....	44
3.2.1 <i>Modelli e strumenti utilizzati</i>	44
3.2.2 <i>Indici utilizzati</i>	45
3.3 INDUSTRIA TESSILE	47
3.3.1 <i>Gruppo di cogenerazione</i>	47
3.3.2 <i>Apparecchiature di monitoraggio</i>	50
3.3.3 <i>Preparazione e formattazione dei dati</i>	51
3.3.4 <i>Rete FF per la previsione oraria</i>	51
3.3.5 <i>Rete FF per la previsione giornaliera</i>	56
3.3.6 <i>Rete FF per la previsione settimanale</i>	60
3.3.7 <i>Rete FF per la previsione settimanale con modello alternativo</i>	64
3.3.8 <i>Rete LSTM per la previsione oraria</i>	66
3.3.9 <i>Rete LSTM per la previsione giornaliera</i>	70
3.3.10 <i>Confronto fra i modelli</i>	74
3.3.11 <i>Rete LSTM per la previsione settimanale</i>	75
3.4 INDUSTRIA ALIMENTARE	79
3.4.1 <i>Gruppo di cogenerazione e impianto fotovoltaico</i>	79
3.4.2 <i>Apparecchiature per il monitoraggio</i>	80
3.4.3 <i>Preparazione e formattazione dei dati</i>	81
3.4.4 <i>Rete FF per la previsione oraria</i>	82

3.4.5	<i>Rete FF per la previsione giornaliera</i>	85
3.4.6	<i>Rete FF per la previsione settimanale</i>	89
3.4.7	<i>Rete LSTM per la previsione oraria</i>	93
3.4.8	<i>Rete LSTM per la previsione giornaliera</i>	96
3.4.9	<i>Rete LSTM per la previsione settimanale</i>	100
3.4.10	<i>Confronto fra i modelli</i>	103
3.4.11	<i>Rete LSTM per previsione settimanale con input esogeni</i>	104
4	CONCLUSIONI.....	113
	APPENDICE.....	118
	RETE FF PER LA PREVISIONE ORARIA	118
	RETE FF PER LA PREVISIONE GIORNALIERA	119
	CALCOLO DEGLI INDICI.....	121
	BIBLIOGRAFIA	124

Indice delle figure

Figura 1: struttura di un sistema per la gestione energetica	11
Figura 2: apparecchiatura PLC.....	12
Figura 3: topologia delle reti di comunicazione	13
Figura 4: dispositivi connessi ad internet [2].....	15
Figura 5: andamento del prezzo medio dei sensori [5].....	16
Figura 6: andamento prezzi dello storage di dati [5].....	16
Figura 7: riferimenti espliciti alla sicurezza nell'IoT.....	17
Figura 8: funzionamento di una smart factory [7].....	19
Figura 9: architettura IoT applicata al monitoraggio energetico	19
Figura 10: le 5V dei big data	20
Figura 11: Architettura di un sistema basato sul machine learning [5]	21
Figura 12: programma tradizionale (sinistra) e programma di machine learning (destra).....	23
Figura 13: struttura base di una rete neurale artificiale	24
Figura 14: schematizzazione matematica di un percettrone	25
Figura 15: funzione gradino di Heaviside	26
Figura 16: funzione lineare a tratti.....	26
Figura 17: funzione di Sigmoid al variare di a.....	27
Figura 18: apprendimento con correzione dell'errore	28
Figura 19: discesa del gradiente	28
Figura 20: classificazione dei pattern	30
Figura 21: confronto fra una buona generalizzazione (a) e una cattiva generalizzazione (b) dovuta ad overfitting.....	31
Figura 22: processo di addestramento di una rete neurale	32
Figura 23: modello in underfitting.....	33
Figura 24: modello con un addestramento efficace.....	33
Figura 25: modello in overfitting	34
Figura 26: metodo di "early stopping"	35
Figura 27: rete feedforward con uno strato nascosto.....	37
Figura 28: convolutional network.....	37
Figura 29: struttura interna	38
Figura 30: rete ricorrente	39
Figura 31: rete modulare	39
Figura 32:andamento tipico della funzione Loss.....	42
Figura 33: produzione tramite cogenerazione e produzione separata	43
Figura 34: schema dei modelli.....	45
Figura 35: gruppo di cogenerazione dall'esterno.....	47
Figura 36: gruppo di cogenerazione, vista posteriore.....	48
Figura 37: schema impianto di cogenerazione.....	49
Figura 38: schema semplificato dell'impianto di cogenerazione	50
Figura 39: strumento di misura dell'energia elettrica prodotta.....	50
Figura 40: struttura originale dei dati acquisiti	51
Figura 41: funzionamento della rete FF per le previsioni orarie	52
Figura 42: andamento dell'addestramento.....	54

Figura 43: distribuzione errore	55
Figura 44: confidenza durante la fase di training	55
Figura 45: comparazione potenze previste con le potenze reali, rete FF oraria	56
Figura 46: funzionamento della rete FF per le previsioni giornaliere	56
Figura 47: andamento dell'errore durante la fase di training	59
Figura 49: confidenza durante la fase di training	59
Figura 48: distribuzione errore	59
Figura 50: comparazione potenze previste con le potenze reali, rete FF giornaliera	60
Figura 51: funzionamento della rete FF per le previsioni settimanale	60
Figura 52: andamento dell'errore durante la fase di training	63
Figura 54: confidenza durante la fase di training	63
Figura 53: distribuzione errore	63
Figura 55: comparazione potenze previste con le potenze reali, rete FF settimanale	64
Figura 56: previsione della rete alternativa	65
Figura 57: funzionamento della rete LSTM per la previsione oraria	66
Figura 58: andamento dell'addestramento	68
Figura 59: confidenza durante la fase di training	69
Figura 60: distribuzione errore	69
Figura 61: comparazione potenze previste con le potenze reali, rete LSTM oraria	69
Figura 62: funzionamento della rete LSTM per le previsioni giornaliere	70
Figura 63: andamento dell'addestramento	72
Figura 64: confidenza durante la fase di training	73
Figura 65: comparazione potenze previste con le potenze reali rete LSTM giornaliera	73
Figura 66: funzionamento della rete LSTM per le previsioni settimanali	75
Figura 67: andamento dell'addestramento	77
Figura 68: confidenza durante la fase di training	78
Figura 69: distribuzione dell'errore	78
Figura 70: comparazione potenze previste con le potenze reali rete LSTM settimanale	78
Figura 71: gruppo di cogenerazione	79
Figura 72: strumento PLC di misura dell'energia elettrica prodotta	80
Figura 73: schema di funzionamento dell'impianto di cogenerazione	81
Figura 74: struttura originale dei dati acquisiti	81
Figura 75: funzionamento della rete FF per le previsioni orarie	82
Figura 76: andamento dell'addestramento	84
Figura 77: confidenza durante la fase di training	84
Figura 78: distribuzione dell'errore	85
Figura 79: comparazione potenze previste con le potenze reali, rete FF oraria	85
Figura 80: funzionamento della rete FF per le previsioni giornaliere	86
Figura 82: andamento dell'addestramento	88
Figura 81: distribuzione dell'errore	88
Figura 83: confidenza durante la fase di training	88
Figura 84: comparazione potenze previste con le potenze reali, rete FF giornaliera	89
Figura 85: funzionamento della rete FF per le previsioni settimanale	89
Figura 86: andamento dell'addestramento	91
Figura 87: confidenza durante la fase di training	92
Figura 88: distribuzione dell'errore	92

Figura 89: comparazione potenze previste con le potenze reali rete FF settimanale	92
Figura 90: rete LSTM per la previsione oraria	93
Figura 91: distribuzione errore	95
Figura 92: andamento dell'addestramento	95
Figura 93: confidenza durante la fase di training	95
Figura 94: comparazione potenze previste con le potenze reali, rete LSTM oraria	96
Figura 95: funzionamento della rete LSTM per le previsioni giornaliere	96
Figura 96: andamento dell'addestramento	98
Figura 97: distribuzione errore	99
Figura 98: confidenza durante la fase di training	99
Figura 99: comparazione potenze previste con le potenze reali, rete LSTM giornaliera	99
Figura 100: funzionamento della rete LSTM per le previsioni settimanali	100
Figura 101: andamento dell'addestramento	102
Figura 102: confidenza durante la fase di training	102
Figura 103: distribuzione dell'errore	102
Figura 104: comparazione potenze previste con le potenze reali rete LSTM settimanale	103
Figura 105: variabili atmosferiche in relazione alla produzione del cogeneratore	104
Figura 106: diagramma del modello con input esogeni	105
Figura 108: fase di training	106
Figura 107: distribuzione dell'errore	106
Figura 109: risultati previsione produzione FV	107
Figura 110: andamento della fase di training	108
Figura 111: distribuzione dell'errore	109
Figura 112: comparazione potenze previste con le potenze reali, rete LSTM giornaliera con input esogeni	109
Figura 113: comparazione potenze previste con potenze reali, rete LSTM settimanale con input esogeni	109
Figura 114: comparazione potenze previste con le potenze reali, rete FF giornaliera con input esogeni	110
Figura 115: comparazione potenze previste con le potenze reali, rete FF settimanale con input esogeni	110
Figura 116: previsione produzione fotovoltaica basata su previsioni metereologiche	111
Figura 117: previsione della produzione del cogeneratore basata su previsioni metereologiche	111
Figura 118: previsione della produzione fotovoltaica basata sulle previsioni metereologiche	112
Figura 119: previsione della produzione del cogeneratore basata su previsioni metereologiche	112
Figura 120: previsione di una settimana con raccolta dati parzialmente mancante	113
Figura 121: previsione della produzione del fotovoltaico in base all'irraggiamento solare	115
Figura 122: modello previsionale giornaliero con impianto fotovoltaico	115

Indice delle tabelle

Tabella 1: tasso di risparmio energetico	10
Tabella 2: costi strumentazione hardware	11
Tabella 3: vulnerabilità in ambito IoT [6]	17
Tabella 4: Prestazioni delle reti	36
Tabella 5: dati di targa dell'impianto di cogenerazione	49
Tabella 6: indici mensili relativi al primo impianto di cogenerazione	49
Tabella 7: capisaldi rete FF oraria.....	52
Tabella 8: prestazioni dell'addestramento rete FF oraria	54
Tabella 9: capisaldi rete FF giornaliera	57
Tabella 10: prestazioni dell'addestramento rete FF giornaliera	58
Tabella 11: capisaldi rete FF giornaliera	61
Tabella 12: prestazioni dell'addestramento rete FF settimanale.....	62
Tabella 13: capisaldi rete LSTM oraria.....	66
Tabella 14: prestazioni dell'addestramento rete LSTM per previsioni orarie.....	68
Tabella 15: capisaldi rete LSTM giornaliera.....	71
Tabella 16: prestazioni dell'addestramento rete LSTM giornaliera	72
Tabella 19: valori medi degli indici di prestazione	74
Tabella 17: capisaldi rete LSTM giornaliera.....	76
Tabella 18: prestazioni dell'addestramento rete LSTM settimanale	76
Tabella 20: dati di targa	79
Tabella 21: indici relativi al secondo impianto di cogenerazione.....	79
Tabella 22: capisaldi rete FF oraria.....	83
Tabella 23: prestazioni dell'addestramento rete FF oraria	83
Tabella 24: capisaldi rete FF giornaliera	86
Tabella 25: prestazioni dell'addestramento rete FF giornaliera	87
Tabella 26: capisaldi rete FF giornaliera	90
Tabella 27: prestazioni dell'addestramento rete LSTM settimanale	91
Tabella 28: capisaldi rete LSTM oraria.....	93
Tabella 29: prestazioni dell'addestramento rete LSTM per previsioni orarie.....	94
Tabella 30: capisaldi rete LSTM giornaliera.....	97
Tabella 31: prestazioni dell'addestramento rete LSTM giornaliera	98
Tabella 32: capisaldi rete LSTM settimanale	100
Tabella 33: prestazioni dell'addestramento rete LSTM settimanale	101
Tabella 34: valori medi degli indici di prestazione	103
Tabella 35: capisaldi rete FF per la previsione del fotovoltaico	106
Tabella 36: capisaldi della rete LSTM con input esogeni per la previsione del cogeneratore	107
Tabella 37: prestazioni rete LSTM con input esogeni.....	108
Tabella 38: prestazioni reti con input esogeni	110
Tabella 39: riassunto prestazioni reti per il primo stabilimento	114
Tabella 40: riassunto prestazioni reti per il secondo stabilimento	116

Introduzione

Nel presente lavoro di tesi si sono sviluppati ed analizzati diversi modelli di previsione dei consumi e della produzione di energia basati sul Machine Learning, sfruttando i dati provenienti da alcuni stabilimenti produttivi; tale analisi ha permesso di individuare punti di forza e criticità dei processi in esame. Inoltre, ha reso possibile una previsione dei consumi strategicamente utile per amministrare al meglio le risorse. Infatti, una corretta regolazione dell'utilizzo dell'energia è fondamentale nel settore industriale per conseguire un'efficace gestione delle risorse energetiche ed economiche.

Negli ultimi anni il grande aumento di disponibilità di informazioni a disposizione, da un punto di vista sia quantitativo sia qualitativo, e lo sviluppo di algoritmi computazionali nel campo dell'intelligenza artificiale hanno consentito lo sviluppo di nuovi modelli di previsione dei consumi. Alcuni di questi modelli saranno esposti nei prossimi capitoli.

La prima parte dello studio descrive gli attuali sistemi di monitoraggio energetico e gli scenari di sviluppo per questo settore nei prossimi anni, concentrandosi sull'integrazione al suo interno di strumenti di Machine Learning. Particolare attenzione è stata dedicata allo studio delle varie tipologie di algoritmi utilizzabili, andando ad individuare quelli in grado di fornire le migliori prestazioni relativamente al problema studiato.

Nella seconda parte del lavoro si sono sviluppati i modelli di previsione dei consumi e della produzione tramite l'utilizzo delle librerie open-source Tensorflow per ambiente Python sviluppate da Google, i quali sono stati valutati basandosi sui dati di funzionamento reale di due stabilimenti produttivi, il primo alimentato solo da un impianto di cogenerazione e il secondo da un impianto di cogenerazione affiancato da un impianto fotovoltaico per la produzione di energia elettrica.

Le analisi effettuate sono di interesse per valutare l'efficacia concreta dei modelli previsionali basati sul Machine Learning in questo campo di applicazione e se essi si possano affermare nel prossimo futuro in sostituzione delle attuali tecnologie.

1 I Sistemi di monitoraggio energetico

1.1 Introduzione al monitoraggio energetico

Negli ultimi decenni, a causa del progredire delle economie dei paesi in via di sviluppo, si è assistito ad un elevato aumento del fabbisogno energetico mondiale. La maggior parte dell'energia è tuttora prodotta a partire da fonti fossili, il cui prezzo e disponibilità sono molto suscettibili delle condizioni geo-politiche mondiali. Per questo motivo, unito alla sempre maggiore sensibilità sulle tematiche ambientali ha reso sempre più centrale la questione del risparmio energetico, sia in ambito civile sia in ambito industriale.

Il settore industriale occupa circa il 29% dell'intero fabbisogno mondiale di energia (in Italia circa il 22%) e molto spesso il consumo di energia è una voce molto importante nel bilancio di un'azienda. Ne consegue che lo sviluppo di politiche volte al risparmio e alla razionalizzazione del consumo energetico costituisca una importantissima opportunità di risparmio economico.

Questo obiettivo si persegue per mezzo di una opportuna strategia di Energy Intelligence, cioè *la creazione di know-how grazie alla rielaborazione delle informazioni sui consumi elettrici e termici di un'utenza energetica, al fine di realizzare un vantaggio competitivo per l'utenza stessa, grazie alla riduzione dei costi energetici* [1]. Lo sviluppo di questa capacità avviene per mezzo di dispositivi hardware per l'acquisizione delle informazioni necessarie e dispositivi software per l'analisi.

Lo scopo è quello di creare un sistema di supervisione in grado di gestire in maniera quasi del tutto autonoma la produzione di uno stabilimento. Un sistema di supervisione è a sua volta costituito da:

- Un **sistema di monitoraggio** che consente di conoscere lo stato delle utenze energetiche che compongono l'impianto, attraverso strumenti hardware che misurano le grandezze di interesse e applicativi software che visualizzano l'andamento dei consumi. Solo conoscendo nel dettaglio tutti i processi coinvolti e i relativi consumi è possibile pianificare un intervento di efficientamento e risparmio realmente efficace;
- Un **sistema di controllo** che permette analizzare via software le grandezze monitorate, confrontarle con baseline e valori di riferimento ed attuare, se necessario, le opportune misure correttive pe mezzo di attuatori fisici. Oltre al miglioramento delle prestazioni attraverso soluzioni tecniche, termodinamiche o dal punto di vista dei materiali impiegati, una importantissima opportunità di razionalizzazione dell'uso dell'energia è costituita da un'efficace regolazione delle apparecchiature. Molto spesso, infatti, gli impianti si trovano a dover lavorare in condizioni non ottimali a causa di errato dimensionamento delle macchine e di sistemi di regolazione approssimativi o mal tarati.

I sistemi di Energy Intelligence sono maggiormente diffusi per le applicazioni industriali rispetto a quelle civili e per la gestione di edifici a causa del maggiore impatto economico derivante dalla loro implementazione. Andando a confrontare il tasso di risparmio energetico che è possibile conseguire (vedi Tabella 1), definito come la riduzione percentuale del costo complessivo della bolletta energetica, con il costo delle apparecchiature necessarie, si nota come esso sia molto più remunerativo nel settore industriale [1].

Tabella 1: tasso di risparmio energetico

Soluzione tecnologica		Processo produttivo	Building	
			Residenziale	Non Residenziale
Monitoraggio	Semplice	3-7%	2-6%	2-7%
	Con analisi	4-9%	3-8%	3-8%
Controllo		10-15%	8-14%	7-13%
Supervisione		15-20%	12-19%	13-18%

Il costo della strumentazione software dipende direttamente dal numero di apparecchiature hardware che devono essere gestite e dalle loro specificità in base all'*affidabilità* (frequenza di errori e malfunzionamenti durante l'elaborazione dei dati), all'*efficienza* (sia dal punto di vista energetico sia dal punto di vista delle capacità software), alla *facilità di utilizzo*, alla *manutenibilità* e alla possibilità di implementare future modifiche. Il costo totale di un intervento è fortemente legato alla dimensione dello stabilimento o dell'edificio analizzato; ipotizzando un costo unitario della strumentazione come riportato nella Tabella 2 e una quantità di apparecchi compresi fra 10 e 15 unità, si ottiene un costo che oscilla fra i 20.000 e 70.000€ per le applicazioni industriali e fra i 15.000 e 55.000€ nel settore edilizio. [1]

Tabella 2: costi strumentazione hardware

Soluzione tecnologica	Processo produttivo	Building
Monitoraggio	350-1.200€	100-750€
Controllo	700-1.600€	600-1.350€
Supervisione	950-2.200€	800-1.500€

1.2 L'architettura tradizionale dei sistemi di monitoraggio energetico

I sistemi di monitoraggio costituiscono lo strumento principale per conseguire e garantire gli obiettivi preposti di efficientamento e corretta gestione dell'energia.

Per mezzo del monitoraggio è possibile conoscere nel dettaglio le grandezze tipiche del funzionamento di un impianto (temperature, portate, pressioni, voltaggi, ecc) e i conseguenti indici derivati (come l'efficienza) permettendo una gestione e regolazione ottimale e anche la capacità di individuare tempestivamente eventuali malfunzionamenti e situazioni critiche.

Un sistema per la gestione dell'energia è sviluppato su tre livelli:

- Un **livello di campo**: è costituito dalle apparecchiature di misura con interfaccia per la comunicazione installati fisicamente nello stabilimento, le quali interagiscono in maniera diretta con esso e consentono di interfacciarsi con il livello di gestione, trasmettendo le informazioni relative alle grandezze misurate ai livelli successivi;
- Un **livello di collegamento**: consente la comunicazione reciproca fra gli altri livelli. A seconda delle esigenze può essere costituito da reti wireless (sia Wi-Fi sia su rete telefonica GPRS) o da sistemi cablati;
- Un **livello di analisi e gestione** dei dati e l'adozione di strategie volte all'efficienza attraverso software dedicati.



Figura 1: struttura di un sistema per la gestione energetica

1.2.1 Livello di campo

Il primo livello del monitoraggio è consistere nella acquisizione delle grandezze di interesse, mediante delle apparecchiature in grado di monitorare i principali parametri di funzionamento di un impianto (temperature, portate, tensioni, pressione, ecc...). La necessità di monitorare le grandezze dell'impianto è dovuta anche all'automatizzazione dei processi; un tempo pompe e valvole erano attivate manualmente da operatori umani gradualmente sostituiti da attuatori automatizzati che necessitano di una complessa rete di sensori connessi a sistemi di controllo PLC o RTU.

Questo da un lato aumenta i costi connessi alla gestione e progettazione dell'impianto e le conoscenze richieste dai tecnici, ma dall'altro lato garantisce una capacità di reazione in caso di necessità molto maggiore in caso di malfunzionamenti od incidenti.

Molta strumentazione deve poter operare esposta agli agenti atmosferici se posizionati all'esterno o elevate temperature se posizionati all'interno degli stabilimenti produttivi; nel caso in cui la strumentazione non sia stata progettata a tale scopo è necessario proteggerla con dei cabinati o altre costruzioni protettive, andando a costituire un'ulteriore voce di spesa.

Molto spesso la strumentazione deve anche garantire una compatibilità elettromagnetica per non generare o subire interferenze con le altre apparecchiature che compongono il sistema.

Gli strumenti maggiormente impiegati in questo ambito sono:

- **PLC** (Programmable Logic Controller) sono "controllori a logica programmabile" che adoperano una memoria integrata per memorizzare istruzioni ed informazioni che vengono utilizzate per svolgere specifiche operazioni, quali codifiche, decodifiche, temporizzazioni, conteggi e operazioni logiche volte a controllare sistemi e processi;
- **DCS** (Distributed Control System) è un "sistema di controllo distribuito" costituito da più sottosistemi, compresi l'acquisizione ed elaborazione dei dati, capace di trasmettere e ricevere autonomamente informazioni con il campo con un'architettura distribuita;
- **RTU** (Remote Terminal Unit) sono "unità a terminale remoto" contenenti un microprocessore che permette di interfacciare dispositivi fisici a sistemi DCS o SCADA per mezzo della trasmissione dei dati acquisiti grazie alla strumentazione sul campo.



Figura 2: apparecchiatura PLC

I Sistemi PLC e RTU sono nati come apparecchiature distinte, ma la convergenza tecnologica nel loro sviluppo le ha portate oggi ad essere molto spesso sovrapponibili.

1.2.2 Livello di trasmissione

Per trasmettere le informazioni raccolte dai sistemi RTU/PLC, solitamente situati sul campo o lungo le tubazioni, all'host informatico dove esse vengono elaborate è necessaria una rete di comunicazione. Poiché

la strumentazione è distribuita su di una superficie spesso molto estesa la realizzazione di una rete efficiente, che svolge la funzione di collante fra le varie componenti, è essenziale.

Fino a una decina di anni fa i sistemi di trasmissione dati erano in gran parte cablati e connessi fra di loro attraverso dei modem situati a distanze periodiche, al giorno d'oggi invece è possibile realizzare gran parte delle reti attraverso soluzioni wireless (ad onde radio o comunicazioni satellitari) riducendo notevolmente i costi.

Una rete è composta da nodi e rami di comunicazione: un nodo (costituito da un dispositivo fisico in grado di comunicare con altri dispositivi) è un punto in cui i dati vengono trasmessi e ricevuti, il ramo invece è lo strumento con cui vengono trasmesse le informazioni (è il cavo il segnale radio).

Vi sono varie tipologie di reti in base alla configurazione fra nodi (N) e rami (R):

- Le **Reti a stella** hanno un nodo centrale (hub) che è in grado di comunicare con tutti gli altri nodi (host). Il nodo centrale controlla e gestisce tutte le funzionalità e le attività della rete, in questo modo è possibile ridurre l'impatto causato da un malfunzionamento in un ramo poiché ogni host è collegato in maniera autonoma all'hub, e di conseguenza si avrà l'isolamento di un singolo host senza inficiare gli altri. Tuttavia, nel caso di malfunzionamento dell'hub centrale l'intera rete smette di funzionare. In queste reti la relazione fra i nodi e i rami è $R=N-1$. Questa tipologia di rete è molto usata nelle reti cablate a doppino, coassiali e con fibra ottica;
- Le **Reti ad anello** sono delle reti lineari chiuse in cui ogni nodo è collegato agli altri due nodi adiacenti. Queste reti non hanno bisogno di un nodo centrale di gestione e permettono una notevole capacità di tollerare guasti al sistema di trasmissione poiché le informazioni possono viaggiare in entrambi i sensi di percorrenza e quindi possono raggiungere qualunque nodo anche nel caso un ramo non funzionante. In queste reti la relazione fra i nodi e i rami è $R=N$.
- Le **Reti a Bus** ogni nodo è collegato a tutti gli altri attraverso un unico ramo condiviso (bus). Queste reti sono molto apprezzate per la loro scalabilità (un singolo nodo può essere aggiunto o rimosso senza inficiare il resto della rete) e quindi molto usate nelle reti di trasmissione dati ethernet.

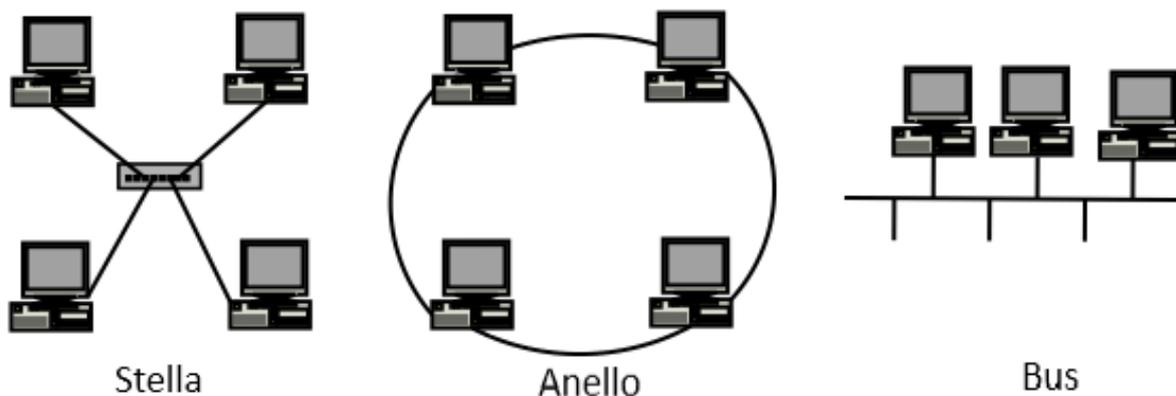


Figura 3: topologia delle reti di comunicazione

Dopo aver realizzato il collegamento fisico che permette la trasmissione dei dati, sia esso cablato o wireless, occorre definire il protocollo di comunicazione utilizzato dalla rete.

Un protocollo di comunicazione è un insieme di regole formali, definite a priori, che devono essere rispettate dalle apparecchiature che compongono la rete al fine di garantire la corretta funzionalità della rete stessa.

Esistono numerose tipologie differenti di protocolli sviluppati che differiscono fra loro in base alle diverse apparecchiature e utilizzi, alcuni sono open source ed altri proprietari, i maggiormente utilizzati sono:

- **MODBUS RTU**: è un protocollo disponibile liberamente senza royalties ed utilizza una comunicazione del tipo Master-Slave consentendo fino a 32 indirizzi sul collegamento (bus) in ingresso e uscita

- (trasmissione e ricezione). La lunghezza massima del collegamento dipende dalla velocità di trasmissione dei dati, arrivando fino ad oltre un kilometro. Aumentando la lunghezza del bus diminuisce la velocità di connessione. Utilizza coppie di cavi twistati e schermati adatti ad apparecchiature elettriche ed elettroniche conformi allo standard EIA RS485, con una impedenza tipica di circa 120 Ω. Viene spesso utilizzato per collegare un'unità remota RTU/PLC ad un computer;
- **M-BUS**: è un protocollo standard europeo definito nella normativa EN 12757-2 "*physical and link layer*" e EN 13757-3 "*application layer*", appositamente studiato per la lettura da remoto di apparecchiature per la contabilizzazione dei consumi (elettrici, idrici, termici, ecc), che utilizza una comunicazione del tipo Master-Slave con fino a 250 dispositivi collegati contemporaneamente. Utilizza coppie di cavi twistati e schermati adatti ad apparecchiature elettriche ed elettroniche conformi allo standard EN 1434-3, per le comunicazioni wireless è conforme alle specifiche contenute nella normativa EN 13757-3. Questo protocollo permette di rilevare e riconoscere automaticamente i dispositivi collegati alla rete;
 - **Profibus-DP**: è un protocollo utilizzato per la comunicazione cablata fra sensori e attuatori, alla velocità di connessione dipende dalla lunghezza del bus e permette la presenza fino a 126 nodi nella rete, oltre questo valore devono essere adoperati dei ripetitori;
 - **Common Industrial Protocol (CIP)**: è un protocollo industriale per l'automazione di applicazioni. Permette l'utilizzo di svariati network Ethernet ed Internet. È supportato e modificato da numerosi produttori in tutto il mondo che lo utilizzano come base per lo sviluppo di protocolli specifici;
 - **DeviceNet**: è utilizzato nei processi di automazione industriale per connettere dispositivi di controllo per lo scambio di dati. Si basa sul Common Industrial Protocol ed è possibile utilizzarlo su moltissimi dispositivi, tipicamente per lo scambio di informazioni.

1.2.3 Livello di analisi e gestione dei dati

Le informazioni raccolte sul campo e trasmesse dalla rete devono essere convogliate in un sistema di archiviazione per poter essere facilmente consultate ed utilizzate da software specifici per il monitoraggio e la gestione energetica. Al fine di rendere maggiormente fruibile la consultazione ed elaborazione dei dati è necessario provvedere ad una opportuna normalizzazione ed omogenizzazione del database.

A questo scopo vengono adoperati dei sistemi SCADA (Supervisory Control And Data Acquisition) che permettono il controllo di supervisione ed acquisizione dati.

Il compito di questi sistemi è acquisire dati da dispositivi remoti come pompe e valvole e permettere un controllo remoto da una piattaforma software di SCADA Host. Ciò consente di controllare localmente i processi accendendo o spegnendo i dispositivi, regolandoli e fornendo eventuali allarmi.

La diffusione di piattaforme di archiviazione cloud e dei sistemi e servizi basati su Internet ha introdotto un nuovo problema in questa filiera: la sicurezza informatica. Tradizionalmente i servizi SCADA erano sistemi isolati e circoscritti all'impianto di produzione (l'acquisizione e trasmissione dei dati avveniva tramite rete cablata ed essi erano archiviati su piccoli server locali) e non erano quindi progettati per essere protetti da eventuali attacchi informatici, oggi, invece, le piattaforme SCADA sono vulnerabili ad intrusioni esterne e necessitano di tool di protezione informatica.

1.3 Il futuro dei sistemi di monitoraggio energetico

1.3.1 Il ruolo del monitoraggio energetico nella quarta rivoluzione industriale

In questi anni il settore industriale sta subendo grandi cambiamenti grazie alla sempre maggiore diffusione di sistemi cibernetici, dell'Internet of Things e dell'Internet dei Servizi, andando a cambiare il paradigma dei settori produttivi; si parla, a tutti gli effetti, della quarta rivoluzione industriale.

Laddove la prima rivoluzione industriale riguardava la meccanizzazione della produzione, grazie ai primi motori a vapore, la seconda ha permesso la produzione di massa e la terza ha introdotto la digitalizzazione e l'automazione dei processi, la quarta rivoluzione è caratterizzata da quattro elementi costitutivi che vanno a modificare i processi produttivi:

- L'aumento esponenziale della quantità di dati raccolti;
- L'avvento di nuovi modelli di analisi per migliorare la produzione;

- Nuove sistemi HMI (Human Machine Interface) che rivoluzionano l'interazione uomo-machina (interfacce touch-screen, visori per la realtà aumentata);
- Miglioramenti nel trasferimento delle istruzioni digitali al mondo fisico (robotica avanzata, additive manufacturing).

Lo sviluppo di queste applicazioni viene generalmente sintetizzato con il termine Industria 4.0, che comprende tutte le tecnologie di automazione, scambio di dati e trasformazione che combinano i sistemi ciberfisici con l'IoT (Internet of Things) e i servizi connessi alla rete.

L'Internet of Things è una network technology nella quale ogni informazione, apparecchiatura e processo sono connessi ad Internet ed interconnessi fra loro allo scopo di generare, raccogliere, condividere e utilizzare informazioni. L'obiettivo dell'Internet of Things consiste nel migliorare il funzionamento delle apparecchiature mediante l'interconnessione fra di esse, agevolare l'automazione dei processi mediante la raccolta di una gran numero di informazioni consentendo una drastica diminuzione dell'intervento umano. Da punto di vista pratico, l'IoT consiste in strumenti software e hardware in grado di integrarsi efficacemente fra loro in una vera e propria rete gestibile da parte dell'utenza tramite un'interfaccia, la quale consente di visualizzare in maniera veloce e sintetica tutte le informazioni necessarie ad una gestione integrata nei modelli di business e nei processi sociali; questa architettura può anche essere sfruttata per garantire una attivazione automatica di uno strumento come risposta al verificarsi di un evento nel mondo fisico (ad esempio alzare od abbassare la temperatura di ambiente climatizzato in base alle condizioni climatiche misurate da sensori esterni) oppure attivare o disattivare uno specifico servizio senza l'intervento attivo di un essere umano. Per poter realizzare efficacemente una infrastruttura basata sull'IoT è necessario utilizzare un gran numero di sensori, tecnologie di comunicazione di rete, strumentazione dotata di processori logici, piattaforme di archiviazione dati e algoritmi per processare ed analizzare le informazioni e tecnologie per proteggere i dati sensibili lungo tutto il processo.

Molti importanti centri di studio e imprese del settore guardano con interesse a questo cambio di paradigma sia nel settore civile sia industriale:

- Cisco Systems prevede che ci saranno oltre 20 miliardi di dispositivi connessi entro il 2020 [2];
- La testata Business Insider stima che nel 2019 il giro di affari relativo all'IoT sarà di 1.700 miliardi di dollari in tutto il mondo [3];
- McKinsey Global Institute che nel 2025 l'impatto economico dell'IoT sarà di circa 11.000 miliardi di dollari nel mondo [4].

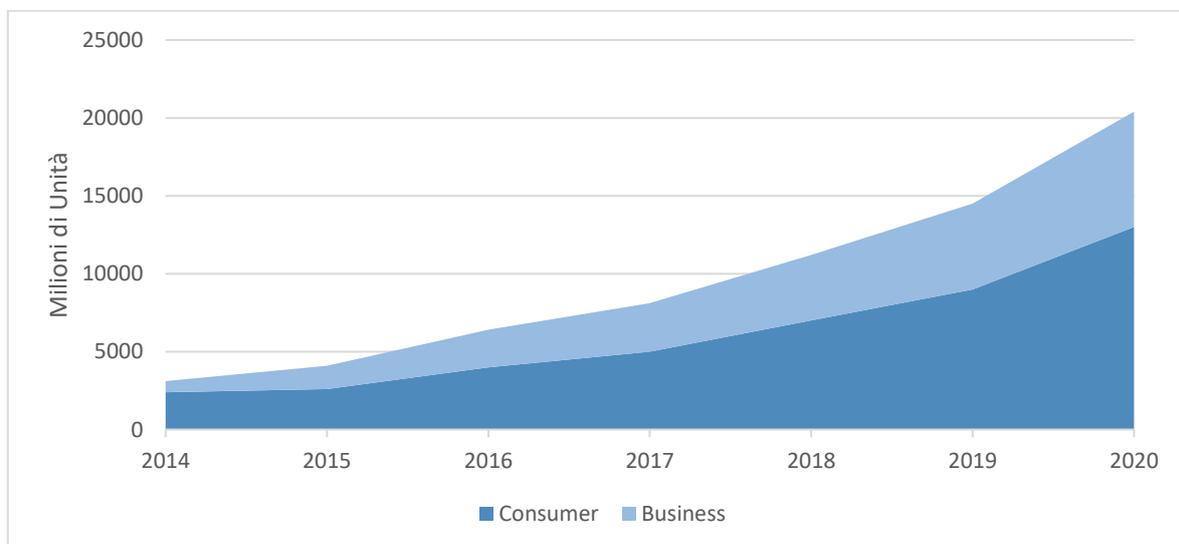


Figura 4: dispositivi connessi ad internet [2]

Questo fenomeno rappresenterà un vero proprio punto di svolta sia per i modelli di business aziendali, per la società e per la vita delle persone. La diffusione di questo nuovo approccio si è resa possibile negli ultimi anni grazie alla notevole diminuzione del costo legato alle apparecchiature di monitoraggio, il quale si è dimezzato nel corso degli ultimi dieci anni, passando da poco meno di 1\$ nel 2008 a circa 0.50\$ odierni e le previsioni indicano un ulteriore deprezzamento negli anni a venire (Vedi Figura 5). [5]

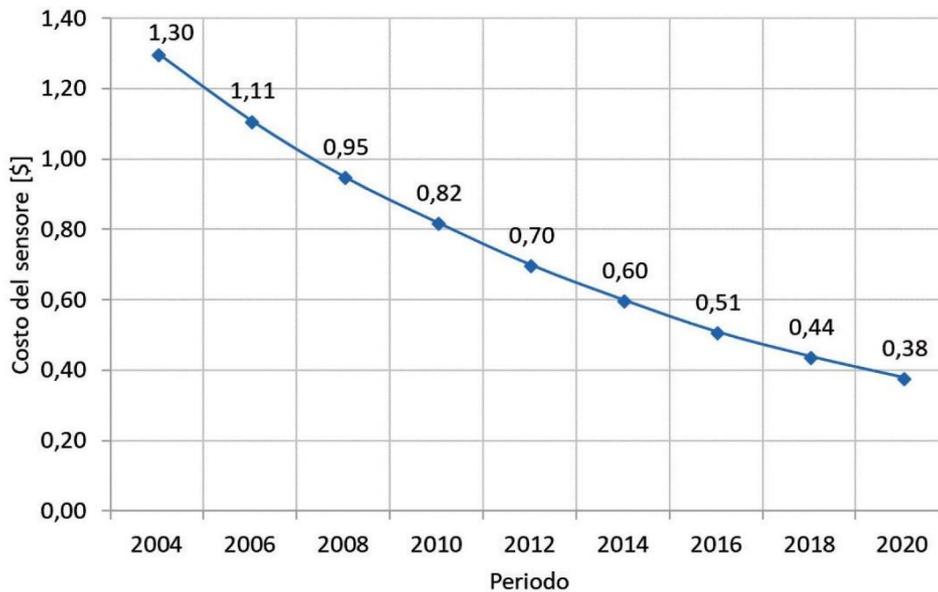


Figura 5: andamento del prezzo medio dei sensori [5]

Parallelamente al prezzo dei sensori si è anche drasticamente ridotto il costo dello storage di dati (vedi Figura 6) passato da circa 0.60\$/GB del 2008 a circa 0.10\$/GB attuali, in questo caso però si prevede che negli anni a seguire il prezzo dell'archiviazione di dati si manterrà costante.

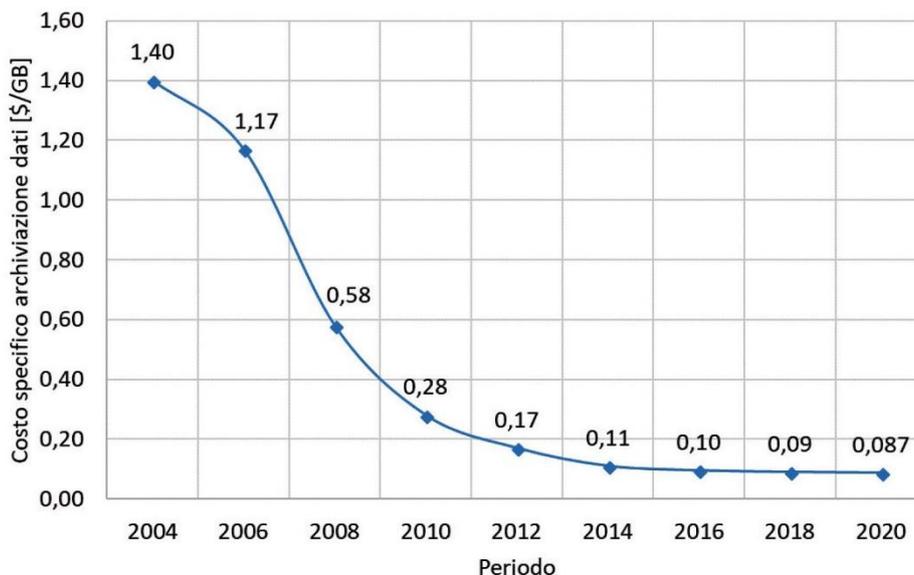


Figura 6: andamento prezzi dello storage di dati [5]

La diffusione di queste nuove piattaforme integrate e connesse alla Rete porta con sé molte delle problematiche delle strumentazioni precedenti, principalmente relative a Wireless Sensor Networks (WSNs),

Machine to Machine (M2M) e Cyber Physical Systems (CPS), ora integrate nei componenti IoT e ne aggiunge molte di nuove, la più importante è senza dubbio legata alla sicurezza.

Il concetto di sicurezza è andato ad assumere un peso sempre più importante. Questa tendenza è facilmente riscontrabile andando ad analizzare la produzione di articoli accademici relativi all'IoT negli ultimi anni; la Figura 7 mostra il rapporto fra gli articoli sull'IoT che trattano esplicitamente il tema della sicurezza rispetto al totale, pubblicato negli ultimi anni (dati basati sugli indici forniti da Google Scholar).

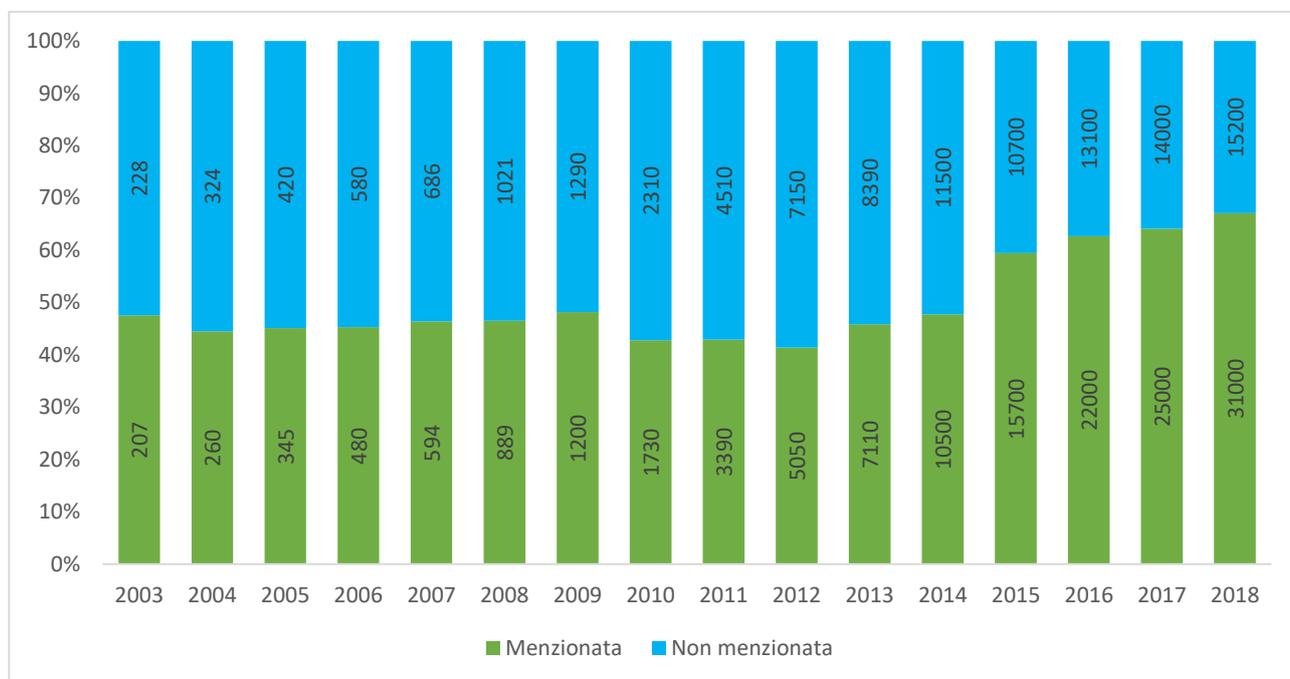


Figura 7: riferimenti espliciti alla sicurezza nell'IoT

L'IoT comprende una grande varietà di dispositivi per moltissime applicazioni differenti che vanno a comporre una rosa di possibili configurazioni differenti pressoché illimitata. Questo enorme livello di complessità comporta una quantità elevatissima di possibili punti deboli utilizzabili per violare il sistema. La letteratura solitamente distingue tre categorie di problematiche legate alla sicurezza nell'IoT:

- Basso Livello: è costituito dai problemi di sicurezza legati alla strumentazione hardware;
- Medio Livello: è costituito dai problemi di sicurezza legati alla comunicazione e trasmissione delle informazioni;
- Alto Livello: è costituito dai problemi di sicurezza legati ai software di gestione dell'IoT.

La Tabella 3 riporta alcuni esempi di vulnerabilità.

Tabella 3: vulnerabilità in ambito IoT [6]

Vulnerabilità	Descrizione	Implicazioni	Livello IoT	Soluzioni
Jamming	Intrusione in reti wireless senza protocolli di protezione	Interruzione o malfunzionamento del servizio	Basso	Implementazione di crittografia nella rete, cambio di frequenza
Interfaccia fisica non sicura	Punti di accesso fisico senza protezione	Violazioni della privacy, interruzione del servizio	Basso	Evitare accesso a software e firmware da USB, evitare tool di debugging

Sleep deprivation	Accensione forzata della strumentazione	Consumo eccessivo di energia	Basso	Sistemi di rilevamento di intrusione
Buffer reservation	Intrusione nei pacchetti di trasmissione dati (buffer)	Blocco della comunicazione fra il campo e il server	Medio	Implementazione di più canali di comunicazione
Sybil Attack	Intrusione di terzi per sfruttare la rete	Rallentamento del servizio	Medio	Controllo dei dispositivi connessi
Violazione del cloud server	Violazione dell'identità nell'accesso	Accesso a dati riservati, violazione della privacy	Medio	Identificazione in due passaggi
Interfacce non sicure	Mancanza di protezione di interfacce web, applicazioni mobile e cloud	Violazioni della privacy, interruzione del servizio	Alto	Utilizzo di connessioni sicure (https), utilizzo di password sicure
Software/firmware non sicuri	Vulnerabilità intrinseche dei software utilizzati, mancanza di aggiornamenti	Violazioni della privacy, interruzione del servizio	Alto	Cambio periodico delle credenziali accesso, autenticazione in due passaggi

L'aumento di interesse nel campo della sicurezza è anche dovuto alla diffusa diffidenza nei riguardi dell'IoT: molto spesso la grande quantità di sensori e apparecchiature che monitorano gli aspetti più disparati viene percepita non come un grande alleato ma come una possibile vulnerabilità. L'intero sistema, se violato da attori esterni malintenzionati in grado di controllarne ogni dettaglio, può essere facilmente utilizzato per raccogliere un gran numero di informazioni sensibili utilizzabili per spionaggio industriale o estorsioni. È proprio la diffidenza dal lato sicurezza la principale barriera contro una diffusione totale dell'IoT, la quale ne sta rallentando la crescita anche in applicazioni che ne trarrebbero indubbi vantaggi; un sondaggio fatto ad aziende ed imprenditori contenuto nel report dell'Intenet of Thing Institute del 2016 identifica le due principali ragioni che per le quali non viene applicato l'IoT:

- Privacy dei dati: dovuto all'enorme danno potenziale nella diffusione di materiale sensibile non autorizzato;
- Affidabilità: il maggiore livello di complessità di queste architetture aumenta il rischio di malfunzionamenti e problematiche altrimenti non presenti nei sistemi tradizionali.

Da questo punto di vista risulta molto promettente lo sviluppo di piattaforme per la trasmissione e scambio dei dati basati sulla tecnologia blockchain. La blockchain è una struttura dati condivisa costituita da blocchi concatenati fra loro che permette lo scambio di dati per mezzo di una rete Peer-to-Peer (P2P) mediante una curva di crittografia ellittica per garantire l'integrità e l'autenticazione dei dati trasmessi. Ogni blocco che costituisce la catena contiene tutte le informazioni relative alle transazioni dei blocchi adiacenti, in questo modo l'intera struttura integra lo storico completo di ogni movimento all'interno di essa garantendo un livello di sicurezza molto alto.

Nell'industria 4.0 cambia radicalmente la concezione di fabbrica come al conoscevamo fino a pochi anni fa, ci si riferisce infatti alla Fabbrica 4.0 o *Smart Factory* in cui i sistemi digitali raccolgono costantemente dati e misurazioni dei sistemi fisici creando modelli virtuali grazie ai quali sono in grado prendere decisioni relative alla gestione degli impianti in maniera autonoma e decentralizzata. [7]

Tutto questo è possibile grazie a quattro elementi cardine:

- L'interoperabilità, grazie alla quale le apparecchiature digitali sono in grado di connettersi e comunicare fra di loro autonomamente;
- La creazione di modelli virtuali modellati per simulare gli elementi fisici reali su cui si sta effettivamente lavorando;
- Il lavoro in sinergia fra essere umano e macchina;

- Capacità dei sistemi ciberfisici di prendere in maniera autonoma molte decisioni richiedendo agli esseri umani la necessità di intervento solo in casi eccezionali.

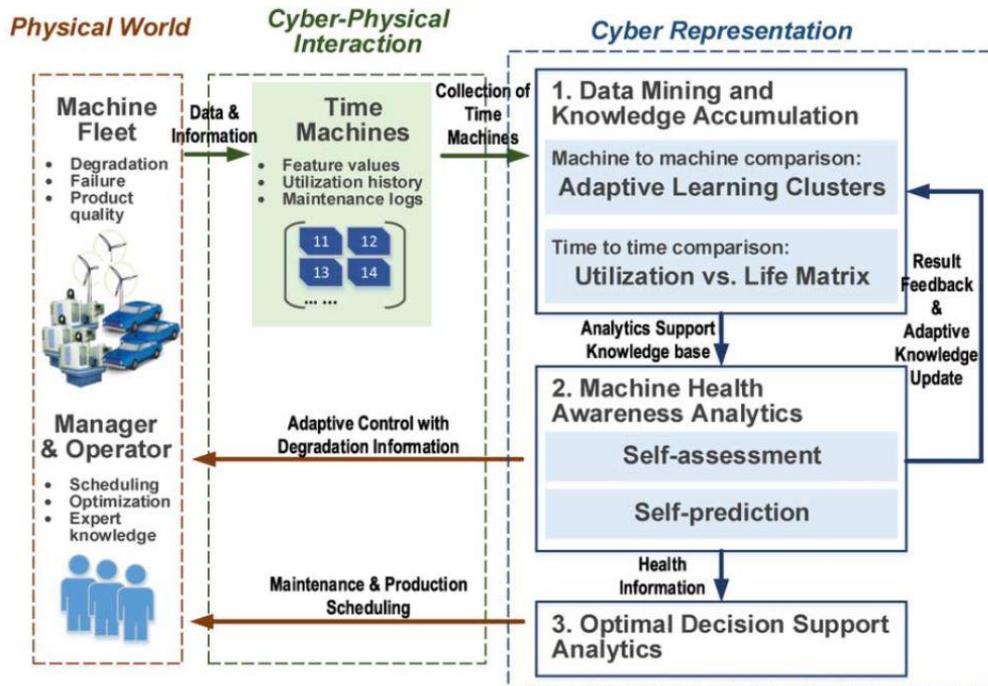


Figura 8: funzionamento di una smart factory [7]

Un settore industriale che ha beneficiato particolarmente di questa rivoluzione è sicuramente quello energetico, in particolare dal punto di vista dell'efficiamento dei consumi.

L'interconnessione fra sistemi hardware e software di monitoraggio e regolazione permette lo sviluppo di Energy Management System (EMS) e Building Energy Management System (BEMS) in grado di massimizzare l'efficienza energetica attraverso la raccolta e la trasmissione delle informazioni necessarie a garantire ottimizzazioni, e quindi risparmi, significativamente superiori rispetto a quanto era possibile fino a pochi anni fa. Questo è possibile grazie all'evoluzione dei sensori e all'integrazione di soluzioni IoT in essi; un moderno sistema di monitoraggio permette ai sensori di misura di connettersi alla rete Wi-Fi tramite un semplice router domestico e di trasmettere i dati direttamente ad un'unità di archiviazione cloud. I dati stoccati nel server in remoto possono essere consultati ed elaborati tramite le opportune piattaforme software da qualunque dispositivo connesso alla rete, anche dispositivi mobili come smartphone e tablet possono essere usati per controllare il monitoraggio in tempo reale tramite apposite app o via browser.

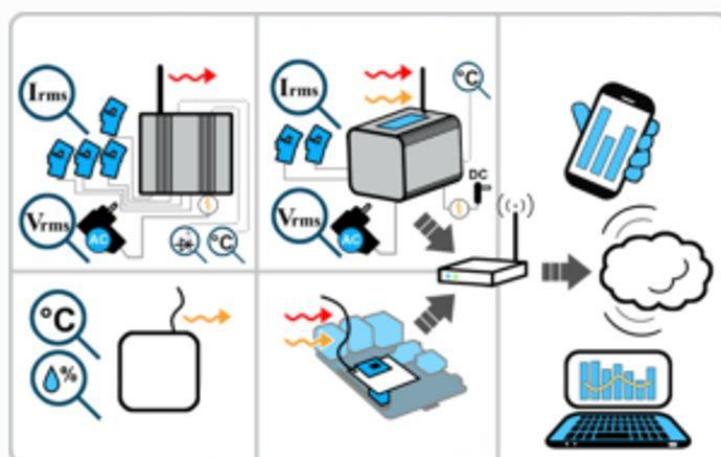


Figura 9: architettura IoT applicata al monitoraggio energetico

1.3.2 I nuovi strumenti software per la gestione

Il grande progresso nell'ambito dell'acquisizione dei dati sul campo è progredito parallelamente allo sviluppo di nuove piattaforme software in grado di sfruttare appieno la mole di informazioni disponibile.

La quantità di informazioni generata dalle moderne apparecchiature di controllo è talmente elevata da risultare ingestibile da un essere umano o anche da un normale database, tale da entrare nel campo dei così detti big data. Non esiste un valore prestabilito relativo alla dimensione di un dataset in grado di definire in maniera oggettiva se esso possa essere definito o meno big data, in primo luogo perché la dimensione tenderà sempre ad aumentare con passare del tempo, e in secondo luogo perché l'aspetto più importante riguarda la *qualità* dei dati oltre che la mera *quantità*.

I big data sono caratterizzati dalle cosiddette 5V:

- **Volume:** la mole di dati elaborati;
- **Velocità:** si riferisce alla rapidità con la quale i dati vengono raccolti e, soprattutto, analizzati;
- **Varietà:** si riferisce alle differenti tipologie di informazioni che possono essere raccolte da un gran numero di fonti differenti con grande frequenza;
- **Veridicità:** riguarda la qualità e l'attendibilità dei dati raccolti e la sicurezza che essi non siano stati alterati o modificati durante la trasmissione;
- **Valore:** è la capacità di estrarre e manipolare le informazioni effettivamente utili per incrementare la produttività dalla mole totale di informazioni.

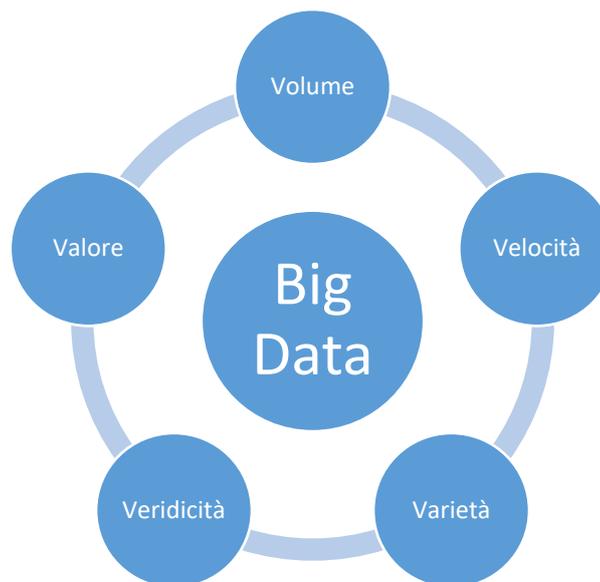


Figura 10: le 5V dei big data

Per poter utilizzare efficacemente i dati a disposizione, ossia effettuare analisi e previsioni sempre più accurate, si stanno evolvendo nuovi strumenti software appositamente preposti; grazie all'evoluzione nel campo dell'intelligenza artificiale e del machine learning si è introdotto il concetto di *smart software* che offre grandi opportunità nel campo del controllo e gestione, permettendo di conseguire risultati di ottimizzazione inarrivabili fino a pochi anni fa.

A differenza dei software di gestione tradizionale che consentono di visualizzare i consumi e gli indicatori passati e presenti, strumenti di Energy Management di nuova concezione permettono di compiere delle vere e proprie previsioni basandosi sui dati storici adoperando l'intelligenza artificiale.

Ciò consente ai software di evolversi da una semplice piattaforma di analisi e visualizzazione dei dati a supporto di un operatore umano ad uno strumento di ottimizzazione automatizzato, poiché le informazioni

e le analisi non sono più utilizzati dall'utente ma dal programma stesso in grado di intervenire attivamente sui macchinari. [5]

Una volta terminato l'addestramento relativo ad uno specifico impianto, mediante l'analisi di una grande quantità di dati storici, un algoritmo di machine learning è in grado di prevedere le curve di carico necessarie a soddisfare le richieste dell'utenza regolando l'impianto di conseguenza.

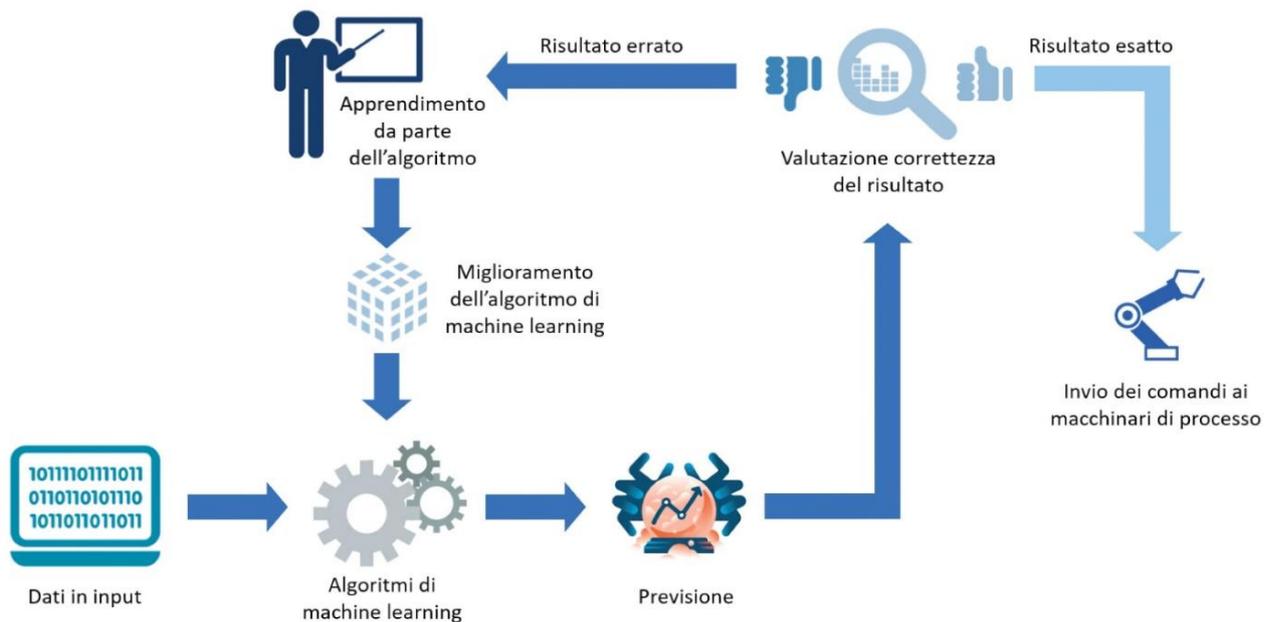


Figura 11: Architettura di un sistema basato sul machine learning [5]

In un impianto cogenerativo di un piccolo stabilimento industriale il sistema sarà in grado di regolare la potenza prodotta in ogni momento coerentemente con le abitudini e le necessità specifiche dell'azienda: aumentare o diminuire la produzione termica in base alla stagione o diminuire la produzione elettrica durante il cambio di turno e la pausa pranzo, e se le sue previsioni dovessero rivelarsi errate il programma stesso sarà in grado di imparare dai suoi stessi errori migliorando le previsioni successive.

1.3.3 L'implementazione dei nuovi strumenti negli impianti esistenti

La transizione da un'architettura tradizionale verso il nuovo paradigma non può avvenire attraverso step graduali, poiché la strumentazione è tuttora acerba o molto esosa dal punto di vista economico, tuttavia alcuni strumenti possono essere implementati in un'architettura già esistente e funzionante.

L'intervento più facilmente integrabile in un'architettura già esistente riguarda il livello di gestione, mediante l'utilizzo di algoritmi di **machine learning**. Grazie alla grande quantità di dati che già tutt'ora sono raccolti sul campo è possibile sfruttare le potenzialità fornite dall'intelligenza artificiale per migliorare l'analisi degli stessi, grazie allo sviluppo di modelli in grado di prevedere il funzionamento dell'impianto e di provvedere alla regolazione automatica.

2 Il Machine Learning nel monitoraggio energetico

2.1 Il Machine Learning e la Data Analysis

Il grande progresso nel campo della raccolta dei dati mette a disposizione una grande quantità di informazioni eterogenee fra loro, le quali una volta raccolte debbono essere opportunamente analizzate per trarne beneficio. Il processo di ispezione, pulizia, trasformazione e modellizzazione dei dati con lo scopo di estrarne informazioni utili è detto Data Analysis. La Data Analysis è un campo estremamente vasto che comprende numerosi approcci e tecniche in base alla natura dei dati a disposizione e del tipo di risultati che si persegue, dalla ricerca scientifica, economia e finanza, scienze sociali ed applicazioni tecniche. Le principali macrocategorie della Data Analysis sono:

- **Applicazioni statistiche:** gli studi statistici ricorrono a grandi quantità di informazioni utilizzabili in diverse applicazioni, la Data Analysis può essere applicata per studi descrittivi di un particolare fenomeno, per analisi esplorative (EDA-Explorative Data Analysis) o per la verifica delle ipotesi di un modello di studio (CDA- Confermatory Data Analysis);
- **Data Mining:** è una tecnica di analisi che si focalizza nella ricerca di correlazioni e pattern all'interno di grandi dataset. Il termine è spesso usato impropriamente poiché si riferisce all' "estrazione" di pattern dai dati a disposizione e non all'estrazione dei dati stessi. Il Data Mining rispetto ad altre tecniche di analisi è completamente rivolto alla creazione di modelli statistici utilizzati con scopi predittivi, in particolare grazie all'utilizzo del *machine learning*. Attualmente il compito di del Data Mining consiste nella analisi semi-automatizzata di grandi quantità di dati alla ricerca di correlazioni (cluster analysis), anomalie e dipendenze al fine di fornire uno strumento di supporto decisionale in numerosi processi.

Le analisi predittive si concentrano sulla creazione di modelli sia per la previsione di fenomeni, sia per problemi di classificazione, come l'analisi di testi nel capo della linguistica e nell'estrazione di informazioni da fonti visive o audiovisive, le cui applicazioni vanno dalla sicurezza negli aeroporti al marketing, grazie ad una intelligenza artificiale.

Il processo di Data Analysis è composto da varie fasi che consentono di ottenere una efficace ed affidabile estrapolazione delle informazioni desiderate. La prima fase è la **raccolta** dei dati, durante la quale vengono selezionati tutti i dati, provenienti dalle più diverse fonti, che saranno utilizzati nel corso dell'analisi. Segue l'**organizzazione** dei dati, dove le informazioni raccolte vengono organizzate in maniera omogenea in una forma adatta allo studio che si intende applicare, la struttura più utilizzata è un database organizzato in righe e colonne. In seguito occorre provvedere ad una **pulizia** dei dati, infatti molti elementi raccolti potrebbero essere incompleti, ripetuti o errati e il loro utilizzo comporterebbe errori e imprecisioni da parte del modello di studio. Solo dopo aver completato questi passaggi si può procedere con la **modellizzazione** vera e propria di un algoritmo in grado di identificare pattern ed eventuali correlazioni causa-effetto contenute nei dati. L'ultimo passaggio consiste nella produzione di un **risultato** dell'analisi in grado di fornire un supporto efficace al processo decisionale.

Con il termine *machine learning*, o apprendimento automatico in italiano, si intende un insieme di metodi e strumenti sviluppati nel campo dell'intelligenza artificiale col fine di rendere possibile ad un calcolatore l'apprendimento di pattern senza la necessità di una programmazione tradizionale esplicita, migliorando progressivamente la qualità dell'analisi effettuata. Il machine learning viene adoperato in tutte quelle situazioni in cui la scrittura di un algoritmo esplicito risulta impossibile o estremamente sconveniente per vari motivi, come ad esempio un numero eccessivo di regole complesse di cui tenere conto, il riconoscimento di immagini o suoni o lo studio di una grande mole di dati di cui non si conosce precisamente la correlazione [8].

L'apprendimento automatico viene spesso utilizzato per elaborare delle predizioni sulla base di dati precedenti sfruttando la principale peculiarità che lo contraddistingue: la capacità di generalizzare in base

alla propria esperienza e di produrre dei “ragionamenti induttivi” altrimenti impossibili con un programma tradizionale.

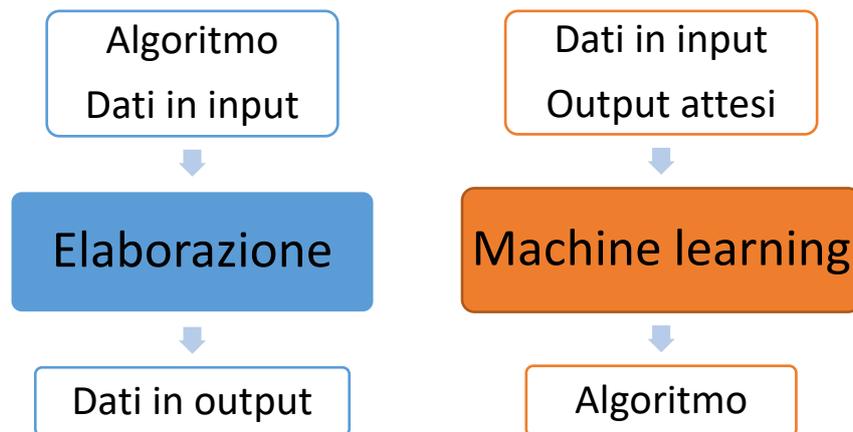


Figura 12: programma tradizionale (sinistra) e programma di machine learning (destra)

La caratteristica principale del machine learning è la capacità di imparare e quindi il processo di apprendimento riveste un ruolo cruciale nella fase di creazione del modello. Esistono due approcci principali a questo procedimento.

L'inconveniente principale legato all'utilizzo di algoritmi di machine learning è che non è possibile conoscere la struttura analitica dell'algoritmo che sviluppato, rendendo, di fatto, quasi impossibile valutare la bontà dello stesso in termini assoluti. Per questo motivo è spesso difficile rilevare o meno la presenza di *overfitting* nel modello. L'*overfitting*, o adattamento eccessivo, è un fenomeno che si presenta nei modelli statistici ad elevato grado di complessità e consiste nell'eccessivo adattamento del modello ai dati causato dal numero eccessivo di parametri rispetto al numero di osservazioni. In pratica un modello non corretto si può comunque adattare e fornire risultati apparentemente validi se risulta sufficientemente complesso, questo può avvenire se, ad esempio, i dati utilizzati per l'addestramento presentassero caratteristiche peculiari di quello specifico dataset, l'algoritmo riconoscerà quindi dei pattern inesistenti, perdendo la capacità di generalizzare nel momento in cui analizzerà dei dataset differenti. Sono stati sviluppati dei procedimenti per prevenire l'*overfitting*, tuttavia non esistono dei parametri oggettivi da misurare e il rischio di incappare in questo inconveniente è sempre presente.

All'interno dei sistemi di monitoraggio e gestione dell'energia gli algoritmi di machine learning svolgono un ruolo fondamentale: essi permettono di imparare nel dettaglio tutta la casistica di funzionamento di un impianto analizzando lo storico delle condizioni di funzionamento per fornire previsioni sul comportamento futuro al fine di effettuare una regolazione ottimale e un conseguente risparmio energetico. I possibili campi di applicazione in campo energetico sono molti, di seguito alcuni esempi:

- HVAC: negli impianti di climatizzazione, i quali costituiscono una consistente frazione nel bilancio energetico in ambito civile, risulta particolarmente interessante la gestione delle unità di trattamento aria di grandi dimensioni (centri commerciali, alberghi, fabbriche). L'intelligenza artificiale è in grado di memorizzare i profili di carico in base agli orari, ad esempio durante la pausa pranzo o il cambio di turni, e regolare di conseguenza il ricircolo di aria garantendo sia un risparmio energetico, riducendo od evitando la climatizzazione in determinati orari, sia le condizioni di confort ed igiene all'interno delle zone climatizzate attraverso un monitoraggio in tempo reale della qualità dell'aria [9];
- Impianti di cogenerazione: il carico di un impianto di cogenerazione all'interno di uno stabilimento industriale segue le richieste dei processi produttivi con cicli ricorrenti ed è quindi facilmente interpretabile dagli algoritmi che possono prevedere gli andamenti del fabbisogno. L'intelligenza artificiale risulta efficace anche in impianti di produzione di grandi dimensioni dedicati alla

generazione elettrica e costituiti da più motori che lavorano in contemporanea; la capacità di elaborare una gran quantità di dati, sia relativi all'impianto sia esterni, contemporaneamente consente di effettuare una regolazione ed una previsione estremamente efficiente [10];

- Fonti rinnovabili: molti impianti di produzione energetica combinano fonti tradizionali e rinnovabili, spesso queste ultime presentano cicli di produzione indipendenti dalle richieste dell'utenza e influenzati prevalentemente da variabili ambientali. Un impianto misto per la produzione di elettricità costituito da un motore a combustione interna e pannelli fotovoltaici presenta notevoli difficoltà nella regolazione della componente tradizionale: i pannelli fotovoltaici producono in base alle condizioni climatiche esterne, le quali non sono controllabili, andando ad influenzare le prestazioni del motore. Analizzando le previsioni metereologiche e la produzione del fotovoltaico è possibile stimare la parte di energia che deve essere integrata dal generatore [11].

Gli algoritmi di machine learning costituiscono quindi una grande opportunità di sviluppo ed evoluzione nel settore energetico consentendo di raggiungere obiettivi e prestazioni inarrivabili fino a poco tempo fa.

2.2 Le reti neurali artificiali

Le reti neurali artificiali, o Artificial Neural Network, sono una delle principali branche nel campo del machine learning. Sono dei sistemi computazionali che tentano di replicare la struttura delle reti neurali biologiche. L'unità principale è il nodo o perceptrone (N), che come il suo corrispettivo biologico, tramite dei collegamenti (L), comunica gli altri nodi; il sistema è uno spazio dimensionale dove $N \gg L$. I nodi sono organizzati in strati (*layer*).

La struttura base prevede:

- Un layer di ingresso, il quale riceve gli input che devono essere elaborati dal sistema;
- Uno o più layer intermedi (detti anche *nascosti*);
- Un layer finale che fornisce gli output.

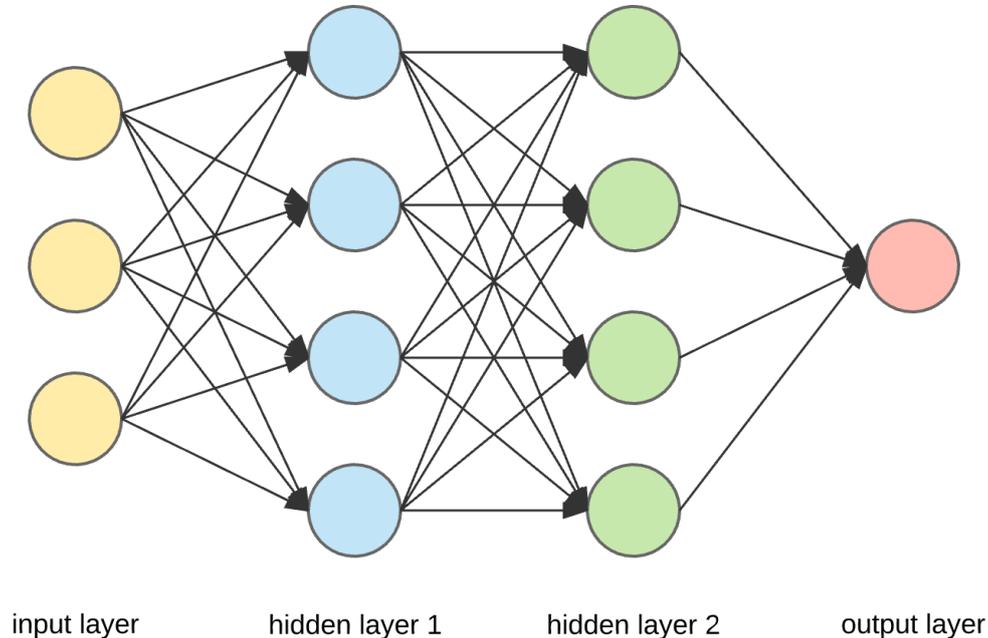


Figura 13: struttura base di una rete neurale artificiale

L'elemento fondamentale di una rete neurale è il *perceptrone*, il quale simula il neurone biologico. Il perceptrone è formato da tre elementi fondamentali:

1. Un numero variabile di connessioni (replica delle sinapsi biologiche), ognuna della quali caratterizzata da un peso, che, diversamente dal corrispettivo biologico, può assumere anche un valore negativo;

2. Un sommatore, il quale somma gli input provenienti dalle varie connessioni in base al loro peso, producendo come output una combinazione lineare degli input ricevuti;
3. Una funzione di attivazione che serve a limitare l'ampiezza dell'output. Solitamente l'output è compreso fra i valori [0,1] o [-1,1].

Il perceptrone comprende anche un valore soglia che permette di amplificare o ridurre l'input netto elaborato dalla funzione di approssimazione.

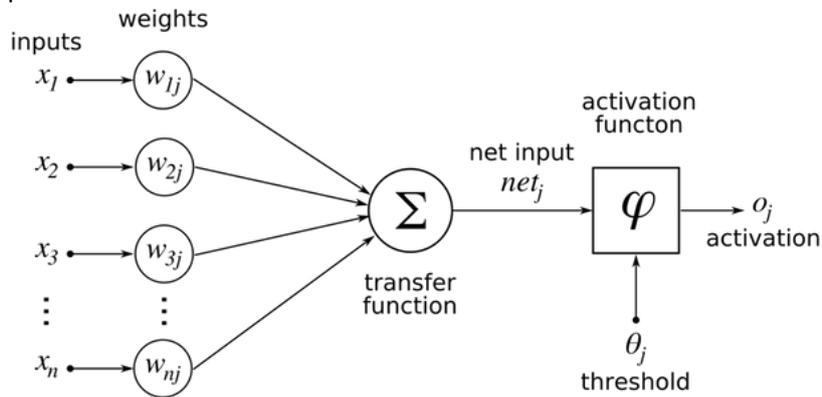


Figura 14: schematizzazione matematica di un perceptrone

Da un punto di vista matematico un neurone k può essere rappresentato dalle seguenti equazioni:

$$u_k = \sum_{j=1}^m w_{kj} \cdot x_j \quad (1)$$

$$y_k = \varphi(u_k + b_k) \quad (2)$$

Dove:

- x_i sono i pesi sinaptici del neurone k;
- u_k è la combinazione lineare degli input del neurone k;
- b_k è il valore di soglia del neurone k;
- $\varphi(x)$ è la funzione di attivazione;
- y_k è l'output elaborato dal neurone k.

2.2.1 Funzioni di attivazione

Ogni neurone che compone uno strato riceve degli input dai neuroni degli strati precedenti che vengono combinati in un unico valore che rappresenta l'input della funzione di attivazione.

Le funzioni di attivazione devono rispettare dei criteri: ricevere degli input normalizzati e fornire un output quanto più possibile prossimo ad 1 se adeguatamente stimolata. Esistono tre tipi di funzione di attivazione generalmente utilizzati nelle reti neurali.

- La *funzione gradino* di Heaviside è la più semplice, è una funzione discontinua che restituisce 0 per tutti valori negativi e 1 per tutti quelli positivi. È definita come:

$$\varphi(v) = \begin{cases} 1 & \text{se } v \geq 0 \\ 0 & \text{se } v < 0 \end{cases} \quad (4)$$

L'andamento di questa funzione è il seguente:

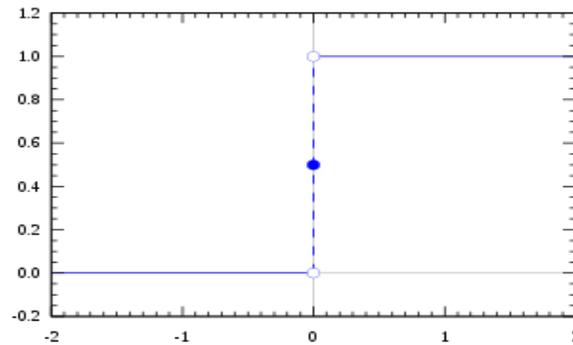


Figura 15: funzione gradino di Heaviside

- *Funzione lineare a tratti*: spesso la funzione gradino risulta troppo semplice e viene utilizzata come base per una funzione definita ad hoc.

$$\varphi(v) = \begin{cases} 1 & \text{se } v \geq \frac{1}{2} \\ v & \text{se } -\frac{1}{2} < v < \frac{1}{2} \\ 0 & \text{se } v \leq -\frac{1}{2} \end{cases} \quad (5)$$

L'andamento di questa funzione è il seguente:

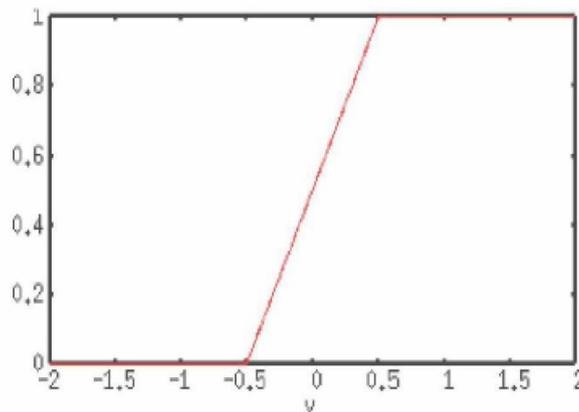


Figura 16: funzione lineare a tratti

- *Funzioni sigmoidee*: sono le funzioni maggiormente utilizzate. Queste funzioni sono strettamente crescenti e comportano un bilanciamento fra un andamento lineare ed un andamento non lineare. Sono definite come:

$$\varphi(v) = \frac{1}{1 + e^{-av}} \quad (6)$$

Dove a è un parametro che permette di variare la pendenza della funzione si noti come per $a \rightarrow +\infty$ la funzione approssima l'andamento della funzione gradino. Questa funzione viene utilizzata perché spesso è utile una normalizzazione dei dati nell'intervallo $[-1,1]$.

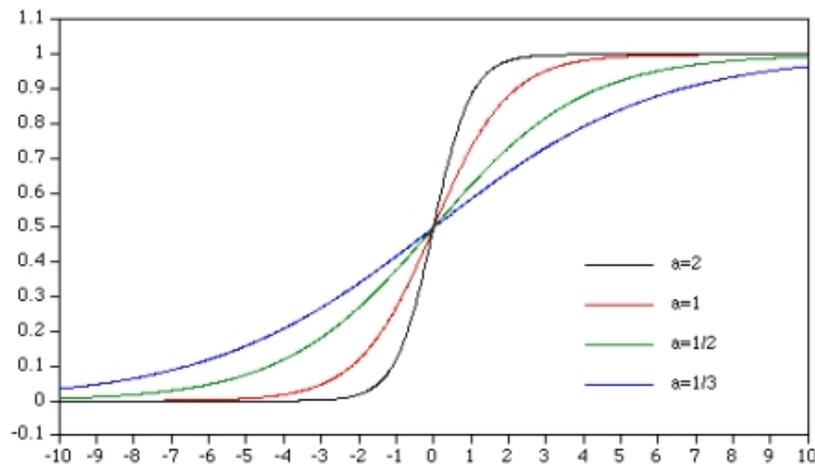


Figura 17: funzione di Sigmoid al variare di a

A questo gruppo di funzioni appartengono la tangente iperbolica $\varphi(v) = \tanh(a \cdot v)$ e la funzione rettificatore $\varphi(v) = v^+ = \max(0, v)$ indicata come ReLU (dall'inglese Rectified Linear Unit).

Lo strato finale di una rete necessita un tipo particolare di funzione di attivazione, la più utilizzata è la funzione softmax.

$$g_k(T) = \frac{e^{T_k}}{\sum_{l=1}^K e^{T_l}} \quad (7)$$

Dove T è il vettore di output.

2.2.2 Apprendimento

Apprendimento Supervisionato

L'apprendimento supervisionato è la soluzione più utilizzata per l'addestramento delle reti neurali. Nell'apprendimento supervisionato vengono sottoposti alla rete degli input ognuno dei quali è correlato a degli specifici output, detti etichette (labels). L'insieme di questi input e dei rispettivi labels costituisce il *training set*.

Nel processo di apprendimento i parametri liberi della rete, ossia il peso dei collegamenti, vengono modificati, attraverso una stimolazione, rispetto alle condizioni in cui essa è inserita.

Come visto in precedenza, esistono varie strategie per effettuare l'apprendimento della rete, il più utilizzato è l'apprendimento attraverso la riduzione dell'errore.

$$e_k(n) = d_k(n) - y_k(n) \quad (8)$$

Dove:

- $d_k(n)$ è la risposta attesa dal k-esimo neurone;
- $y_k(n)$ è la risposta generata dal sistema dal k-esimo neurone;
- $e_k(n)$ è il segnale di errore del k-esimo neurone.

Il segnale di errore attiva un procedimento di autocontrollo del sistema attraverso una serie di aggiustamenti del peso di ogni connessione del k-esimo neurone con l'obiettivo di ridurre la differenza fra la risposta generata e la risposta attesa.

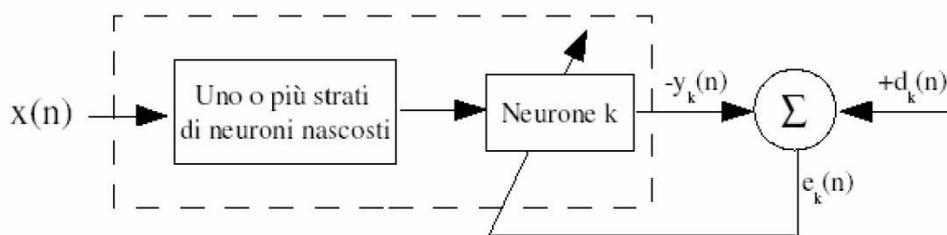


Figura 18: apprendimento con correzione dell'errore

Questo procedimento avviene uno step alla volta andando a minimizzare una funzione costo:

$$E(n) = \frac{1}{2} \cdot e_k^2(n) \quad (9)$$

La minimizzazione della funzione costo avviene attraverso il metodo della discesa del gradiente. Chiamati $w_{kj}(n)$ i pesi delle connessioni del k-esimo neurone eccitati da un input $x_j(n)$ l'aggiustamento effettuato su $w_{kj}(n)$ è definito come:

$$\Delta w_{kj} = \eta \cdot e_k(n) \cdot x_j(n) \quad (10)$$

Dove η è una costante positiva chiamata tasso di apprendimento. I nuovi pesi così calcolati saranno:

$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n) \quad (11)$$

Il meccanismo di apprendimento attraverso la correzione dell'errore costituisce un sistema ricorrente in cui la stabilità dello stesso dipende dai parametri che lo costituiscono, in modo particolare da η poiché influenza direttamente la convergenza e la stabilità dell'intero processo.

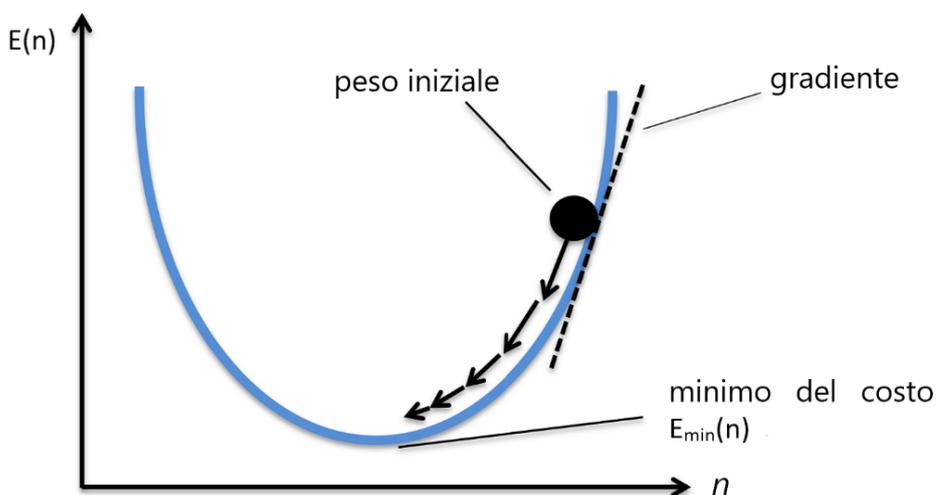


Figura 19: discesa del gradiente

In seguito all'addestramento le prestazioni della rete sono valutate per mezzo di un insieme di input e labels differenti dal training set, detto *validation set*, attraverso il quale viene testata la capacità di generalizzazione da parte della rete (è fondamentale che questi dati non siano mai stati visualizzati in precedenza dal sistema). I principali algoritmi per l'apprendimento supervisionato sono:

- K-Nearest neighbors (KNN);
- Logistic Regression (LR);

- Support Vector Machine (SVM);
- Decision Tree (DT);
- Random Forest (RF);
- Artificial Neural Network (ANNs).

Apprendimento non supervisionato

Nell'apprendimento non supervisionato vengono presentati alla rete un insieme di input a cui, diversamente rispetto all'apprendimento supervisionato, non sono associati a delle etichette. La classificazione degli input sarà svolta dal sistema andando ad analizzare i dati e trovando caratteristiche in comune fra di essi e provvedendo ad elaborare la classificazione.

Questo genere di apprendimento è utilizzato nei problemi di clustering, nei quali la conoscenza delle correlazioni fra i dati non è nota a priori e il compito della rete neurale consiste proprio nella loro individuazione; l'utilizzo di un apprendimento supervisionato risulterebbe impraticabile per questa tipologia di problemi. [8]

2.2.3 Attività di apprendimento

L'attività di apprendimento può essere eseguita in vari modi.

Associazione di Pattern

Viene sviluppata una memoria associativa, ossia una memoria distribuita che apprende per associazioni. Le associazioni possono essere di due tipi:

- **Autoassociazioni:** vengono memorizzati i pattern presentando alla rete gli input varie volte. Quando alla rete viene presentato un nuovo input essa cerca di richiamare i pattern memorizzati anche nel caso in cui questi siano parziali o distorti. Solitamente il processo di apprendimento non è supervisionato;
- **Eteroassociazioni:** durante l'apprendimento ogni pattern viene associato ad un altro pattern. Si ha dunque un pattern chiave x_k al quale viene associato un pattern y_k che viene memorizzato. Il pattern x_k funge da stimolo per richiamare il pattern y_k . Rispetto al caso precedente ($x_k = y_k$) l'abbinamento è $x_k \neq y_k$.

Il processo di associazione avviene in due fasi: nella prima, detta di memorizzazione, avviene l'associazione dei pattern, nella seconda, detta di richiamo, il pattern memorizzato viene richiamato in maniera integrale o parziale da uno stimolo. [8]

Riconoscimento di Pattern

In questo procedimento il pattern viene associato ad una specifica classe o categoria. In seguito al processo di addestramento (nel quale il pattern viene presentato più volte) la rete è in grado di riconoscere i pattern che le vengono sottoposti in base alle categorie con le quali sono stati classificati. Nel caso in cui venisse sottoposto alla rete un pattern mai incontrato in precedenza, la rete provvederà a creare una nuova categoria sfruttando le conoscenze apprese dai pattern già noti.

Ogni pattern costituisce un singolo punto all'interno dello spazio di decisioni multidimensionale. Lo spazio decisionale è composto da varie regioni ad ognuna delle quali è associata una categoria. Il processo di apprendimento definisce il confine fra le varie regioni.

Il riconoscimento di pattern può essere effettuato in due modi:

- La rete viene suddivisa in due parti. Una sottorete non supervisionata che estrapola le caratteristiche ed una sottorete supervisionata che effettua la classificazione;
- L'attività di estrapolazione e classificazione avviene in ogni singolo strato nascosto che compone la rete.

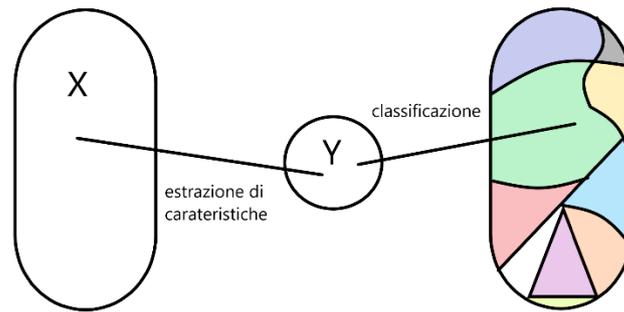


Figura 20: classificazione dei pattern

Approssimazione di funzioni

Una mappatura input-output non lineare può essere descritta dalla relazione:

$$d = f(x) \tag{12}$$

Dove:

- x rappresenta l'input;
- d rappresenta l'output;
- f è la funzione incognita che relazione input e output.

Vengono fornite numerose coppie di elementi input-output descritte da dalla relazione f:

$$\{(x_i, d_i)\}_{i=1}^n \tag{13}$$

Queste coppie vengono usate dalla rete neurale per creare una funzione F che approssimi la funzione f tale per cui la mappa di input e output realizzata dalla rete sia vicina in senso euclideo ad ogni input:

$$\|F(x) - f(x)\| < \varepsilon, \forall x \tag{14}$$

Dove ε è un numero positivo sufficientemente piccolo, tanto più piccolo quanto maggiore è il numero di parametri liberi che compongono la rete. L'approssimazione di funzioni è particolarmente indicata per l'apprendimento di supervisionato che utilizza vettori in input. [8]

2.2.4 Generalizzazione

Lo scopo finale di una rete è quello di sviluppare la capacità di generalizzare efficacemente una tipologia di problemi. Una generalizzazione è corretta quando la mappa di input e output generata durante la fase di apprendimento è corretta per degli esempi di test ma visionati in precedenza dalla rete (il set di dati usati per il test appartiene alla medesima popolazione dei dati usati per il training).

La capacità di generalizzare è simile ad un problema di approssimazione di una curva: considerando la rete neurale come una mappa non lineare degli input e degli output, una buona generalizzazione equivale ad una buona interpolazione rispetto input dati.

Una generalizzazione efficace consente di produrre una mappatura corretta anche se gli input utilizzati sono lievemente differenti rispetto a quelli utilizzati durante l'addestramento; una rete che ha visionato troppi esempi (coppie input output) nella fase di training corre il rischio di perdere la capacità di generalizzare

correttamente andando a memorizzare informazioni non necessarie. Questo fenomeno è chiamato overfitting.

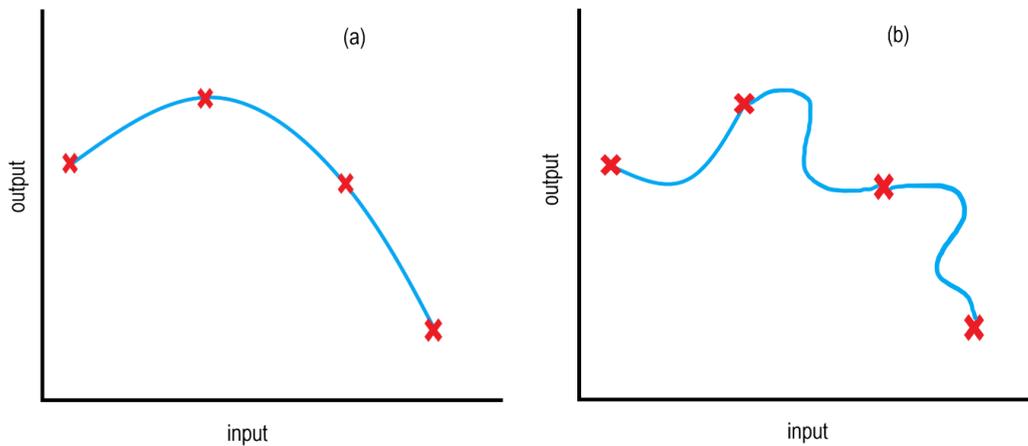


Figura 21: confronto fra una buona generalizzazione (a) e una cattiva generalizzazione (b) dovuta ad overfitting

La generalizzazione è funzione di tre fattori:

- La dimensione del set di dati usato per addestrare il modello;
- La dimensione e il tipo di reti neurali;
- La complessità del problema analizzato.

La complessità del problema che si vuole studiare con la rete neurale è un fattore sul quale non è possibile apportare modifiche, dunque per migliorare le prestazioni della rete occorre intervenire sugli altri due punti. Nel caso in cui la struttura della rete sia prefissata bisogna stabilire la dimensione ottimale del set di dati da usare nell'addestramento, nel caso in cui sia prefissata la quantità di dati del training set bisogna determinare la struttura ottimale della rete. [8]

Una regola empirica che consente di ottenere buone generalizzazioni consiste nel rispettare la seguente condizione:

$$N = O\left(\frac{W}{\varepsilon}\right) \tag{15}$$

Dove:

- N è la dimensione del set di dati usato per l'addestramento;
- W è il numero di parametri liberi della rete;
- ε è la frazione di errori di classificazione sul set di dati usato per il test.

Di seguito è riportato un diagramma di flusso che schematizza il processo di addestramento di una rete neurale.

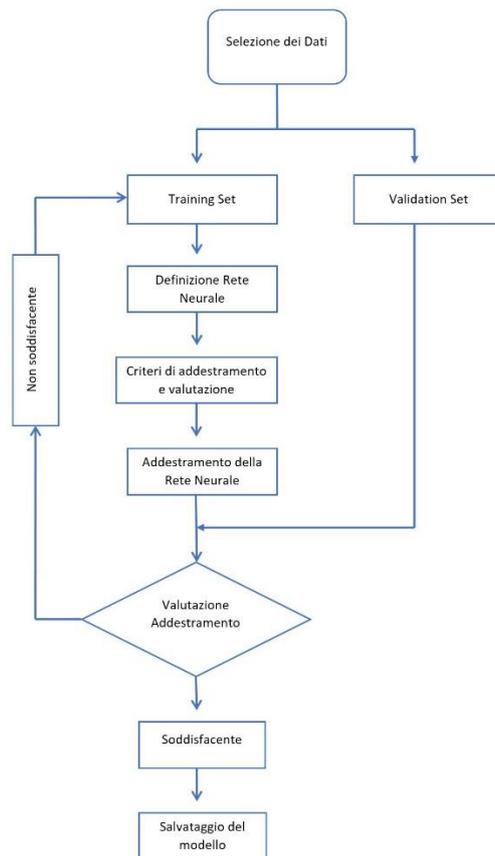


Figura 22: processo di addestramento di una rete neurale

2.2.5 La funzione obiettivo e l'ottimizzatore

Il modello creato necessita di avere una funzione da minimizzare (Loss) ed un ottimizzatore. In base al fenomeno analizzato si adoperano diverse funzioni, le due più utilizzate sono `binary_crossentropy` (nel caso in cui occorra calcolare delle probabilità) e `mean_squared_error` (usata nei modelli di predizione). Esistono vari ottimizzatori implementati nelle API Keras, ognuno con le sue peculiarità ed adatti ad uno specifico modello di calcolo. Il più utilizzato è `AdamOptimizer`, particolarmente performante nelle applicazioni stocastiche.

2.2.6 Prevenzione di Overfitting e Underfitting

Come visto in precedenza il modello viene allenato a riconoscere un determinato pattern attraverso la fase di apprendimento e ci si aspetta che sia in grado di *generalizzare* questo pattern anche con dati differenti, ossia di predire i risultati anche su dati che non ha mai visto in precedenza. Tuttavia, se l'apprendimento è stato eccessivo si corre il rischio che il modello perda la capacità di generalizzare poiché è troppo adattato ai dati usati per istruirlo (tipicamente si focalizza su alcuni aspetti tipici esclusivamente del training set, ma assenti nella maggior parte degli altri casi), questo fenomeno è detto **overfitting**.

Esiste anche l'eventualità contraria, cioè che il modello non sia stato addestrato in maniera sufficiente. Questo fenomeno può verificarsi per vari motivi: i dati analizzati non sono organizzati in maniera adeguata, il modello richiede una complessità della rete (numero di strati e di neuroni) maggiore o i dati analizzati non sono abbastanza. Il tutto si traduce di nuovo nell'incapacità di generalizzare.

Spesso risulta difficile valutare la qualità di un modello in maniera univoca, ma esistono alcune regole empiriche che permettono di effettuare una valutazione qualitativa sull'addestramento della rete.

È buona norma implementare una cross-validation, ossia la divisione del training set in due insiemi distinti:

- Training sub-set: effettivamente usato per l'addestramento del modello;
- Validation sub-set: usato per la validazione del modello.

Osservando l'andamento reciproco dell'accuratezza dei due set di dati all'aumentare delle epoche dell'addestramento è possibile individuare eventuali fenomeni di overfitting e underfitting.

Un modello in underfitting ha una precisione migliore sul training sub-set rispetto al validation sub-set e questo può essere facilmente identificato se la curva del training sta al di sotto della curva del validation e quest'ultimo ha un andamento che suggerisce un miglioramento nel caso di un numero maggiore di epoche.

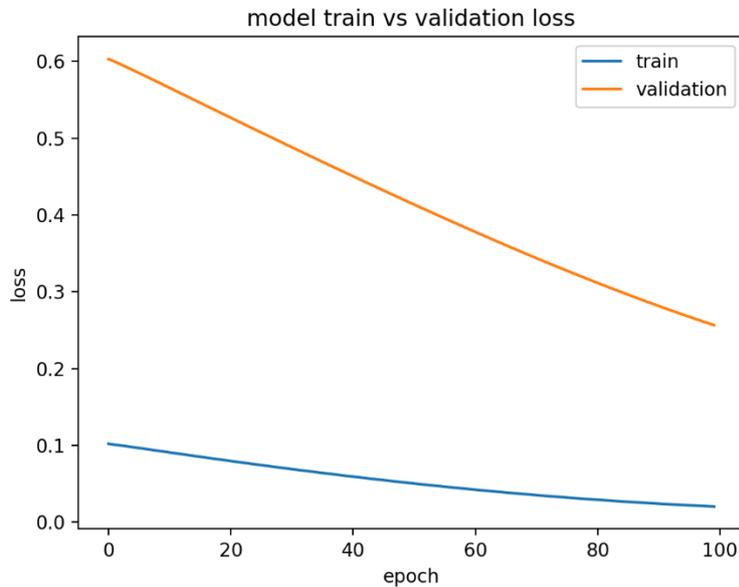


Figura 23: modello in underfitting

Un modello con delle buone performance presenta un andamento convergente del training e validation set, in cui entrambi diminuiscono e si stabilizzano attorno allo stesso valore.

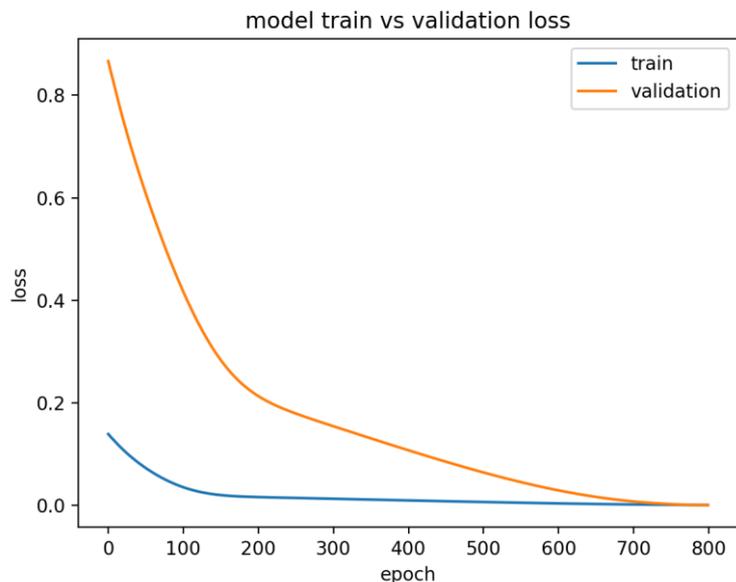


Figura 24: modello con un addestramento efficace

In fine un modello affetto da overfitting è caratterizzato da prestazioni del training set che continuano ad aumentare mentre quelle del validation, ad un certo punto, tendono a peggiorare. Si nota solitamente che la curva di validation ha un minimo oltre il quale cambia pendenza.

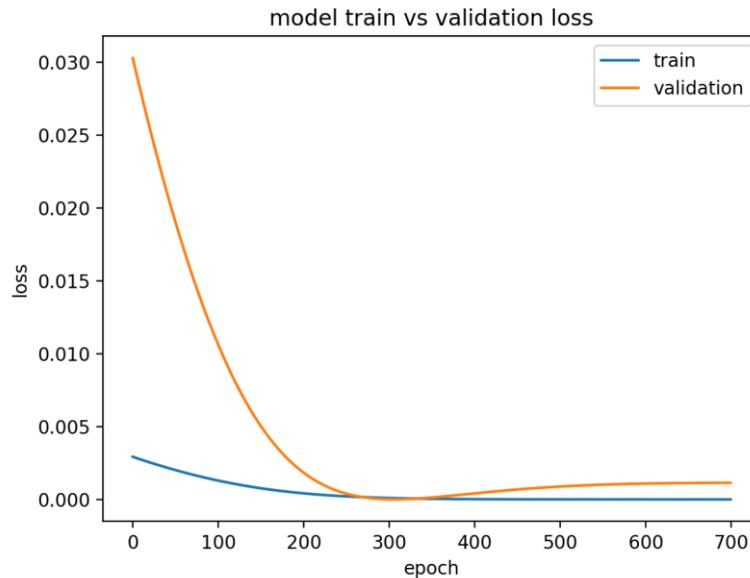


Figura 25: modello in overfitting

Tensorflow, grazie alle API Keras, consente di implementare alcune funzioni utili a prevenire l'overfitting e l'underfitting, di seguito sono illustrate quelle utilizzate in questa analisi.

Early Stopping

In linea di principio il modo migliore per addestrare il modello consiste nel fargli visionare il maggior numero di dati differenti possibile, ma come abbiamo visto il rischio di incorrere in overfitting è sempre presente e, purtroppo, non esistono dei parametri analitici che possono indicare in maniera univoca la presenza di overfitting. Si può tuttavia ricorrere ad alcuni accorgimenti nella scrittura del codice che permettono di ridurre al minimo l'overfitting o riuscire ad intuire quando esso si verifica.

Una possibilità consiste nell'implementare una funzione che controlli il miglioramento della predizione rispetto ai dati attesi mano a mano che aumenta il numero di epoche ed interrompere l'addestramento nel momento in cui non si registrino più miglioramenti significativi. Questo metodo è detto "early stopping" ed è facilmente implementabile in Tensorflow attraverso una specifica funzione. [12] [13]

```

model = build_model()

# The patience parameter is the amount of epochs to check for improvement
early_stop = keras.callbacks.EarlyStopping(monitor='val_loss', patience=20)

history = model.fit(train_data, train_labels, epochs=10000,
                    validation_split=0.2, verbose=0,
                    callbacks=[early_stop, PrintDot()])

plot_history(history)

```

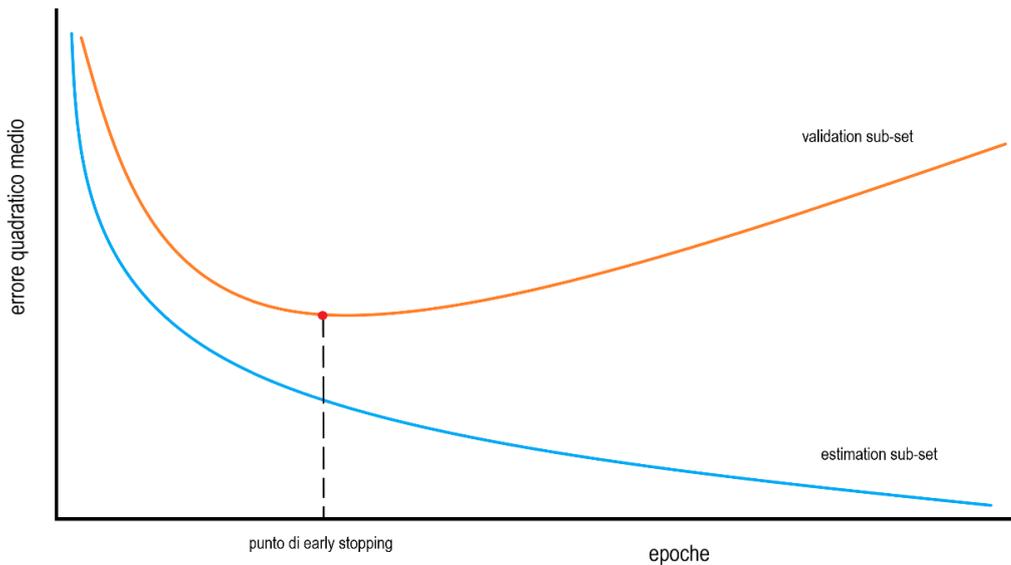


Figura 26: metodo di "early stopping"

Questa operazione viene fatta per validare la rete con dei dati differenti da quelli utilizzati nella fase di training, è così possibile valutare il set di addestramento calcolando l'errore alla fine di ogni epoca. Osservando l'andamento dell'errore quadratico medio riferito all'estimation sub-set si nota come esso si riduca all'aumentare delle epoche portando a pensare che la precisione aumenti in maniera inversamente proporzionale ad esse. Tuttavia una volta superato il punto di early stopping la rete apprende solamente il rumore contenuto nei dati, quindi proseguire l'addestramento risulta dannoso.

Weight regularization

Un'altra possibilità è consistere nel "weight regularization": il modello che si viene a creare all'interno della rete è determinato dal peso specifico che ha ogni connessione fra i nodi della rete, più un modello è complesso e maggiore è il numero di possibili combinazioni che possono descriverlo efficacemente. Per mitigare l'overfitting si può forzare il modello a sviluppare il set più semplice possibile aggiungendo un costo alla funzione obiettivo associato ad avere dei pesi elevati.

Può avvenire in due modi:

- Regolarizzazione **L1**: il costo aggiunto è proporzionale al valore assoluto del peso dei coefficienti;
- Regolarizzazione **L2**: il costo aggiunto è proporzionale al quadrato del peso dei coefficienti. Questa tipologia di regolarizzazione è detta anche *decadimento del peso*.

Anche in questo caso è prevista una funzione specifica da implementare quando si crea il modello.

```
model = keras.Sequential([
    keras.layers.Dense(128, kernel_regularizer=keras.regularizers.l2(0.001),
                        activation=tf.nn.relu),
    keras.layers.Dense(10, activation=tf.nn.softmax) ])
```

Dropout

L'ultimo metodo, detto "dropout", consiste nell'eliminare alcuni output di un layer durante la fase di apprendimento. Questo sistema consiste nell'eliminazione casuale di un certo numero valori di un layer durante l'addestramento.

Immaginiamo di avere un layer che, dato un input, restituisca in output il vettore $V=[0.2, 0.5, 1.3, 0.8, 1.1]$; applicando il drop out alcuni elementi del vettore vengono modificati in zero, ottenendo un novo vettore in

output $V^1=[0, 0.5, 1.3, 0.1.1]$. Il rateo di dropout è la frazione dei valori che vengono tramutati in zero, solitamente compreso fra il 20% e il 50%.

```
dpt_model = keras.models.Sequential([
    keras.layers.Dense(16, activation=tf.nn.relu, input_shape=(NUM_WORDS,)),
    keras.layers.Dropout(0.5),
    keras.layers.Dense(16, activation=tf.nn.relu),
    keras.layers.Dropout(0.5),
    keras.layers.Dense(1, activation=tf.nn.sigmoid) ])
```

Il dropout viene applicato attraverso lo specifico comando implementato

2.3 Classificazione delle reti neurali

Esistono molti tipi di reti neurali, possono essere suddivisi in base alla struttura, agli operatori matematici che implementano e alle applicazioni in cui vengono utilizzate, ognuna con i suoi pro e contro. Le principali categorie sono riassunte nella Tabella 4: Prestazioni delle reti.

Tabella 4: Prestazioni delle reti

Tipologia di rete	Ottime prestazioni	Buone prestazioni
Feed Forward	-Dataset tabulari -Prevedere una classificazione -Prevedere una regressione	-Classificazione di immagini -Classificazione di testi -Serie temporali
Convoluzionali	-Classificazione di immagini -Prevedere una classificazione -Prevedere una regressione	-Classificazione di immagini -Serie temporali -Sentiment analysis
Ricorrenti	-Classificazione di testi -Comprensione del linguaggio -Prevedere una regressione -Prevedere una classificazione	-Serie temporali

2.3.1 Feedforward

Le reti feedforward costituiscono il modello più semplice di struttura neurale. In queste reti le informazioni si muovono sempre in un'unica direzione e i layer intermedi sono densi, ossia ogni neurone è connesso con tutti i neuroni dello strato precedente e successivo (*fully connected*). La struttura molto semplice consente a questo tipo di rete di essere molto performante nei problemi di classificazione e regressione. Hanno lo svantaggio di essere tendenti all'overfitting quando la rete diventa molto complessa (molti layer e molti neuroni).

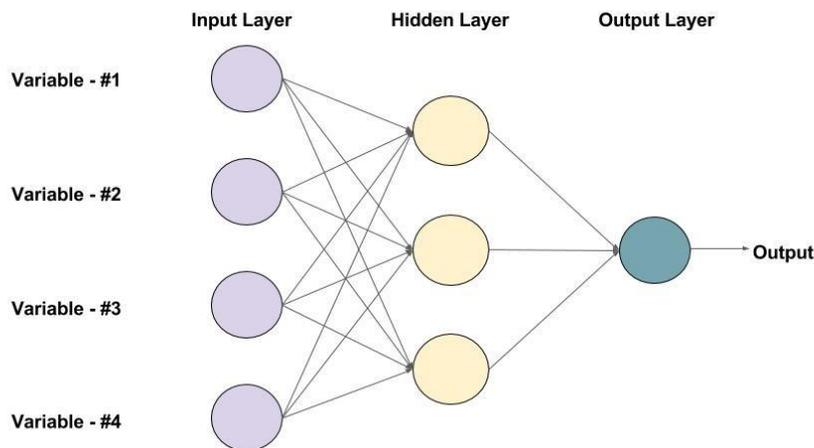


Figura 27: rete feedforward con uno strato nascosto

Le reti dette Radial Basis network sono un particolare tipo di reti feedforward, si differenziano dal caso generale per l'utilizzo di un particolare tipo di funzione invece della classica funzione logistica (Sigmoid). Le funzioni logistiche generano una scala di valori da 0 ad 1 e possono rispondere a domande con risposte binarie (sì o no), tuttavia lavorano male quando devono maneggiare valori continui, le funzioni radial basis, invece, valutano la distanza del risultato ottenuto dall'obiettivo che si vuole raggiungere. Risultano quindi ottime per problemi in cui si verifica necessariamente un'approssimazione, come, ad esempio, il controllo di macchine operatrici.

Questo genere di reti è ottimo per i problemi di predizione dove gli input forniti sono classificati in label e nei problemi di regressione dove occorre predire dei valori basandosi su di uno storico. Data la notevole flessibilità possono tuttavia essere utilizzate anche per il riconoscimento di immagini e testi.

2.3.2 Convolutionali

Le reti convoluzionali elaborano le informazioni sotto forma di array multipli e sono le più utilizzate per il riconoscimento di immagini. Estraggono le caratteristiche di un'immagine attraverso la convoluzione, partendo dai bordi, e man mano che si procede attraverso la rete migliora il livello di precisione con la quale vengono riconosciuti i dettagli che formano l'immagine. Un'immagine a colori, ad esempio, viene rappresentata come tre array bidimensionali contenenti l'intensità dei tre colori primari per ogni singolo pixel.

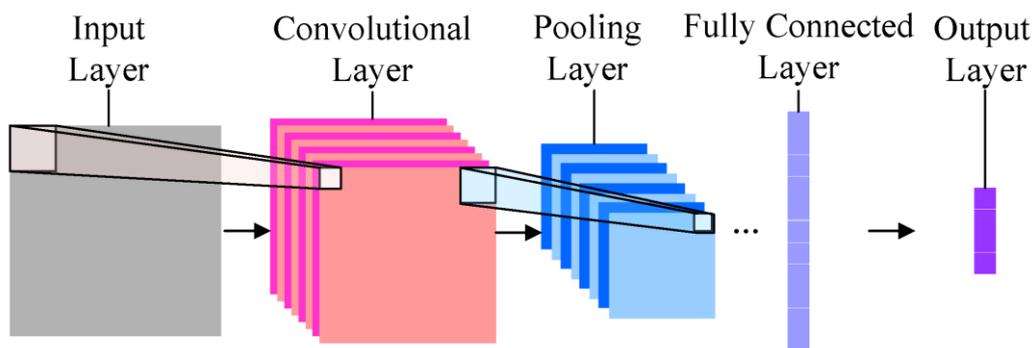


Figura 28: convolutional network

La principale differenza rispetto alle altre tipologie di reti neurali è costituita dalla loro capacità di operare direttamente sulle immagini e non sulle caratteristiche estratte dalle stesse. Le reti convoluzionali sono costituite da blocchi strati: i primi blocchi sono *convolutional layer* e i secondi *pooling layer*.

Le reti convoluzionali utilizzano il concetto di connettività locale, cioè la capacità di sfruttare le correlazioni spaziali presenti all'interno dei dati in input e che si propagano lungo gli strati della rete. Gli input dei neuroni dello strato n sono gli output di un sottoinsieme dei neuroni dello strato $n-1$ e hanno campi ricettivi con una contiguità spaziale.

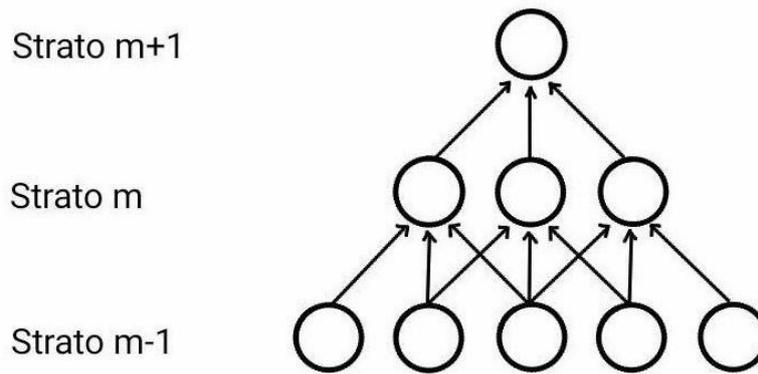


Figura 29: struttura interna

Supponendo che lo strato $m-1$ sia quello di input, i nodi dello strato m risultano avere un campo di ricezione con ampiezza 3 rispetto all'input, poiché ognuno di essi è connesso con solo tre nodi del layer precedente, mentre i nodi dello strato $n+1$ hanno un'ampiezza di 3 rispetto allo strato n e un'ampiezza di 5 rispetto allo strato $n-1$. Ogni unità di un layer elabora le informazioni che appartengono al proprio campo ricettivo indipendentemente da eventuali variazioni esterne ad esso. Andando a sovrapporre più strati connessi fra di loro le informazioni elaborate da ogni nodo hanno un carattere sempre più globale man mano che si sale di livello, aumentando il livello di astrazione. La connettività locale, dunque, consiste nel fatto che i dati organizzati sotto forma di array, in particolare nel caso di immagini, i sottoinsiemi locali e adiacenti di dati sono fortemente correlati fra loro e possono essere agevolmente riconosciuti dalla rete.

I nodi degli strati convoluzionali che compongono la rete sono connessi ai sottoinsiemi dei nodi degli strati precedenti per mezzo di un insieme di pesi, chiamato *filter bank*, con cui viene svolto il processo di convoluzione vero e proprio. Il valore dei pesi è inizialmente casuale e vengono mano a mano aggiornati durante l'addestramento della rete per mezzo dell'algoritmo di back propagation.

Un'altra peculiarità delle reti convoluzionali è la condivisione dei pesi, i nodi degli strati convoluzionali sono organizzati in gruppi detti feature map condividendo fra di essi il valore di pesi. Le feature map comprendono tutti gli input del layer precedente e, per mezzo del filter bank condiviso, elaborano l'immagine e possono andare individuarne le caratteristiche peculiari indipendentemente dalla loro posizione.

Il vantaggio nell'utilizzo di queste reti è la loro abilità di sviluppare una rappresentazione interna di immagini 2D, permettendo al modello di imparare la posizione e la scala di molteplici strutture (i.e. riconoscere un gilè da una camicia dalla presenza o meno di maniche).

Queste reti sono particolarmente performanti la classificazione di immagini e, in generale, con tutti i dati legati relazioni spaziali (come la disposizione di parole in un testo, *sentiment analysis*). [12] [13]

2.3.3 Recurrent-Long Short Term Memory

Le reti ricorrenti, a differenza delle altre tipologie di reti, nelle quali gli input vengono trattati come ognuno indipendente dagli altri, analizzano le informazioni in modo sequenziale tenendo conto dell'ordine con cui essi sono presentati al sistema.

La rete elabora gli elementi sequenziali uno per volta mantenendo la memoria di quelli precedenti e hanno la capacità di propagare i dati non solo in avanti ma anche in senso inverso. Questa caratteristica è particolarmente utile quando si analizzano dati di lunghezza variabile, come frasi o tracce audio. Sono utilizzate per estrarre un elemento (ad esempio una singola parola) e capire il suo significato all'interno del contesto.

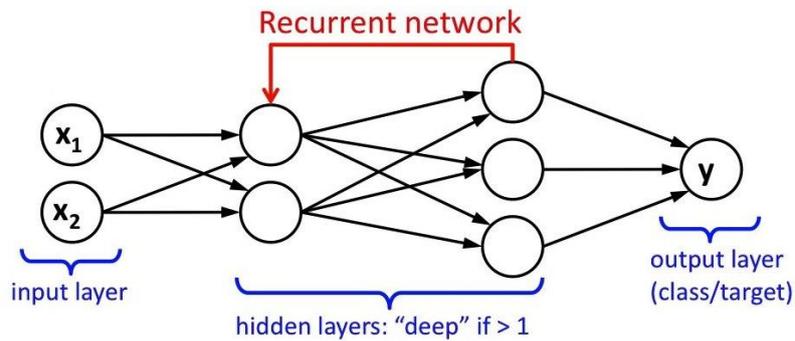


Figura 30: rete ricorrente

Durante la fase di addestramento queste reti possono essere affette dal fenomeno di *vanishing* o *exploding gradient*, poiché i gradienti che vengono propagati all'indietro hanno la tendenza ad aumentare o diminuire per ogni istante temporale e far quindi divergere verso infinito o convergere verso zero.

Per ovviare a questo problema le reti LSTM sono dotate di particolari unità nascoste, dette celle di memoria che hanno la capacità di ricordare memorizzare gli input precedenti per lunghi periodi. Queste celle comparano l'input dell'istante corrente con quello dell'istante precedente e decidono quale mantenere in memoria, garantendo la capacità di memorizzare le informazioni importanti durante tutto il procedimento di apprendimento.

Sono state progettate per lavorare con problemi di predizione sequenziale, ossia problemi in cui è fondamentale la dipendenza temporale. Risultano quindi molto performanti nella comprensione di un discorso (sia scritto sia in forma audio) e per generare modelli che forniscono output in sequenza. Non sono adatti per la classificazione di immagini.

2.3.4 Reti modulari

Le reti modulari sono composte assemblando più reti di tipo diverso fra loro, le quali lavorano in maniera indipendente fra di loro ma contribuiscono ognuna all'output finale. Il vantaggio di una rete modulare è che ogni singola rete che la compone ha un livello di complessità relativamente basso. Riducendo il rischio di overfitting e consentendo di poter effettuare delle modifiche mirate senza rischiare di stravolgere la struttura generale.

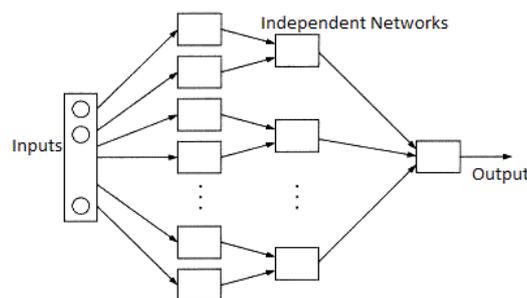


Figura 31: rete modulare

2.4 Modellizzazione di una rete neurale

Esistono molti software che permettono lo sviluppo di reti neurali, uno dei più utilizzati è Tensorflow, sviluppato da Google e basato sul linguaggio Python. Il progetto Tensorflow si propone l'obiettivo di permettere l'utilizzo di API molto complesse, come Keras, attraverso un'interfaccia relativamente semplice e adatta a chi si avvicina per la prima volta al machine learning.

L'idea di base consiste nell'utilizzare le varie istruzioni delle API Keras come dei mattoncini da costruzione, i quali possono essere facilmente assemblati fra loro sfruttando poche righe di codice. Gli step logici da seguire per creare una rete neurale sono i seguenti:

- 1) **Raccogliere un dataset:** questa è la parte più delicata del processo. Occorre infatti che i dati raccolti siano archiviati con criterio e coerenza fra di loro; spesso, nei casi reali, si ha a che fare con gruppi di dati estremamente eterogenei fra di loro e una gran parte del lavoro consiste nell'assicurarsi che essi vengano caricati e manipolati in maniera corretta dal programma. È buona prassi visualizzare a schermo i dati dopo averli caricati per verificare che la loro formattazione si effettivamente quella desiderata.

```
(train_data, train_labels), (test_data, test_labels) = dataset.load_data()
```

- 2) **Costruire il modello:** questo step rappresenta il cuore del procedimento, tuttavia sono sufficienti poche righe di codice per creare la rete neurale vera e propria.

```
model = keras.Sequential([
    keras.layers.Dense(128, activation=tf.nn.relu),
    keras.layers.Dense(10, activation=tf.nn.softmax) ])
```

Una volta inizializzata la rete vanno definiti i vari layer. In questo caso la rete è composta da tre strati: il primo (quello di input) non deve essere definito, il secondo strato (nascosto), composto da 128 nodi in questo caso, è il componente principale e la maggior parte dei processi avvengono al suo interno, il terzo ed ultimo è il layer di uscita, composto da 10 nodi, fornisce l'output del processo.

Aggiungere ulteriori strati o nodi al modello aumenta la capacità dello stesso di riconoscere i pattern presentati, tuttavia si corre il rischio di aumentare la *memorizzazione* di uno specifico set di dati. Questo inconveniente, detto *overfitting*, riduce la capacità del modello di generalizzare il pattern acquisito poiché troppo legato al training set usato e quindi poco performante su altri dataset.

```
model.compile(optimizer=tf.train.AdamOptimizer(),
              loss='binary_crossentropy',
              metrics=['accuracy'])
```

Occorre definire l'ottimizzatore e il criterio da seguire per minimizzare la funzione Loss, che rappresenta la differenza fra l'output del sistema e l'output atteso. Esistono vari criteri, a seconda del tipo di problema affrontato. Gli argomenti necessari alla funzione sono:

- optimizer:** definisce la funzione di ottimizzazione;
- loss:** definisce la funzione obiettivo che deve essere minimizzata dall'ottimizzatore;
- metrics:** definisce la funzione metrica. Serve a valutare le performance del modello.

Gli ottimizzatori utilizzabili sono:

- **SGD:** gradiente stocastico discendente. Include il supporto per il momentum e il Nesterov momentum;
- **RMSprop:** ottimizzatore performante sulle reti neurali ricorsive;
- **Adagrad:** ottimizzatore con parametri di apprendimento specifici, i quali vengono adottati relativamente a quanti frequentemente un parametro viene aggiornato durante la fase di apprendimento. Più un parametro viene aggiornato, minore sarà l'aggiornamento.
- **Adadelta:** è una estensione più robusta di Adagrad che adatta il tasso di apprendimento in base al gradiente degli aggiornamenti, invece di accumulare tutti i precedenti gradienti.
- **Adam:** ottimizzatore settato di default. Sostanzialmente un ottimizzatore RMSprop con momentum;
- **Adamax:** variante di Adam basata sulla norma infinita;
- **Nadam:** ottimizzatore Nesterov Adam, ossia un Adam con Nesterov momentum;

Le funzioni obiettivo disponibili sono:

- **mean_squared_error:** errore quadratico medio;
- **mean_absolute_error:** errore assoluto medio;

- **mean_absolute_percentage_error**: errore assoluto medio percentuale;
- **mean_squared_logarithmic_error**: errore quadratico medio in scala logaritmica;
- **ogcosh**: logaritmo del coseno iperbolico dell'errore predetto. $\log(\cosh(x))$ è approssimativamente uguale a $(x ** 2) / 2$ per piccoli valori di x e a $\text{abs}(x) - \log(2)$ per valori elevati di x. Questo significa che "logcosh" funziona circa come l'errore quadratico medio ma tende ad essere meno affetto da errori casuali di predizione;
- **categorical_crossentropy**: modello basato sul metodo Monte Carlo il cui output è una probabilità. Utile nelle reti che devono maneggiare dati stocastici. Manipola input di tipo float32 o float64.
- **sparse_categorical_crossentropy**: uguale a categorical_crossentropy ma utilizza solo dati in formato int32 e int64;
- **binary_crossentropy**: sostanzialmente uguale alle funzioni crossentropy precedenti, restituisce un output binario (0 oppure 1). Utile se sono presenti sol due tipologie di label;
- **kullback_leibler_divergence**: applica la divergenza di Kullback-Leibler o entropia relativa rappresenta la divergenza non simmetrica fra due distribuzioni probabilistiche.
- **poisson**: applica la distribuzione di Poisson, utile per applicazioni con dati discreti;
- **cosine_proximity**: utilizza la tecnica euristica della similarità del coseno, usato prevalentemente nell'analisi dei testi.

In fine le funzioni metriche, le quali servono a valutare le performance del modello. Sono simili alle funzioni obiettivo (loss), ma con la differenza che i risultati della funzione metrics non sono usati per addestrare il modello. Le funzioni metrics disponibili sono:

- **binary_accuracy**: calcola l'accuratezza media per le predizioni di tipo binario;
- **categorical_accuracy**: calcola l'accuratezza media per tutti le predizioni effettuate;
- **sparse_categorical_accuracy**: equivalente alla categorical_accuracy ma particolarmente performante per i problemi con variabili sparse;
- **mean_squared_error**: calcola il mse fra i valori predetti e i valori attesi;
- **mean_absolute_error**: calcola il mae fra i valori predetti e i valori attesi;
- **mean_absolute_percentage_error**: calcola il mape fra i valori predetti e i valori attesi;
- **mean_squared_logarithmic_error**: calcola il male fra i valori predetti e i valori attesi;
- **hinge**: calcola la hinge loss (?), definita come $\max(1 - y_{\text{true}} * y_{\text{pred}}, 0)$;
- **categorical_crossentropy**: calcola la cross-entropy nei problemi di classificazione. Nota: funziona con le matrici, non coi vettori;
- **sparse_categorical_crossentropy**: come la precedente, ma si aspetta un array composto da interi;
- **precision**: calcola la precisione, utile per problemi multi-label;
- **recall**: simile a precision;
- **fbeta_score**: calcola l'Fscore, ossia la media pesata di precision e recall;

3) Apprendimento: ora il modello deve essere "addestrato" dandogli da analizzare un set di dati appositamente preparato. Come accennato in precedenza, vengono forniti al modello una serie di dati reali e i risultati attesi, anch'essi reali, e il sistema procede con l'elaborazione dell'algoritmo che definisce le relazioni esistenti fra i due. Questa operazione richiede una sola riga di codice.

```
history = model.fit(train_data, train_labels, epochs=40)
```

Il parametro "epochs"(epoche) definisce il numero di volte che i dati vengono processati dall'intero sistema, questo è un aspetto molto delicato. Più i dati vengono riprocessati dalla rete, maggiore è l'accuratezza del modello, tuttavia questo processo non scala in modo lineare.

Si individua quindi un punto di minimo oltre il quale aumentare le epochs sarebbe poco redditizio e si rischierebbe di incorrere in overfitting.

Il processo di apprendimento può richiedere molto tempo, in base alla complessità della rete e della quantità di dati analizzati, ed è quindi possibile salvare il modello addestrato e ricaricarlo in un secondo momento per analizzare nuovi dati senza ripetere il training.

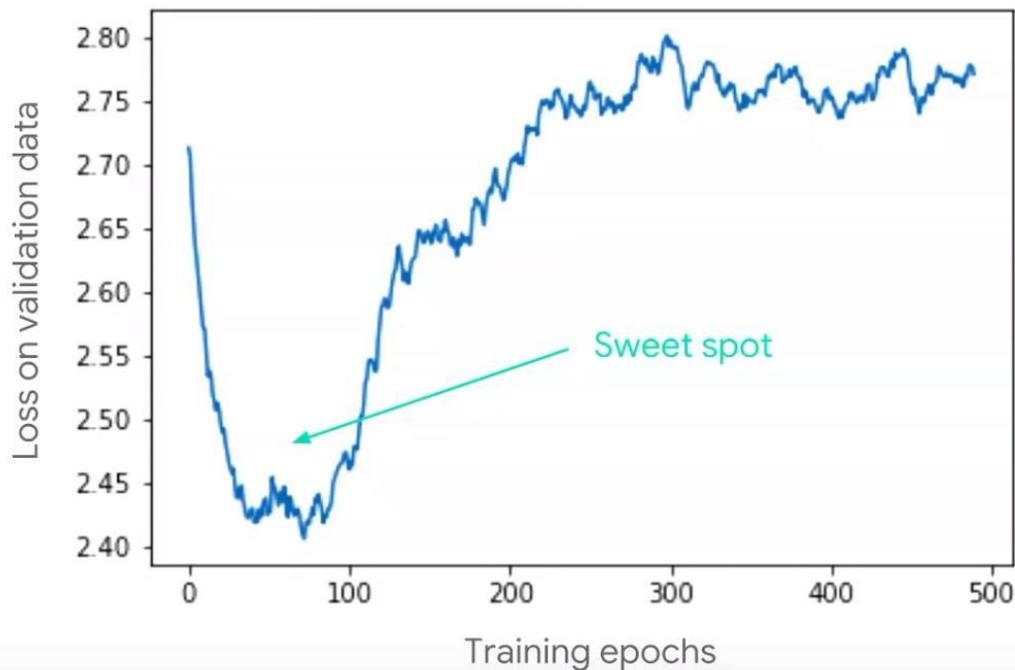


Figura 32: andamento tipico della funzione Loss

- 4) **Valutazione:** dopo la fase di apprendimento il modello deve essere testato facendogli elaborare un set di dati che non ha mai visto in precedenza. I risultati del modello vengono comparati con i risultati attesi (reali).

```
results = model.evaluate(test_data, test_labels)
```

- 5) **Predizione:** ora il modello è pronto per essere utilizzato. Fornendo dei dati coerenti rispetto a quelli utilizzati in fase di apprendimento è grado di predirne i risultati efficientemente.

```
prediction = model.predict(new_data)
```

3 Esempi di applicazioni del Machine Learning in ambito energetico

3.1 Tipologia degli impianti analizzati

In questo studio si andrà a sviluppare un modello software in grado di predire l'andamento del fabbisogno di energia elettrica di due stabilimenti produttivi nell'ora immediatamente successiva e il fabbisogno di un'intera giornata fornendo i dati del giorno precedente, entrambi dotati di un impianto di cogenerazione autonomo, permettendo, da un lato, la capacità di effettuare una migliore regolazione della produzione, poiché conoscendo in anticipo il carico di lavoro dell'impianto di cogenerazione, questo può essere pre-regolato automaticamente operando sempre nelle condizioni ottimali, evitando in questo modo sbalzi improvvisi nella produzione con conseguente diminuzione del rendimento e, dall'altro lato, la capacità di rilevare immediatamente eventuali malfunzionamenti o processi non in linea con il comportamento atteso dello stabilimento, poiché sarebbero facilmente individuabili a causa dello scostamento dalla predizione.

Verranno analizzati gli impianti di cogenerazione di due stabilimenti produttivi, i quali utilizzano energia sia elettrica sia energia termica durante il ciclo produttivo:

- Un **lanificio**: l'energia elettrica è utilizzata per l'alimentazione dei numerosi macchinari e per le apparecchiature ausiliarie (impianto di illuminazione, computer, sistemi di ventilazione, ecc...) mentre l'energia termica è utilizzata prevalentemente sotto forma di vapore ad alta pressione e temperatura direttamente nella lavorazione dei tessuti;
- Un **industria alimentare**: in questo caso l'energia termica, prevalentemente sotto forma di acqua calda viene utilizzata per portare alle temperature desiderate gli ingredienti utilizzati nella preparazione alimentare.

La cogenerazione è la produzione simultanea di energia elettrica ed energia termica a partire da un'unica fonte, all'interno di un unico sistema integrato. L'utilizzo di un impianto di cogenerazione consente nella capacità di riutilizzare il calore usato per la produzione di elettricità, il quale, nella produzione tradizionale, andrebbe disperso in ambiente.

La cogenerazione ha luogo negli impianti di cogenerativi, solitamente composti dall'accoppiata di un motore collegato ad un alternatore per la produzione di energia elettrica e un sistema adibito al recupero del calore. L'energia termica recuperata può essere impiegata per riscaldamento civile, sistemi di raffreddamento ad assorbimento o per impieghi industriali. La cogenerazione permette di risparmiare fino al 30% di energia primaria rispetto alla produzione separata dei due vettori (elettrico e termico), risultando estremamente vantaggiosa sia da un punto di vista economico sia da un punto di vista ambientale poiché riduce significativamente l'emissione di CO₂ in atmosfera.

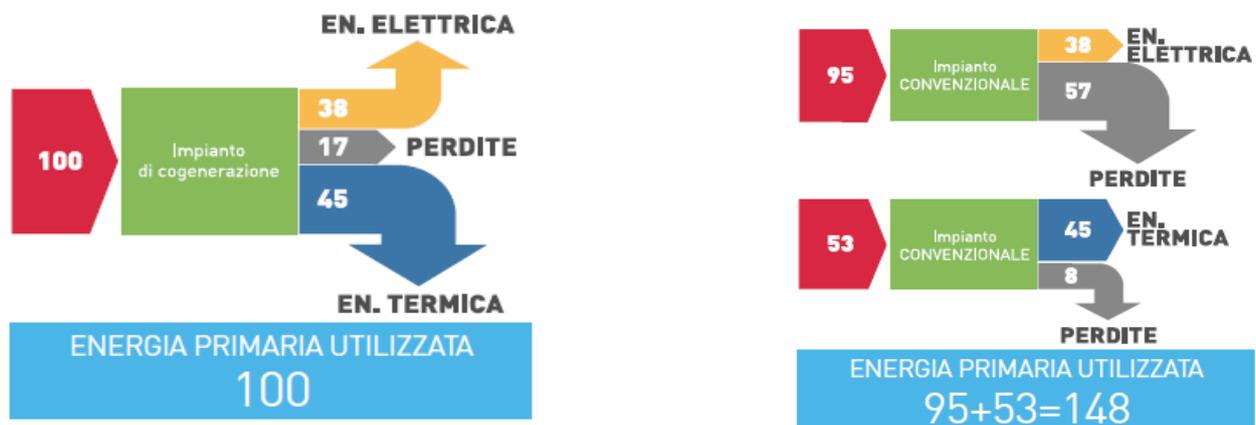


Figura 33: produzione tramite cogenerazione e produzione separata

Molti stabilimenti industriali hanno al loro interno un impianto di produzione di energia elettrica al loro interno per garantire un approvvigionamento di energia indipendente dalla rete nazionale e necessitano di vettori termici nei processi produttivi, di conseguenza l'adozione di impianti cogenerativi risulta essere una soluzione particolarmente vantaggiosa per numerose aziende.

3.2 Impostazione dello studio

3.2.1 Modelli e strumenti utilizzati

Prima di procedere con lo sviluppo vero e proprio di un software basato sulle reti neurali per supportare la gestione energetica di un impianto di produzione, è necessario compiere una fase preliminare in cui si impostano gli step di lavoro:

- **Definire l'obiettivo dello studio:** la rete neurale sarà sviluppata con lo scopo di fornire una previsione relativa al fabbisogno futuro di potenza elettrica dello stabilimento;
- **Acquisire i dati necessari all'analisi:** i set di dati utilizzato è costituito dalle misurazioni prese sul campo, attraverso apparecchi SCADA, delle principali grandezze relative ad un impianto di cogenerazione. In questo studio ci si concentra in particolare sulla produzione elettrica, di conseguenza le grandezze analizzate saranno la potenza elettrica prodotta localmente, la potenza elettrica prelevata dalla rete e i riferimenti temporali;
- **Scegliere i modelli da utilizzare:** come visto nel capitolo precedente, esistono numerose tipologie differenti di reti neurali, ognuna con le sue caratteristiche e peculiarità. È dunque fondamentale utilizzare un modello performante per la tipologia di studio che si intende effettuare;
- **Sviluppare i modelli:** una volta scelti modelli più appropriati si può sviluppare ed addestrare la rete neurale;
- **Valutare i modelli:** l'ultimo passaggio consiste nella valutazione dei modelli, andando ad analizzare le previsioni fornite e i parametri statistici caratteristici.

Come visto nel capitolo precedente, esistono numerose tipologie differenti di reti neurali, ognuna con le sue caratteristiche e peculiarità. È dunque fondamentale utilizzare un modello performante per la tipologia di studio che si intende effettuare.

I dati in nostro possesso costituiscono una serie di valori legati fra loro da correlazioni temporali da utilizzare per compiere una regressione e prevedere l'andamento futuro. Dalla Tabella 4 si nota come le due tipologie che consentono di ottenere le migliori performance per questa tipologia di problemi sono le reti Feed Forward (FF) e le reti ricorrenti Long Short Term Memory (LSTM). Si è quindi scelto di sviluppare parallelamente entrambe le soluzioni e di confrontarle fra di loro per valutare la soluzione ottimale.

Per ogni tipologia di rete si sono sviluppati due modelli distinti:

- Uno in grado di prevedere la potenza elettrica richiesta nell'ora successiva, basandosi sulle potenze precedenti;
- Uno in grado di prevedere l'andamento dell'intera richiesta elettrica di una giornata basandosi sulla curva di produzione della giornata precedente.

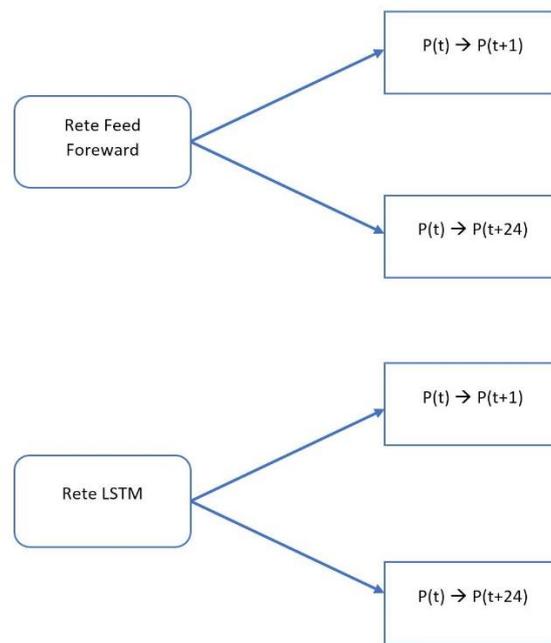


Figura 34: schema dei modelli

Gli strumenti utilizzati nello sviluppo del modello sono:

- **Python:** è un linguaggio di programmazione ad alto livello adatto alla computazione numerica, gratuito e molto flessibile. Sono disponibili moltissime librerie, anch'esse gratuite, che ampliano enormemente le possibilità di sviluppo. Rispetto a molti altri software che permettono di sviluppare algoritmi di machine learning Python può essere utilizzato anche senza avere una conoscenza approfondita del linguaggio ed è supportato da una vastissima community on-line;
- **Tensorflow:** è una libreria per il machine learning sviluppata da Google che fornisce moduli testati ed ottimizzati molto facili da usare per merito della sintassi estremamente sintetica ed intuitiva. Include al suo interno le API Keras che sono una delle più utilizzate librerie nell'ambito dell'apprendimento automatico. Tensorflow è pienamente compatibile con i linguaggi Python, C++, Java R e Scala. Si consiglia l'utilizzo su macchine dotate di una scheda per l'accelerazione grafica dedicata, in quanto i calcoli sui tensori risultano molto più performanti se eseguiti da una GPU rispetto ad una CPU (non è possibile utilizzare l'accelerazione grafica nei sistemi MacOS di Apple)
- **Scikit-Learn:** è una libreria Python per il machine learning. Al suo interno sono presenti numerosi comandi e funzioni per la classificazione, il clustering e la regressione e l'analisi statistica. Questa libreria è ideata per funzionare assieme Numpy;
- **Numpy:** acronimo di Numeric Python, è una delle più diffuse librerie per applicazioni scientifiche ed include moltissime funzioni precompilate per calcoli matematici e scientifici. È molto performante nella computazione di array multidimensionali e matrici;
- **Pandas:** è una libreria utilizzata per manipolare database in modo semplice e intuitivo. È progettata per funzionare con archivi relazionali e labeled, risulta quindi molto utile nella creazione di dataframe da utilizzare in un addestramento supervisionato (il sistema adoperato in questo studio). Viene utilizzato in coppia con Numpy.

3.2.2 Indici utilizzati

Per valutare la bontà delle previsioni espone in seguito sono stati utilizzati alcuni indici caratteristici. In particolare si è fatto riferimento a:

- **Coefficiente di determinazione o R^2 :** è una proporzione tra la variabilità dei dati e la correttezza del modello statistico utilizzato. Esso misura la frazione della varianza della variabile dipendente espressa dalla regressione. È calcolato come:

$$R^2 = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (16)$$

Dove: y_i sono i dati osservati, \bar{y} è la media dei dati osservati e \hat{y} sono i dati stimati dal modello.

- **Errore medio assoluto o MAE:** è la misura della differenza fra due variabili. È calcolato come:

$$MAE = \frac{\sum_{i=1}^n |\hat{y}_i - y_i|}{n} \quad (17)$$

Dove: y_i sono i dati osservati e \hat{y}_i sono i dati stimati dal modello.

3.3 Industria Tessile

3.3.1 Gruppo di cogenerazione

Il primo stabilimento preso in oggetto è un lanificio, questa tipologia di stabilimento produttivo utilizza grandi quantità di acqua calda e vapore nei processi di lavorazione, rendendo particolarmente vantaggioso l'utilizzo di un impianto di cogenerazione. Il fabbisogno di energia elettrica non coperto dal cogeneratore è integrato prelevando energia dalla rete elettrica nazionale.

Il tessuto subisce differenti lavorazioni, a seconda del prodotto finale desiderato, nei diversi reparti dello stabilimento. Tra le operazioni principali vi sono il finissaggio (a umido o a secco), la tintura e la garzatura.

Le operazioni di finissaggio sono molteplici e si svolgono con ripetuti passaggi dei tessuti sia nel reparto a secco che in quello a umido. Si ha innanzi tutto un rammendo (se necessario) delle pezze grezze provenienti dalla tessitura che vengono poi avviate ai cucì sacco, ai lavaggi e alla follatura, nonché ai lavaggi in continuo e infine agli apri corda. Successivamente i tessuti sono asciugati nella rameuse. Vi è poi (eventualmente) una nuova fase di rammendo e rammendo fine; altre operazioni sono la cimatura e il decatissaggio, la calandratura; ulteriore operazione per conferire particolari proprietà al tessuto è il vaporizzo.

Il gruppo di cogenerazione è illustrato in Figura 35 e Figura 36.



Figura 35: gruppo di cogenerazione dall'esterno



Figura 36: gruppo di cogenerazione, vista posteriore

Gli elementi visibili dall'esterno sono:

- 1) container esterno che funge da protezione e per il contenimento del rumore;
- 2) silenziatore per i fumi di scarico;
- 3) dissipatore di emergenza, utilizzato nel caso in cui l'utenza non necessiti dell'energia termica;
- 4) sistema di immissione dell'aria;
- 5) sala di controllo;
- 6) caldaia per il recupero dei fumi;
- 7) by-pass dei fumi di scarico;
- 8) camino per i gas di scarico.

Il modulo cogenerativo fornisce una potenza elettrica a pieno carico pari a 900 kW_{el} e una potenza termica tramite cogenerazione di 980 kW_{th}.

Il modulo cogenerativo ha un primo stadio di recupero termico all'interno del gruppo motore (attraverso il circuito dell'olio, il circuito acqua delle camicie motore e il primo stadio di interrefrigerazione) che consente un recupero di 580 kW_{th} in forma di acqua calda. Uno scambiatore di calore successivo recupera la potenza termica proveniente dal circuito di raffreddamento del motore e 70 kW_{th} della batteria di preriscaldamento installata nel circuito per il recupero dei fumi che produce acqua calda a circa 80 °C.

I fumi di combustione del motore passano attraverso uno scambiatore per il recupero fumi in grado di recuperare 335 kW_{th}, dei quali 56 kW_{th} sotto forma di vapore saturo a 188,7 °C interamente autoconsumato dallo stabilimento nel ciclo di produzione.

Lo stabilimento dal calore recuperato produce:

- Acqua calda a circa 80 °C;
- Vapore saturo secco a 11,2 bar e 188,7 °C;

La restante energia termica non viene recuperata poiché è a bassa temperatura, circa 43 °C, e viene dissipata tramite uno scambiatore installato sul tetto.

Tabella 5: dati di targa dell'impianto di cogenerazione

PCI metano	9,5	kWh/Nm ³
Consumo di gas	228	Nm ³ /h
Potenza in ingresso	2,170	kW
Potenza elettrica erogata	900	kW
Potenza termica recuperabile	980	kW
Rendimento elettrico	41.5	%
Rendimento termico	45.2	%
Rendimento complessivo	86.7	%

Con i dati utilizzati nello studio sono stati valutati gli indici di prestazione relativi alla cogenerazione (vedi 0) e riportati nella Tabella 6: indici mensili relativi al primo impianto di cogenerazione, contenente la stima del risparmio durante un anno di esercizio (si veda la normativa in allegato).

Tabella 6: indici mensili relativi al primo impianto di cogenerazione

PES	RISP [MWh/anno]	Certificati Bianchi annuali
0,19	1400	173

Ne risulta che lo stabilimento rientra nella categoria CAR (Cogenerazione ad Alto Rendimento).

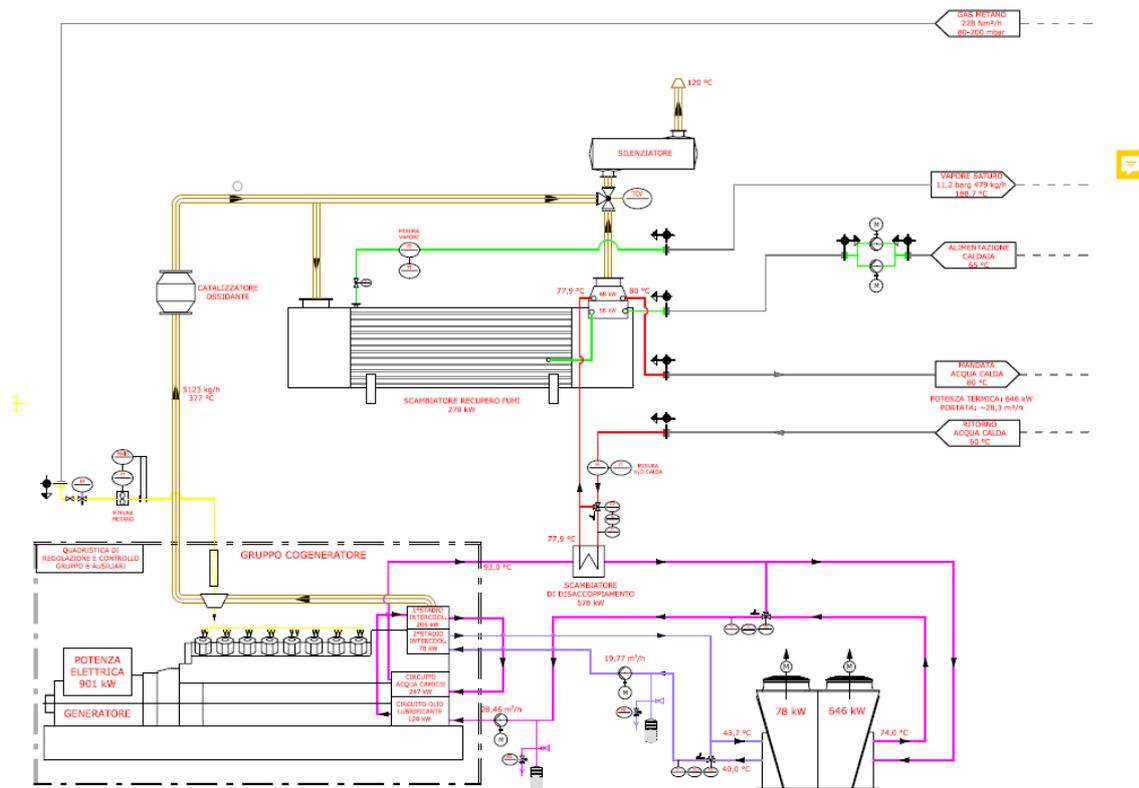


Figura 37: schema impianto di cogenerazione

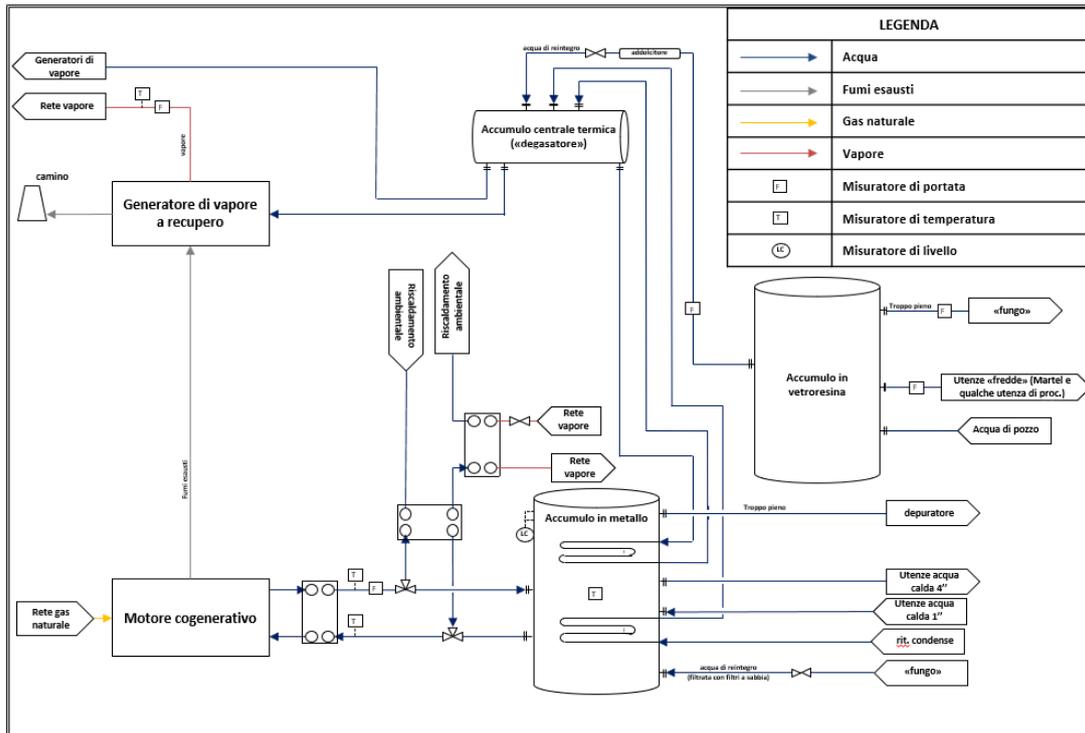


Figura 38: schema semplificato dell'impianto di cogenerazione

3.3.2 Apparecchiature di monitoraggio

L'energia elettrica autoprodotta nell'impianto di cogenerazione viene contabilizzata tramite un apposito gruppo di misura fiscale installato subito a valle dell'alternatore. Il gruppo misura l'energia elettrica attiva di tipo trifase statico.

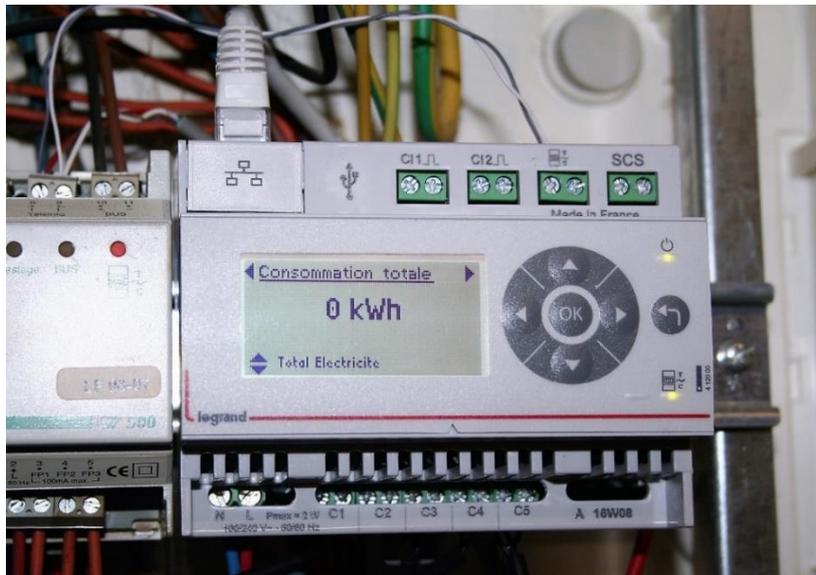


Figura 39: strumento di misura dell'energia elettrica prodotta

L'energia elettrica prodotta, al netto della quota parte utilizzata per l'alimentazione delle apparecchiature ausiliarie del gruppo di generazione degli assorbimenti della centrale e le perdite dovute alla rete e alla trasformazione delle tensioni, è interamente usata per soddisfare il fabbisogno energetico dello stabilimento.

Eventuali picchi di richiesta non soddisfabili dall'impianto di cogenerazione sono soddisfatti prelevando energia dalla rete elettrica nazionale

3.3.3 Preparazione e formattazione dei dati

I parametri di funzionamento caratteristici dell'impianto sono acquisiti attraverso una rete di monitoraggio attraverso delle apparecchiature SCADA ed inviati ad una piattaforma cloud di stoccaggio. Le misurazioni sono disponibili in un file .csv in cui è registrati, per ogni intervallo temporale la potenza prodotta dall'impianto di cogenerazione dello stabilimento (termica ed elettrica), la potenza assorbita dalla rete e i consumi di combustibile.

Timestamp	P CHP [kW]	P RETE [kW]	P VAPORE [kW]	P H2O [kW]	GAS NATURALE [Smc]
18/09/2018 11:00:00	624,81	250,47	0	332,83	
18/09/2018 12:00:00	449,91	331,61	0	226,98	0
18/09/2018 13:00:00	520,65	197	44,89	298,48	0
18/09/2018 14:00:00	576,45	130,99	240,79	340,12	0
18/09/2018 15:00:00	614,46	195,28	262,27	351,32	0
18/09/2018 16:00:00	691,58	147,51	206,76	408,55	0
18/09/2018 17:00:00	690,34	157,11	259,21	404,23	0
18/09/2018 18:00:00	668,06	128,48	235,16	387,7	0
18/09/2018 19:00:00	614,75	139,49	276,69	350,42	0
18/09/2018 20:00:00	541,37	124,92	245,07	301,62	0
18/09/2018 21:00:00	489,65	146,87	235,38	262,14	0
18/09/2018 22:00:00	402,16	559,08	256,54	211,49	0
18/09/2018 23:00:00	0	431,15	0	0	0
19/09/2018 00:00:00	0	463,94	0	0	0
19/09/2018 01:00:00	0	451,93	0	0	0

Figura 40: struttura originale dei dati acquisiti

Il formato .csv viene utilizzato poiché è molto leggero e consente una lettura più veloce da parte del programma rispetto ad un file .xlsx (file in formato Excel).

Il primo e più importante passo nella scrittura del programma consiste nell'organizzare opportunamente i dati a disposizione da far visionare alla rete per il processo di addestramento poiché senza una corretta formattazione la rete non sarà in grado di analizzare correttamente gli input che le vengono forniti. Particolarmente importante è la normalizzazione delle grandezze utilizzate, poiché, sebbene la rete sia in grado di analizzare i dati anche con le loro grandezze "reali", applicando una normalizzazione la lettura risulta molto più veloce. In questo studio si è applicata una normalizzazione sul valore medio delle grandezze.

3.3.4 Rete FF per la previsione oraria

Creazione della rete

Una volta che i dati sono stati preparati adeguatamente si può procedere alla realizzazione della struttura della rete. La prima tipologia di rete studiata è una rete Feed Forward, la tipologia più semplice di rete ma che risulta performante in una grande varietà di applicazioni.

Il primo modello ha l'obiettivo di prevedere la potenza elettrica di un'ora basandosi sulle 24 ore precedenti. La rete può essere schematizzata come due blocchi separati composti dagli strati nascosti e dallo strato finali di output, i quali ricevono l'input, lo elaborano in base ai pesi delle connessioni e alla rispettiva funzione di attivazione dello strato. Fornendo quindi le potenze precedenti reali $P(t)$ il modello restituisce la potenza prevista per l'ora successiva $P'(t+1)$.

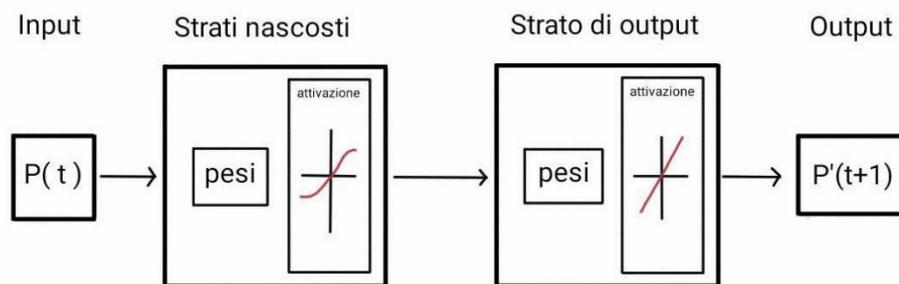


Figura 41: funzionamento della rete FF per le previsioni orarie

La sintassi per la creazione della struttura della rete è molto compatta. Il numero di nodi che compongono gli strati nascosti della rete può essere agevolmente per testare varie configurazioni e trovare quella più performante in base allo studio effettuato, mentre il numero di neuroni dello strato di output deve necessariamente essere pari al numero di output che si vuole ottenere, in questo caso un singolo valore.

```
#=====
model = keras.Sequential([
    keras.layers.Dense(25, activation=tf.nn.relu,
                        input_shape=(train_x.shape[1],)),
    keras.layers.Dropout(0.2)
    keras.layers.Dense(12, activation=tf.nn.relu),
    keras.layers.Dropout(0.2)
    keras.layers.Dense(1)
])

model.compile(loss='mse',
              optimizer='Adam',
              metrics=['mae'])
#=====
```

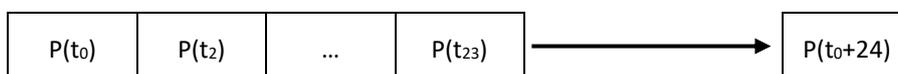
Si sono utilizzati due strati densi nascosti e uno di output, alla fine di ogni strato denso è stato applicato un layer con funzione dropout per prevenire fenomeni di overfitting. Gli strati nascosti utilizzano la funzione di attivazione *relu* (vedi 2.2.1). Successivamente si è definita la funzione per compilare la rete definendo i criteri di addestramento e valutazione. La funzione obiettivo utilizzata è l'errore quadratico medio (Mean Squared Error-MSE), poiché nelle previsioni regressive è un parametro significativo, ottimizzato attraverso l'algoritmo AdamOptimizer. La funzione metrica è utilizzata per visualizzare la progressione del processo di apprendimento è l'errore medio assoluto (Mean Absolute Error-MAE).

La struttura della rete è riassunta nella Tabella 7.

Tabella 7: capisaldi rete FF oraria

Rete	Layer	Tipo di layer	N° di nodi
Feed Forward	Nascosto	Denso	25
	Nascosto	Denso	12
	Output	Denso	1

La finestra di dati utilizzata comprende le 24 ore precedenti al valore calcolato, come schematizzato di seguito.



```

#=====
early_stop = keras.callbacks.EarlyStopping(monitor='val_loss', patience=5)

EPOCHS=200
BATCH_SIZE=24
history = model.fit(train_x, train_y,
                    batch_size=BATCH_SIZE, epochs=EPOCHS, validation_split=0.1,
                    callbacks=[early_stop], verbose=1)
#=====

```

In fine sono state definite le modalità di addestramento, in questo caso è stato impostato un valore limite di epoche molto alto, ma grazie all'implementazione di una funzione di Early Stopping il modello è in grado di fermare l'apprendimento nel momento in cui non si otterrebbero ulteriori miglioramenti, evitando di incorrere in fenomeni di overfitting.

Organizzazione dei dati

Il file .csv contenente le misurazioni viene caricato e le celle con data e ora vengono convertite in un formato leggibile dal programma. Vengono eliminate le celle vuote e si crea una nuova colonna in cui le potenze sono sfalsate di un'ora. I valori di questa colonna saranno utilizzati come labels durante la fase di addestramento. Successivamente si definiscono le funzioni necessarie per normalizzare i dati relativi alle potenze elettriche e per la creazione degli input e delle etichette da utilizzare nel training set e nel test set.

Si è scelto di suddividere il dataset in due sottoinsiemi:

- Il **Training Set**: comprende il 95% dei dati e viene utilizzato per addestrare il modello. A sua volta sarà suddiviso automaticamente da programma durante la fase di addestramento nel Training Subset e nel Validation Subset (VEDI) rispettivamente 10% e 85% del totale;
- Il **Test Set**: è composto dal 5% dei dati e viene utilizzato per verificare la bontà del modello testandolo su dai dati che non ha mai visto in precedenza.

A questo scopo si è utilizzato l'indice del dataset generato automaticamente per creare i due sottoinsiemi ed a entrambi si è applicata la funzione per pre-processare i dati.

Risultati e prestazioni

Sono state provate varie configurazioni della rete variando il numero di nodi che compongono gli strati nascosti per valutare la configurazione più performante. In tutti i casi lo script è impostato per fermare automaticamente l'apprendimento una volta che la funzione Loss selezionata (in questo caso si è adoperato l'errore quadratico medio) non migliori dopo 5 epoche. I risultati relativi alle reti Feed Forward sono illustrati nella Tabella 8. Data la natura stocastica del funzionamento delle reti neurali, per ogni configurazione è stato ripetuto il processo di training varie volte ottenendo risultati leggermente diversi ma con uno scostamento dal valore medio non significativo.

Tabella 8: prestazioni dell'addestramento rete FF oraria

Struttura	Training		Validation		R ²	MAE [kW]	epoche
	MSE	MAE	MSE	MAE			
10-4	0.0676	0.172	0.0496	0.157	0.882	66.3	84
	0.0689	0.167	0.0544	0.157	0.882	66.4	86
	0.0624	0.168	0.0489	0.159	0.892	66.3	81
	0.0655	0.170	0.0497	0.158	0.882	65.8	78
25-12	0.0645	0.171	0.0500	0.158	0.937	50.3	56
	0.0677	0.170	0.0514	0.159	0.932	54.0	55
	0.0598	0.169	0.0503	0.158	0.933	53.7	54
	0.0636	0.170	0.0507	0.159	0.933	53.5	57
50-25	0.0676	0.171	0.0484	0.162	0.922	57.8	42
	0.0689	0.173	0.0487	0.161	0.930	59.3	40
	0.0624	0.172	0.0479	0.162	0.931	55.6	42
	0.0655	0.172	0.0484	0.162	0.930	55.4	41
100-50	0.0676	0.172	0.0502	0.166	0.932	53.3	29
	0.0689	0.172	0.0499	0.166	0.931	56.1	27
	0.0624	0.172	0.0478	0.166	0.931	54.2	27
	0.0655	0.173	0.0511	0.165	0.931	55.8	30
Ora Precedente	-	-	-	-	0.721	78.8	-

Si noti come le prestazioni delle varie reti siano molto simili fra loro, l'unico elemento di differenza è, come prevedibile, il numero di epoche: una rete più semplice impiegherà un tempo maggiore per imparare i pattern caratteristici del dataframe. Le differenze rispetto gli altri indici caratteristici (errore medio assoluto (MAE) sul training set e sul validation set, R² e l'errore medio sui valori denormalizzati) sono trascurabili e assumono tutti ottimi risultati. Per questa analisi un buon compromesso risulta essere la configurazione a tre strati rispettivamente composti da 25,12 e 1 neurone. Si ricorda che il numero di neuroni dello strato finale deve essere uguale al numero di output attesi, in questo caso solamente uno essendo la previsione su di una singola ora. I risultati della previsione del modello sono anche stati confrontati rispetto al profilo di carico dell'ora precedente traslata di una singola ora rispetto a quella analizzata per valutare la reale necessità di un modello di previsione dei carichi, infatti, in alcuni casi, l'andamento potrebbe risultare così costante nel tempo da non richiederne uno. Tuttavia, in questo caso, il semplice scostamento di una singola ora risulta insoddisfacente poiché viene sballata la previsione tutti i picchi di produzione che si presentano nella giornata, risultando, di fatto, inutile.

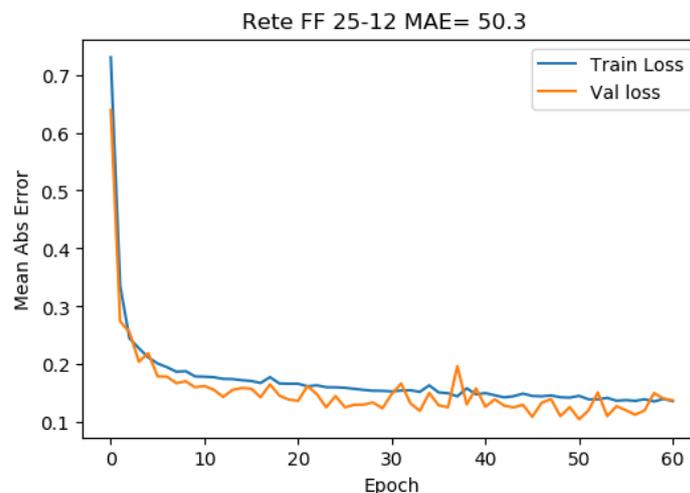


Figura 42: andamento dell'addestramento

La progressione dell'apprendimento del modello durante la fase di training è mostrata nella Figura 42, si osserva l'andamento iperbolico dell'errore in funzione delle epoche. L'errore del Training Set e del Validation Set mantengono valori molto simili durante tutto il processo e alla fine tendono a stabilizzarsi sullo stesso risultato. I dati utilizzati per l'addestramento compongono il 95% del totale (Training Set), il restante 5% è stato utilizzato per effettuare le previsioni e compararle con i valori reali (Test Set).

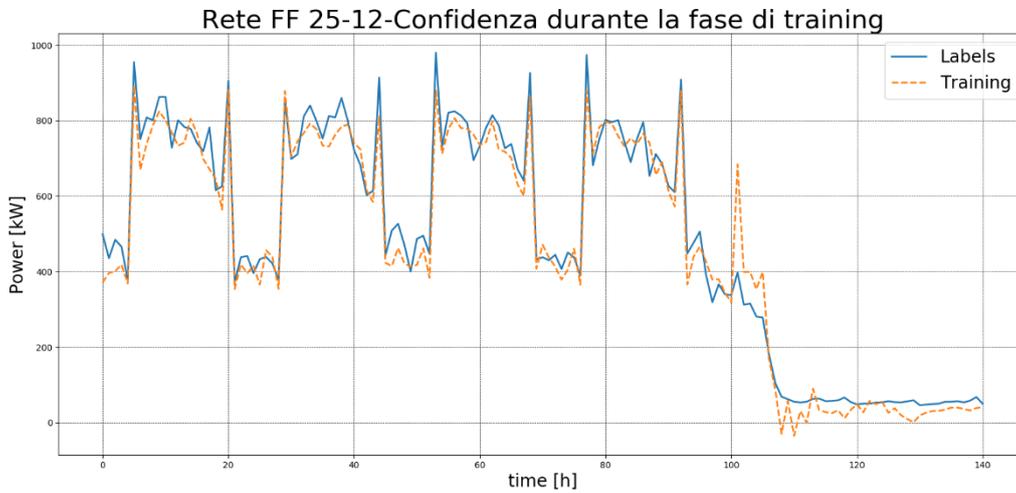


Figura 44: confidenza durante la fase di training

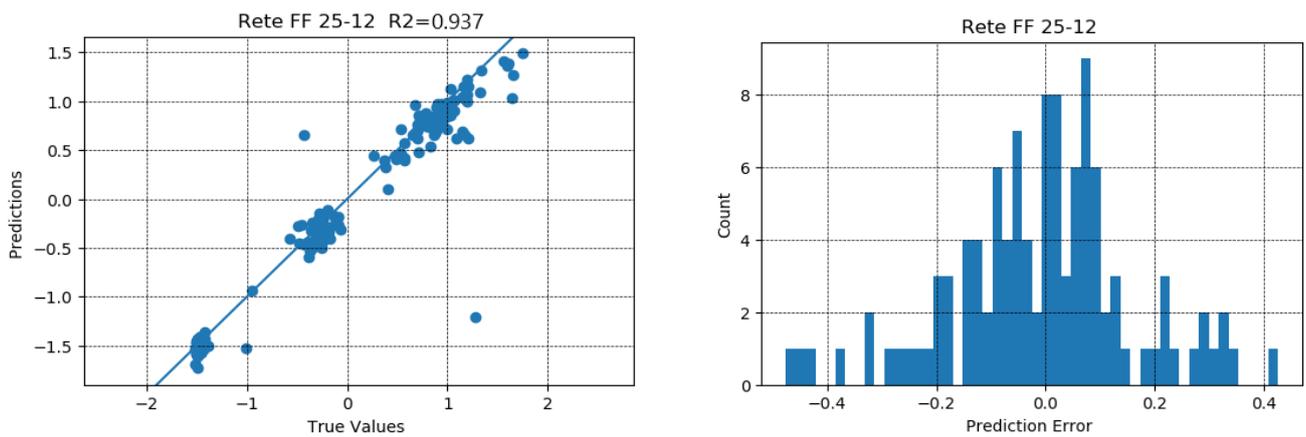


Figura 43: distribuzione errore

Dalla Figura 43 si osserva come le previsioni della rete abbiano un errore contenuto rispetto ai valori reali e con una distribuzione che tende alla distribuzione normale.

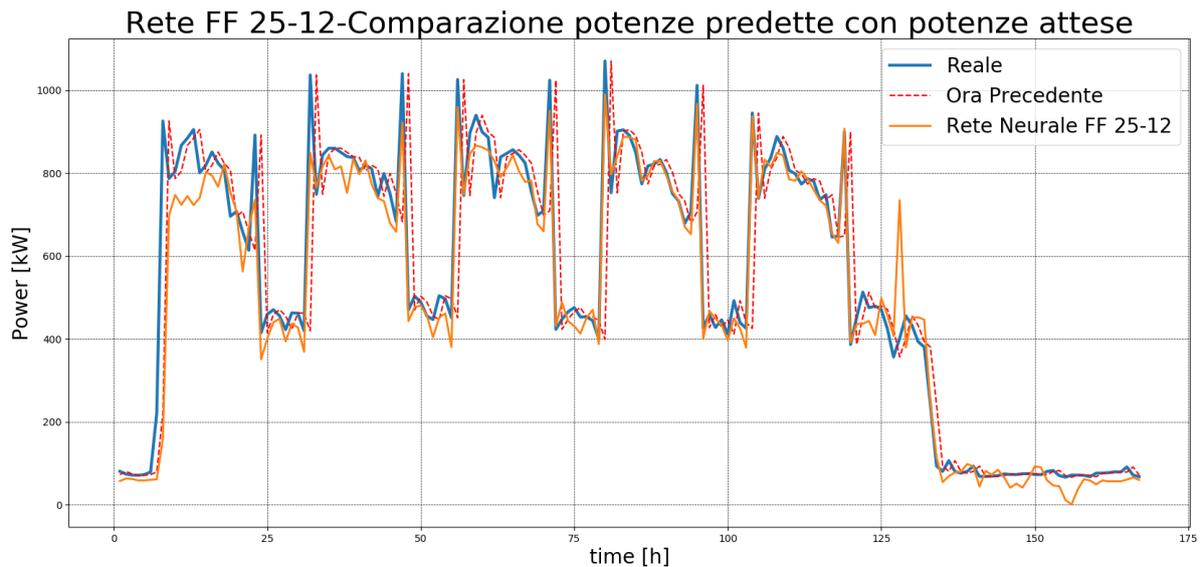


Figura 45: comparazione potenze previste con le potenze reali, rete FF oraria

La Figura 45 mostra l'andamento dei consumi predetti dalla rete neurale rispetto ai consumi reali nell'arco di una settimana. Si osserva come la previsione tenda ad avere una varianza di valori inferiore rispetto al caso reale e abbia un andamento più costante; questo comportamento è dovuto al fatto che i profili di consumo giornaliero sono sempre lievemente differente gli uni dagli altri e quindi il modello di previsione deve tenere conto di una enorme variabilità di possibili valori, di conseguenza tende a fornire un valore medio rispetto alle precedenti osservazioni. Lo scostamento medio fra la previsione e il valore reale si mantiene solitamente nell'ordine di qualche decina di kW_{el} ed eventuali picchi di assorbimento sono soddisfatti dalla rete elettrica nazionale, permettendo al cogeneratore di lavorare il più possibile alle condizioni nominali.

3.3.5 Rete FF per la previsione giornaliera

Creazione della rete

Una volta che i dati sono stati preparati adeguatamente si può procedere alla realizzazione della struttura della rete. Il secondo modello ha l'obiettivo di prevedere la potenza elettrica di un'intera giornata basandosi sulle misurazioni della giornata precedente. La rete può essere schematizzata come due blocchi separati composti dagli strati nascosti e dallo strato finali di output, i quali ricevono l'input, lo elaborano in base ai pesi delle connessioni e alla rispettiva funzione di attivazione dello strato. Fornendo quindi tutte le potenze nell'arco di una giornata $P_i(t)$ il modello restituisce la potenza prevista per la stessa ora nella giornata successiva $P_i'(t+24)$.

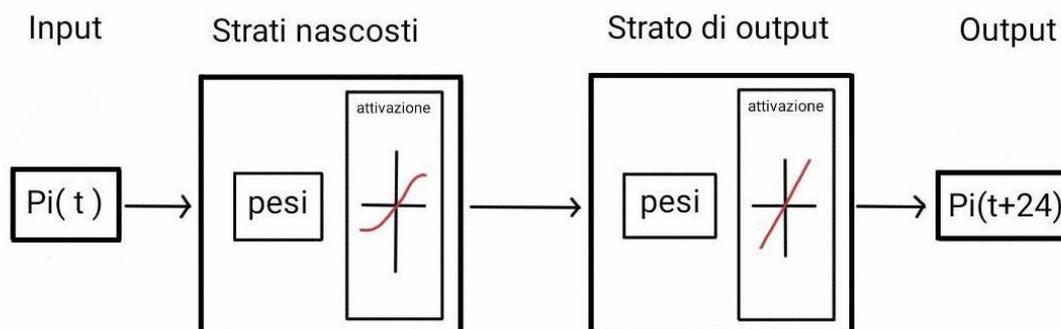


Figura 46: funzionamento della rete FF per le previsioni giornaliere

La sintassi per la creazione della struttura della rete è molto compatta. Il numero di nodi che compongono gli strati nascosti della rete può essere agevolmente per testare varie configurazioni e trovare quella più

performante in base allo studio effettuato, mentre il numero di neuroni dello strato di output deve necessariamente essere pari al numero di output che si vuole ottenere, in questo caso 24 valori.

```

#=====
model = keras.Sequential([
    keras.layers.Dense(70, activation=tf.nn.relu,
                        input_shape=(train_x.shape[1],)),
    keras.layers.Dropout(0.2)
    keras.layers.Dense(50, activation=tf.nn.relu),
    keras.layers.Dropout(0.2)
    keras.layers.Dense(24)
])

model.compile(loss='mse',
              optimizer='Adam',
              metrics=['mae'])
#=====

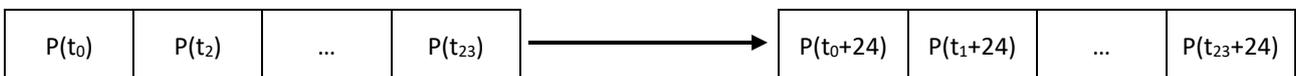
```

La struttura della rete è riassunta nella Tabella 11.

Tabella 9: capisaldi rete FF giornaliera

Rete	Layer	Tipo di layer	N° di nodi
Feed Forward	Nascosto	Denso	70
	Nascosto	Denso	50
	Output	Denso	24

Il modello utilizza un set di 24 valori (tutte le ore di una giornata) e prevede simultaneamente 24 valori, corrispondenti alle 24 ore della giornata successiva, come schematizzato di seguito.



Si sono utilizzati due strati densi nascosti e uno di output, alla fine di ogni strato denso è stato applicato un layer con funzione dropout per prevenire fenomeni di overfitting. Gli strati nascosti utilizzano la funzione di attivazione *relu* (vedi 2.2.1). Successivamente si è definita la funzione per compilare la rete definendo i criteri di addestramento e valutazione. La funzione obiettivo utilizzata è l'errore quadratico medio (Mean Squared Error-MSE), poiché nelle previsioni regressive è un parametro significativo, ottimizzato attraverso l'algoritmo AdamOptimizer. La funzione metrica è utilizzata per visualizzare la progressione del processo di apprendimento è l'errore medio assoluto (Mean Absolute Error-MAE).

```

#=====
early_stop = keras.callbacks.EarlyStopping(monitor='val_loss', patience=5)

EPOCHS=200
BATCH_SIZE=24
history = model.fit(train_x, train_y,
                    batch_size=BATCH_SIZE, epochs=EPOCHS, validation_split=0.1,
                    callbacks=[early_stop], verbose=1)
#=====

```

In fine sono state definite le modalità di addestramento, in questo caso è stato impostato un valore limite di epoche molto alto, ma grazie all'implementazione di una funzione di Early Stopping il modello è in grado di fermare l'apprendimento nel momento in cui non si otterrebbero ulteriori miglioramenti, evitando di incorrere in fenomeni di overfitting.

Organizzazione dei dati

Per questo nuovo modello si utilizza lo stesso set di dati usato in precedenza, tuttavia le funzioni atte a preprocessare i dati variano leggermente.

Risultati e prestazioni

Sono state provate varie configurazioni della rete variando il numero di nodi che compongono gli strati nascosti per valutare la configurazione più performante. In tutti i casi lo script è impostato per fermare automaticamente l'apprendimento una volta che la funzione Loss selezionata (in questo caso si è adoperato l'errore quadratico medio) non migliora dopo 5 epoche. I risultati relativi alle reti Feed Forward sono illustrati nella Tabella 12. Data la natura stocastica del funzionamento delle reti neurali, per ogni configurazione è stato ripetuto il processo di training varie volte ottenendo risultati leggermente diversi ma con uno scostamento dal valore medio non significativo.

Tabella 10: prestazioni dell'addestramento rete FF giornaliera

Struttura	Training		Validation		R ²	MAE [kW]	epoche
	MSE	MAE	MSE	MAE			
35-30	0.395	0.482	0.638	0.640	0.623	145	61
	0.394	0.481	0.638	0.640	0.598	149	60
	0.397	0.480	0.637	0.641	0.613	146	64
	0.395	0.479	0.638	0.640	0.613	145	61
50-30	0.367	0.444	0.621	0.630	0.691	128	42
	0.370	0.448	0.621	0.632	0.689	130	42
	0.368	0.441	0.620	0.633	0.689	129	43
	0.367	0.443	0.621	0.631	0.689	132	42
70-50	0.350	0.442	0.486	0.565	0.691	129	44
	0.351	0.440	0.487	0.565	0.689	132	45
	0.350	0.442	0.485	0.564	0.689	131	42
	0.349	0.442	0.486	0.564	0.689	131	45
100-60	0.350	0.496	0.626	0.629	0.664	141	34
	0.354	0.497	0.626	0.629	0.664	140	34
	0.352	0.497	0.626	0.630	0.654	142	33
	0.349	0.495	0.626	0.628	0.654	141	34
Giorno Precedente	-	-	-	-	0.110	170.1	-

Anche in questo caso i risultati ottenuti variano molto poco al variare della complessità della rete, tranne che per le epoche necessarie per addestrare il modello. Rispetto al modello preliminare si osserva come la precisione sia lievemente inferiore pur restando molto buona, in particolare R² si mantiene su un valore di circa 0.70. Le prestazioni risultano inferiori rispetto al modello preliminare, ma bisogna tenere conto che, in questo caso la previsione è molto più complessa. Per questa analisi un buon compromesso risulta essere la configurazione a tre strati rispettivamente composti da 70,50 e 24 neuroni. I risultati della previsione del modello sono anche stati confrontati rispetto al profilo di carico dell'ora precedente traslata di una singola ora rispetto a quella analizzata per valutare la reale necessità di un modello di previsione dei carichi, infatti, in alcuni casi, l'andamento potrebbe risultare così costante nel tempo da non richiederne uno.

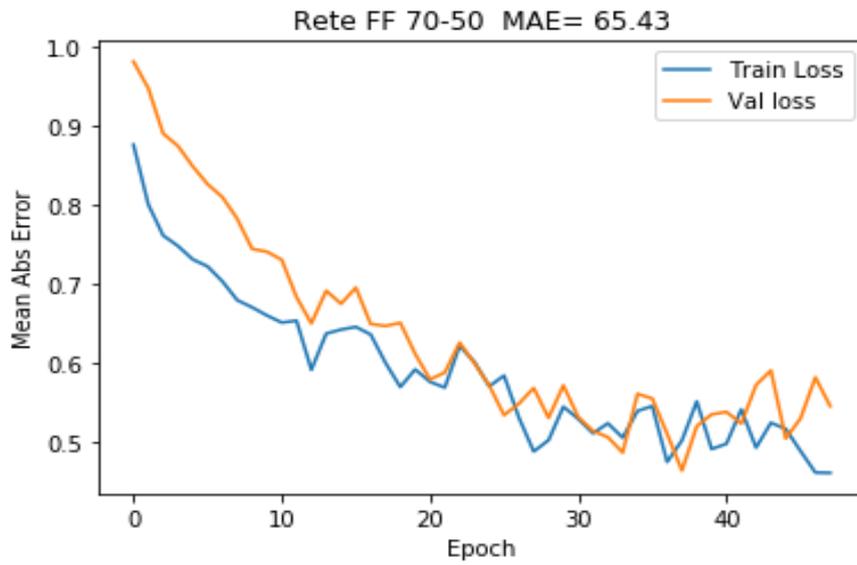


Figura 47: andamento dell'errore durante la fase di training

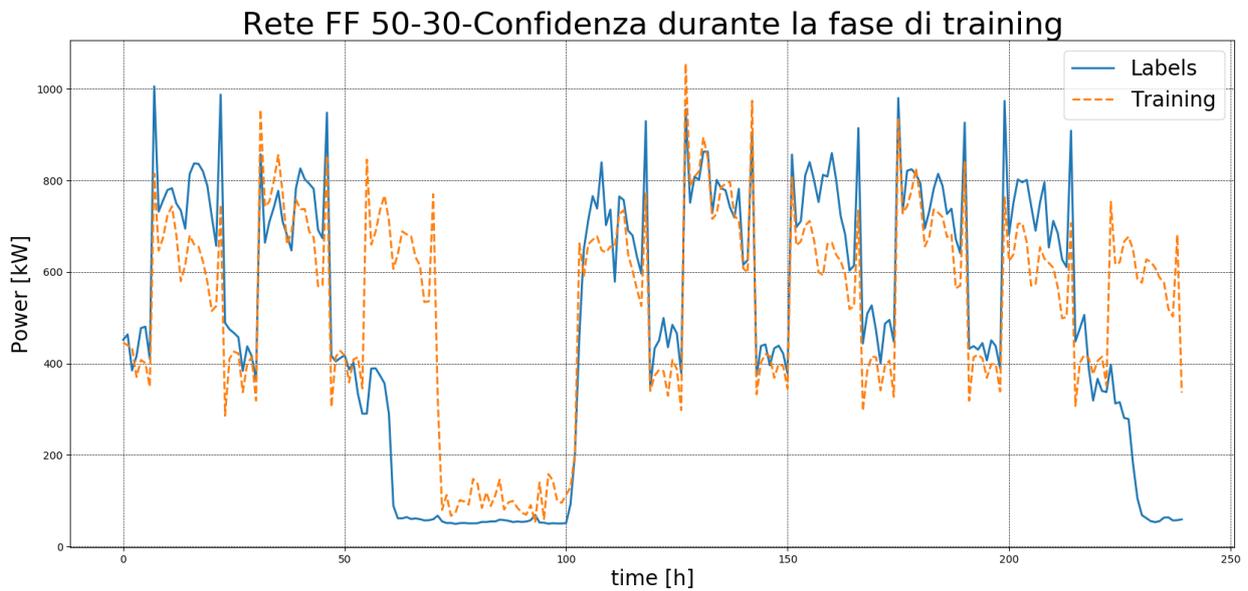


Figura 49: confidenza durante la fase di training

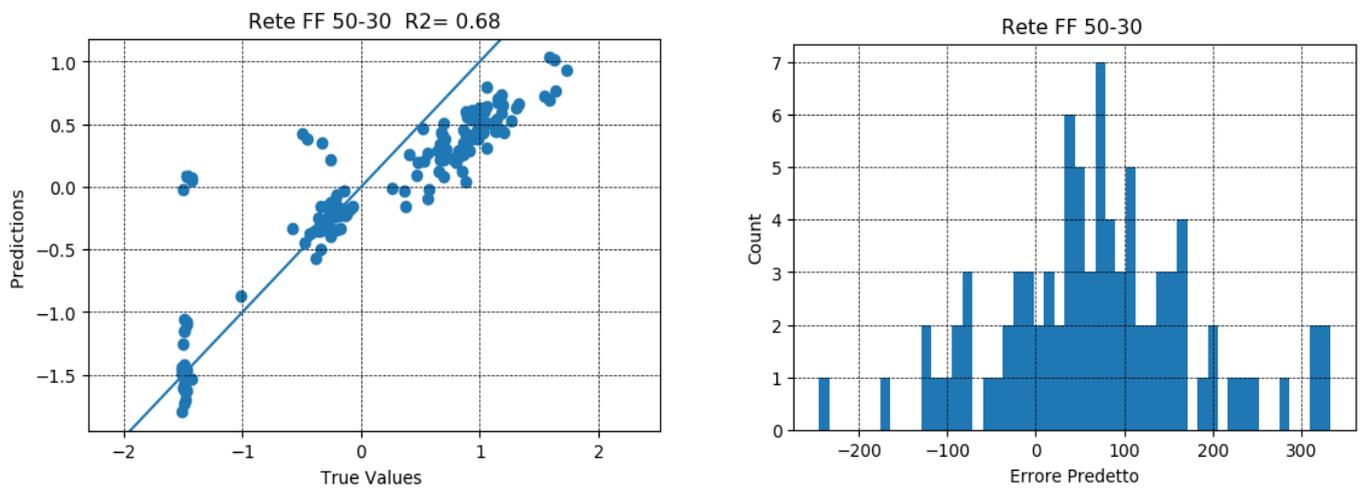


Figura 48: distribuzione errore

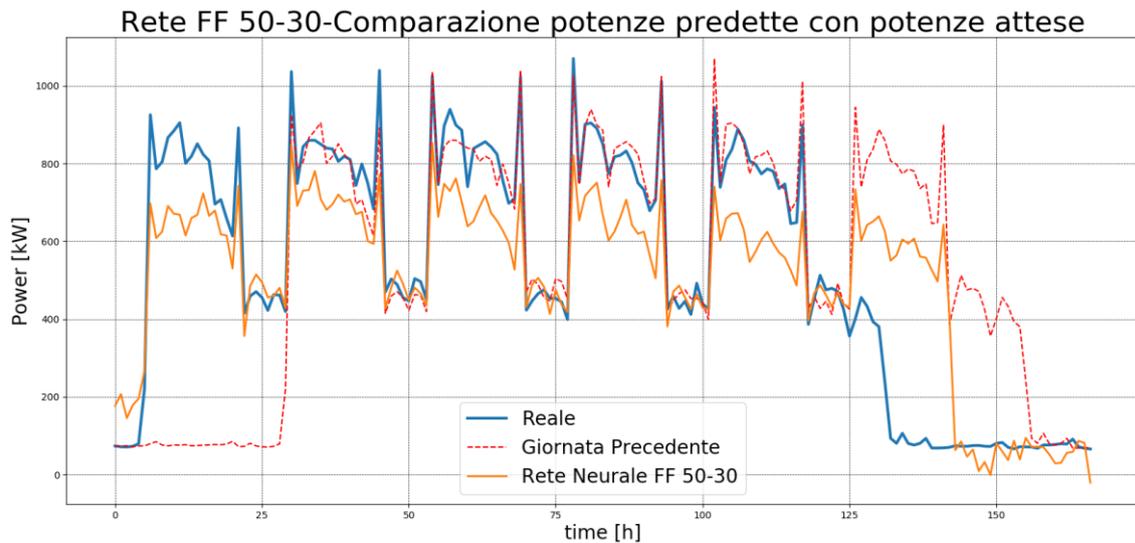


Figura 50: comparazione potenze previste con le potenze reali, rete FF giornaliera

Come si osserva nella Figura 50 il modello gestisce in maniera soddisfacente i giorni feriali, mentre non è in grado di riconoscere il sabato e la domenica, nei quali i consumi sono ridotti. Si è dunque proceduto con la creazione di un modello in grado di prevedere i consumi tenendo conto delle giornate lavorative e non.

3.3.6 Rete FF per la previsione settimanale

Creazione della rete

Una volta che i dati sono stati preparati adeguatamente si può procedere alla realizzazione della struttura della rete. Il secondo modello ha l'obiettivo di prevedere la potenza elettrica di un'intera settimana basandosi sulle misurazioni della settimana precedente. La rete può essere schematizzata come due blocchi separati composti dagli strati nascosti e dallo strato finali di output, i quali ricevono l'input, lo elaborano in base ai pesi delle connessioni e alla rispettiva funzione di attivazione dello strato. Fornendo quindi tutte le potenze nell'arco di una settimana $P_i(t)$ il modello restituisce la potenza prevista per la stessa ora nella settimana successiva $P_i(t+168)$.

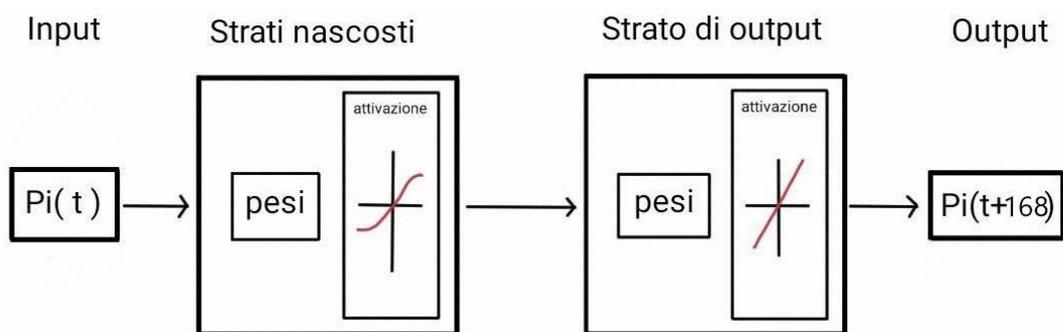


Figura 51: funzionamento della rete FF per le previsioni settimanale

La sintassi per la creazione della struttura della rete è molto compatta. Il numero di nodi che compongono gli strati nascosti della rete può essere agevolmente per testare varie configurazioni e trovare quella più performante in base allo studio effettuato, mentre il numero di neuroni dello strato di output deve necessariamente essere pari al numero di output che si vuole ottenere, in questo caso 168 valori.

```
#=====
model = keras.Sequential([
    keras.layers.Dense(70, activation=tf.nn.relu,
                       input_shape=(train_x.shape[1],)),
    keras.layers.Dropout(0.2)
```

```

keras.layers.Dense(50, activation=tf.nn.relu),
keras.layers.Dropout(0.2)
keras.layers.Dense(168)
])

model.compile(loss='mse',
              optimizer='Adam',
              metrics=['mae'])
#=====

```

La struttura della rete è riassunta nella Tabella 11.

Tabella 11: capisaldi rete FF giornaliera

Rete	Layer	Tipo di layer	N° di nodi
Feed Forward	Nascosto	Denso	70
	Nascosto	Denso	50
	Output	Denso	168

Il modello utilizza un set di 24 valori (tutte le ore di una giornata) e prevede simultaneamente 24 valori, corrispondenti alle 24 ore della giornata successiva, come schematizzato di seguito.



Si sono utilizzati due strati densi nascosti e uno di output, alla fine di ogni strato denso è stato applicato un layer con funzione dropout per prevenire fenomeni di overfitting. Gli strati nascosti utilizzano la funzione di attivazione *relu* (vedi 2.2.1). Successivamente si è definita la funzione per compilare la rete definendo i criteri di addestramento e valutazione. La funzione obiettivo utilizzata è l'errore quadratico medio (Mean Squared Error-MSE), poiché nelle previsioni regressive è un parametro significativo, ottimizzato attraverso l'algoritmo AdamOptimizer. La funzione metrica è utilizzata per visualizzare la progressione del processo di apprendimento è l'errore medio assoluto (Mean Absolute Error-MAE).

```

#=====
early_stop = keras.callbacks.EarlyStopping(monitor='val_loss', patience=5)

EPOCHS=200
BATCH_SIZE=24
history = model.fit(train_x, train_y,
                   batch_size=BATCH_SIZE, epochs=EPOCHS, validation_split=0.1,
                   callbacks=[early_stop], verbose=1)
#=====

```

In fine sono state definite le modalità di addestramento, in questo caso è stato impostato un valore limite di epoche molto alto, ma grazie all'implementazione di una funzione di Early Stopping il modello è in grado di fermare l'apprendimento nel momento in cui non si otterrebbero ulteriori miglioramenti, evitando di incorrere in fenomeni di overfitting.

Organizzazione dei dati

Per questo nuovo modello si utilizza lo stesso set di dati usato in precedenza, tuttavia le funzioni atte a preprocessare i dati variano leggermente. In questo caso, infatti, i dati in input e in output sono tutte e 168 le ore della settimana.

Risultati e prestazioni

Sono state provate varie configurazioni della rete variando il numero di nodi che compongono gli strati nascosti per valutare la configurazione più performante. In tutti i casi lo script è impostato per fermare automaticamente l'apprendimento una volta che la funzione Loss selezionata (in questo caso si è adoperato l'errore quadratico medio) non migliori dopo 5 epoche. I risultati relativi alle reti Feed Forward sono illustrati nella Tabella 12. Data la natura stocastica del funzionamento delle reti neurali, per ogni configurazione è stato ripetuto il processo di training varie volte ottenendo risultati leggermente diversi ma con uno scostamento dal valore medio non significativo.

Tabella 12: prestazioni dell'addestramento rete FF settimanale

Struttura	Training		Validation		R ²	MAE [kW]	epoche
	MSE	MAE	MSE	MAE			
35-30	0.395	0.482	0.638	0.640	0.821	78.6	51
	0.394	0.481	0.638	0.640	0.791	80.2	50
	0.397	0.480	0.637	0.641	0.819	77.9	54
	0.395	0.479	0.638	0.640	0.810	78.0	51
50-30	0.367	0.444	0.621	0.630	0.951	45.4	32
	0.370	0.448	0.621	0.632	0.951	46.3	32
	0.368	0.441	0.620	0.633	0.943	49.1	33
	0.367	0.443	0.621	0.631	0.952	47.5	32
70-50	0.350	0.442	0.486	0.565	0.943	50.0	34
	0.351	0.440	0.487	0.565	0.951	46.4	35
	0.350	0.442	0.485	0.564	0.951	45.9	32
	0.349	0.442	0.486	0.564	0.951	46.2	35
100-60	0.350	0.496	0.626	0.629	0.953	44.8	30
	0.354	0.497	0.626	0.629	0.964	43.4	31
	0.352	0.497	0.626	0.630	0.951	45.3	19
	0.349	0.495	0.626	0.628	0.951	46.7	30
Settimana Precedente	-	-	-	-	0.932	48.0	-

Anche in questo caso i risultati ottenuti variano molto poco al variare della complessità della rete, tranne che per le epoche necessarie per addestrare il modello. Rispetto al modello precedente si osserva come la precisione sia lievemente inferiore pur restando molto buona, in particolare R² si mantiene su un valore di circa 0.95. Le prestazioni risultano inferiori rispetto al modello preliminare, ma bisogna tenere conto che, in questo caso la previsione è molto più complessa. Per questa analisi un buon compromesso risulta essere la configurazione a tre strati rispettivamente composti da 70,50 e 168 neuroni.

Nella Figura 52 è rappresentato l'andamento dell'errore durante la fase di addestramento; si noti come sia per il Validation Set sia per il Training Set l'errore si mantenga simile durante tutto il processo e tenda a stabilizzarsi su un valore quasi coincidente negli step finali.

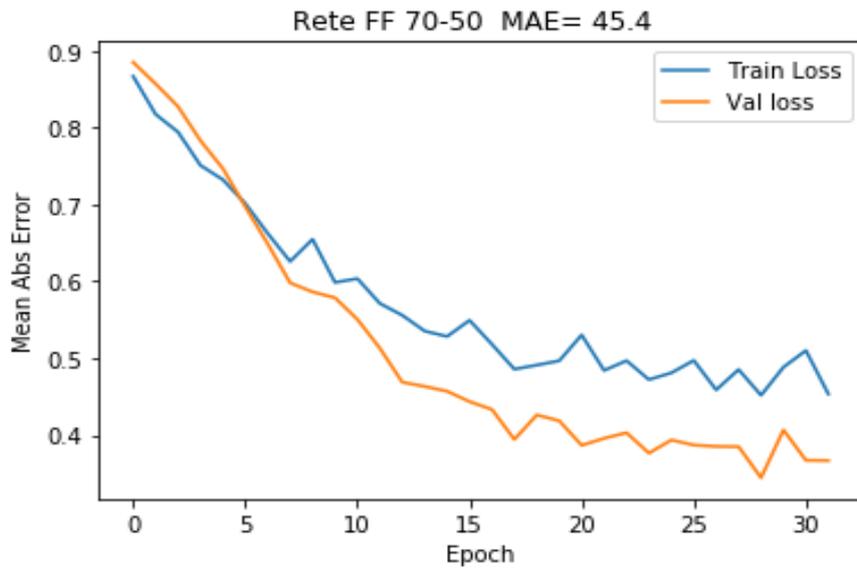


Figura 52: andamento dell'errore durante la fase di training

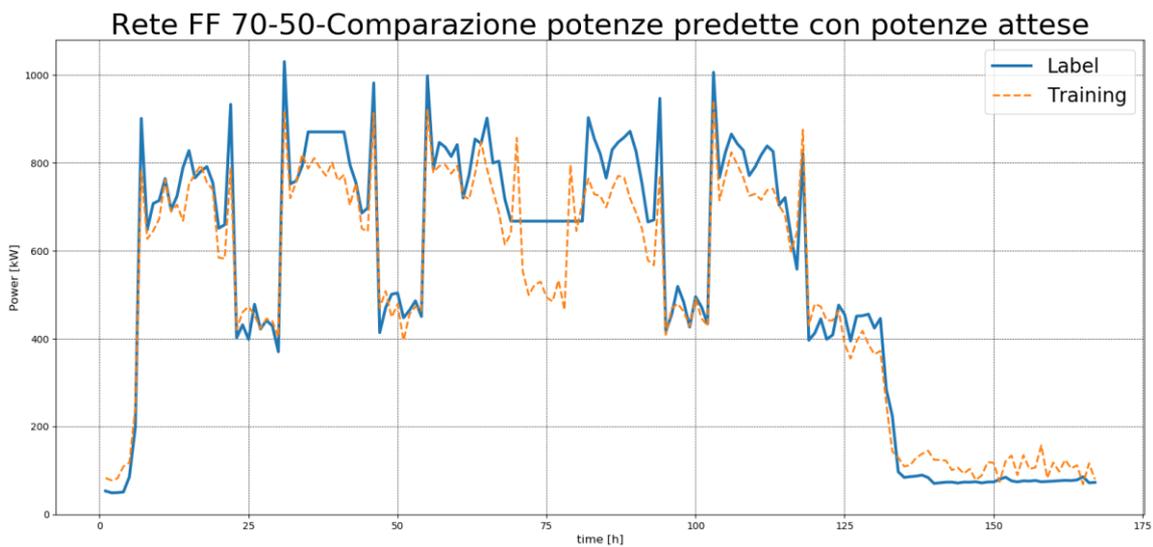


Figura 54: confidenza durante la fase di training

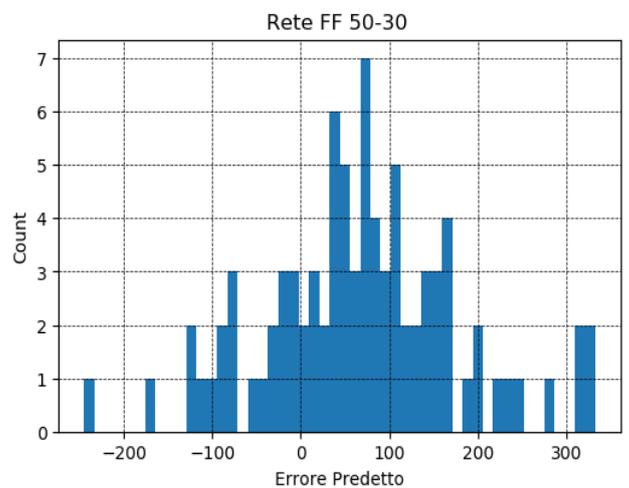
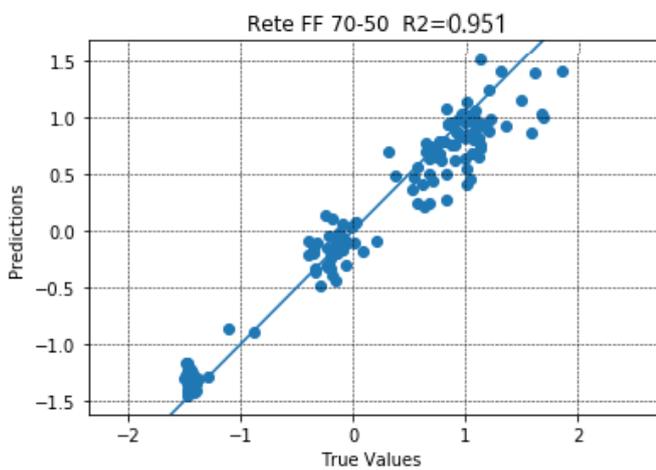


Figura 53: distribuzione errore

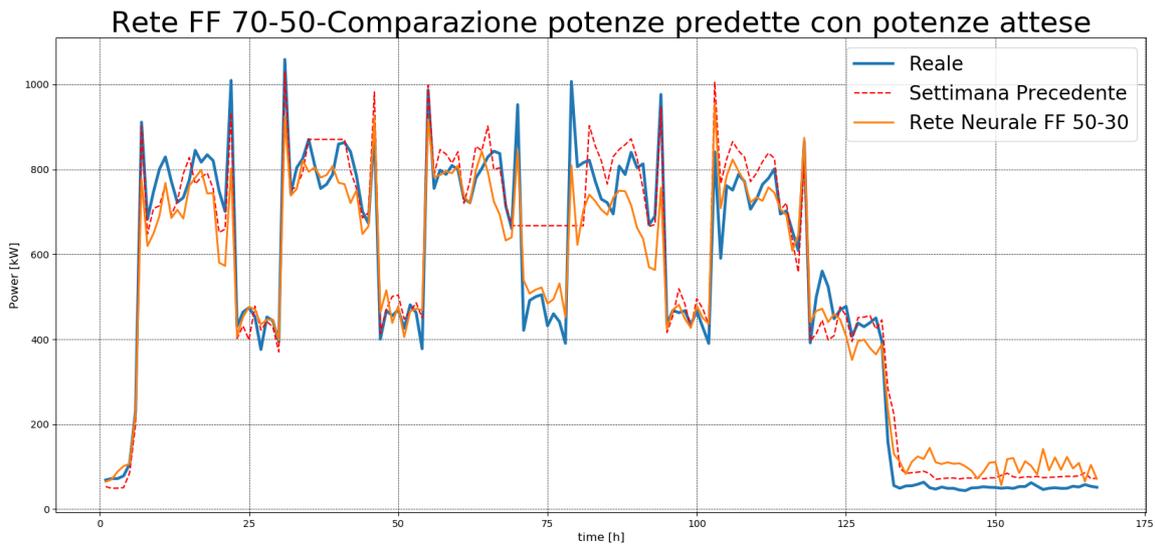
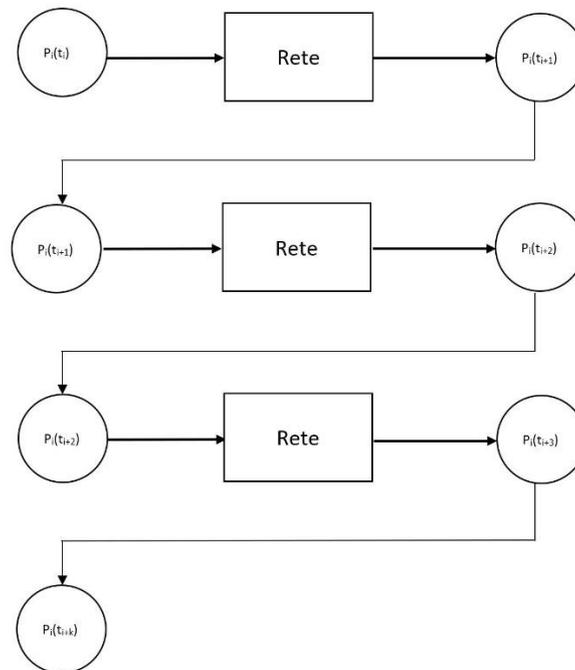


Figura 55: comparazione potenze previste con le potenze reali, rete FF settimanale

Osservando la comparazione fra la curva delle previsioni e quella dei dati reali nella Figura 55, si osserva come la previsione di un'intera settimana sia meno precisa rispetto a alla previsione di una singola ora, tuttavia il profilo di carico predetto segue in maniera fedele l'andamento reale distaccandosi mediamente di poche decine di kW_{el}.

3.3.7 Rete FF per la previsione settimanale con modello alternativo

Durante lo sviluppo del modello per la previsione giornaliera si è valutata la costruzione di un modello alternativo in grado di svolgere lo stesso compito. Questo modello effettua una previsione sull'arco di un'intera giornata in maniera simile al modello visto nel paragrafo 3.3.4 utilizzando valori di partenza (reali) solamente la prima previsione, le previsioni per le ore successive si basano sulle previsioni stesse, come mostrato nello schema sottostante.



Il risultato di ogni previsione viene utilizzato come base per la previsione dell'ora successiva.

Le prestazioni del modello si sono, tuttavia, dimostrate significativamente inferiori, poiché, sebbene l'errore del modello su di una singola previsione risulti contenuto (solitamente fra il 5% e 8%), ogni previsione è affetta anche dagli errori precedenti e questo fa sì che la previsione risulti inaffidabile dopo pochi step, poiché non riesce a prevedere efficacemente i picchi di richiesta, come mostrato in Figura 56.

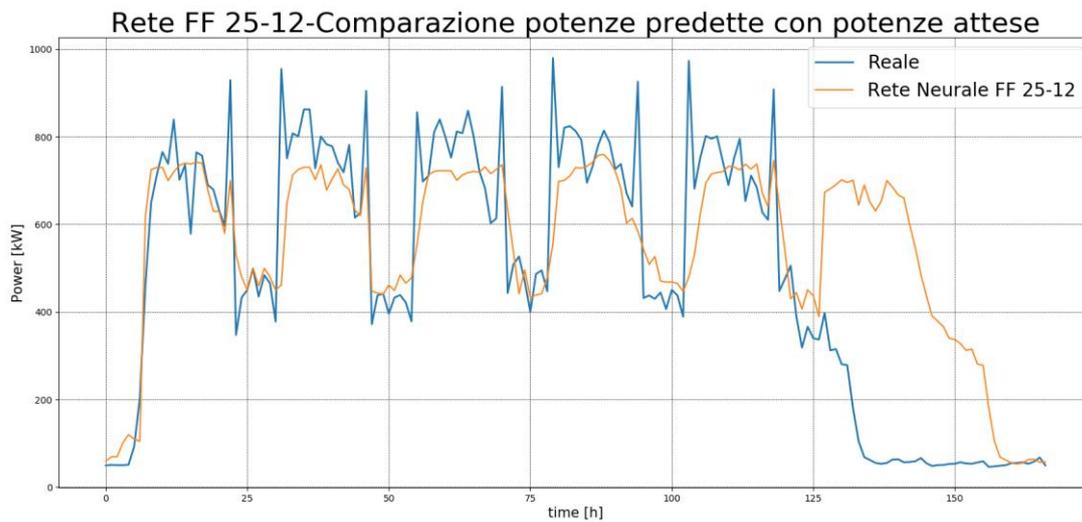


Figura 56: previsione della rete alternativa

3.3.8 Rete LSTM per la previsione oraria

Creazione della rete

Una volta che i dati sono stati preparati adeguatamente si può procedere alla realizzazione della struttura della rete. Il modello ha l'obiettivo di prevedere la potenza elettrica di un'ora basandosi sulle 24 ore precedenti. La rete può essere schematizzata come due blocchi separati composti dagli strati nascosti e dallo strato finale di output, i quali ricevono l'input, lo elaborano in base ai pesi delle connessioni e alla rispettiva funzione di attivazione dello strato. Fornendo quindi le potenze precedenti reali $P(t)$ il modello restituisce la potenza prevista per l'ora successiva $P'(t+1)$.

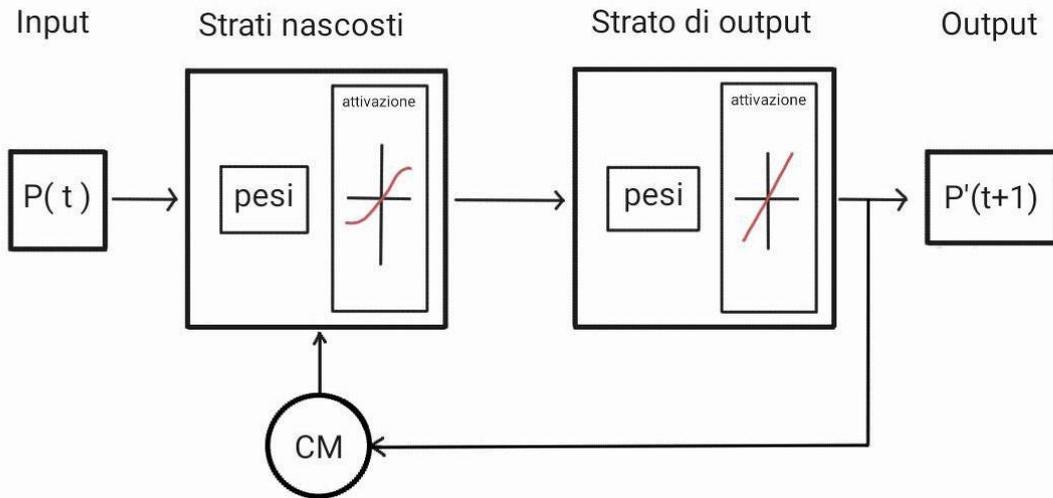


Figura 57: funzionamento della rete LSTM per la previsione oraria

La sintassi per la creazione della struttura della rete è molto compatta. Il numero di nodi che compongono gli strati nascosti della rete può essere agevolmente per testare varie configurazioni e trovare quella più performante in base allo studio effettuato, mentre il numero di neuroni dello strato di output deve necessariamente essere pari al numero di output che si vuole ottenere, in questo caso un solo valore.

```

=====
model = keras.Sequential([
    keras.layers.LSTM(25, activation=tf.nn.relu,
                      input_shape=(train_x.shape[1],)),
    keras.layers.Dropout(0.2)
    keras.layers.LSTM(12, activation=tf.nn.relu),
    keras.layers.Dropout(0.2)
    keras.layers.Dense(1) ])

model.compile(loss='mse',
              optimizer='Adam',
              metrics=['mae'])
=====

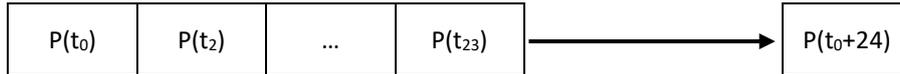
```

La struttura della rete è riassunta nella Tabella 13.

Tabella 13: capisaldi rete LSTM oraria

Rete	Layer	Tipo di layer	N° di nodi
Long short Term Memory	Nascosto	LSTM	25
	Nascosto	LSTM	12
	Output	Denso	1

La finestra di dati utilizzata comprende le 24 ore precedenti al valore calcolato, come schematizzato di seguito.



Si sono utilizzati due strati densi nascosti LSTM e uno di output di tipo Dense, alla fine di ogni strato è stato applicato un layer con funzione dropout per prevenire fenomeni di overfitting. Gli strati nascosti utilizzano la funzione di attivazione *relu* (vedi 2.2.1). Successivamente si è definita la funzione per compilare la rete definendo i criteri di addestramento e valutazione. La funzione obiettivo utilizzata è l'errore quadratico medio (Mean Squared Error-MSE), poiché nelle previsioni regressive è un parametro significativo, ottimizzato attraverso l'algoritmo AdamOptimizer. La funzione metrica è utilizzata per visualizzare la progressione del processo di apprendimento è l'errore medio assoluto (Mean Absolute Error-MAE).

```

#=====
early_stop = keras.callbacks.EarlyStopping(monitor='val_loss', patience=5)

EPOCHS=200
BATCH_SIZE=24
history = model.fit(train_x, train_y,
                    batch_size=BATCH_SIZE, epochs=EPOCHS, validation_split=0.1,
                    callbacks=[early_stop], verbose=1)
#=====

```

In fine sono state definite le modalità di addestramento, in questo caso è stato impostato un valore limite di epoche molto alto, ma grazie all'implementazione di una funzione di Early Stopping il modello è in grado di fermare l'apprendimento nel momento in cui non si otterrebbero ulteriori miglioramenti, evitando di incorrere in fenomeni di overfitting.

Organizzazione dei dati

La fase di preparazione dei dati è analoga a quanto fatto nel paragrafo 3.3.3, con la creazione di una colonna contenente le potenze sfalsate di un'ora che fungerà da label in fase di addestramento.

Risultati e prestazioni

Sono state provate varie configurazioni della rete variando il numero di nodi che compongono gli strati nascosti per valutare la configurazione più performante. In tutti i casi lo script è impostato per fermare automaticamente l'apprendimento una volta che la funzione Loss selezionata (in questo caso si è adoperato l'errore quadratico medio) non migliori dopo 5 epoche. I risultati relativi alle reti Feed Forward sono illustrati nella Tabella 14. Data la natura stocastica del funzionamento delle reti neurali, per ogni configurazione è stato ripetuto il processo di training varie volte ottenendo risultati leggermente diversi ma con uno scostamento dal valore medio non significativo.

Tabella 14: prestazioni dell'addestramento rete LSTM per previsioni orarie

Struttura	Training		Validation		R ²	MAE [kW]	epoche
	MSE	MAE	MSE	MAE			
10-4	0.205	0.240	0.188	0.115	0.872	51.3	70
	0.208	0.242	0.187	0.112	0.872	51.1	71
	0.210	0.239	0.188	0.114	0.872	51.3	74
	0.206	0.240	0.184	0.115	0.872	51.4	74
25-12	0.135	0.204	0.114	0.121	0.971	31.6	53
	0.134	0.204	0.114	0.121	0.971	32.3	52
	0.235	0.203	0.114	0.122	0.962	33.1	52
	0.234	0.204	0.114	0.120	0.972	32.9	55
50-25	0.112	0.155	0.098	0.101	0.968	38.7	45
	0.112	0.154	0.1001	0.102	0.960	40.0	46
	0.113	0.155	0.0991	0.102	0.968	39.7	45
	0.120	0.157	0.1004	0.111	0.971	32.3	45
100-50	0.171	0.221	0.114	0.154	0.961	32.5	40
	0.175	0.224	0.112	0.151	0.970	30.3	39
	0.174	0.225	0.116	0.159	0.972	33.2	40
	0.174	0.227	0.117	0.153	0.972	32.4	41
Ora Precedente	-	-	-	-	0.721	78.8	-

Nel caso delle reti LSTM le prestazioni non subiscono grandi oscillazioni al variare della complessità della rete, tranne che per il numero di epoche necessarie ad effettuare l'addestramento. Rispetto alle reti Feed Forward il valore di R² ha un tende ad avere un valore migliore (≈0.90 circa contro ≈0.80) anche se in entrambi i casi lo scarto fra le previsioni e i dati reali è molto contenuto in termini assoluti.

Anche con l'utilizzo di questa tipologia di rete la configurazione 25-12 risulta essere il compromesso migliore, infatti andando ad aumentare il numero di noti e la complessità della rete non si ottengono miglioramenti significativi, mentre la rete 10-4 ha prestazioni decisamente inferiori.

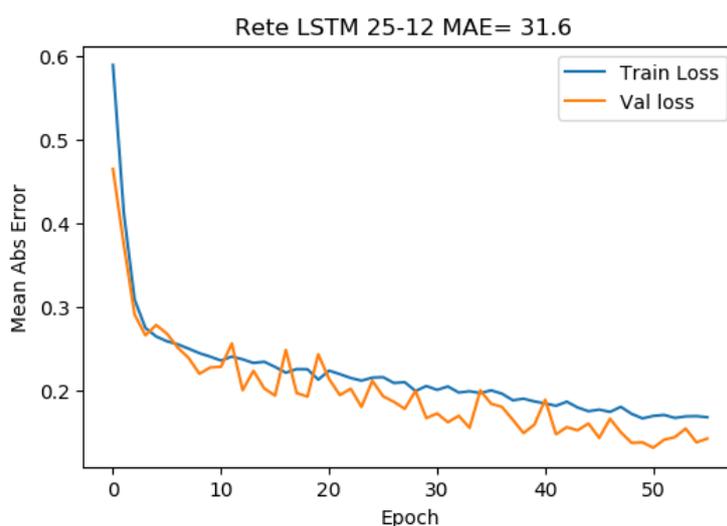


Figura 58: andamento dell'addestramento

La Figura 58 mostra l'andamento della precisione del modello in funzione delle epoche durante la fase di addestramento, rispetto alla rete Feed Forward di pari complessità la rete LSTM impiega circa la metà delle epoche per stabilizzarsi sugli stessi valori di precisione.

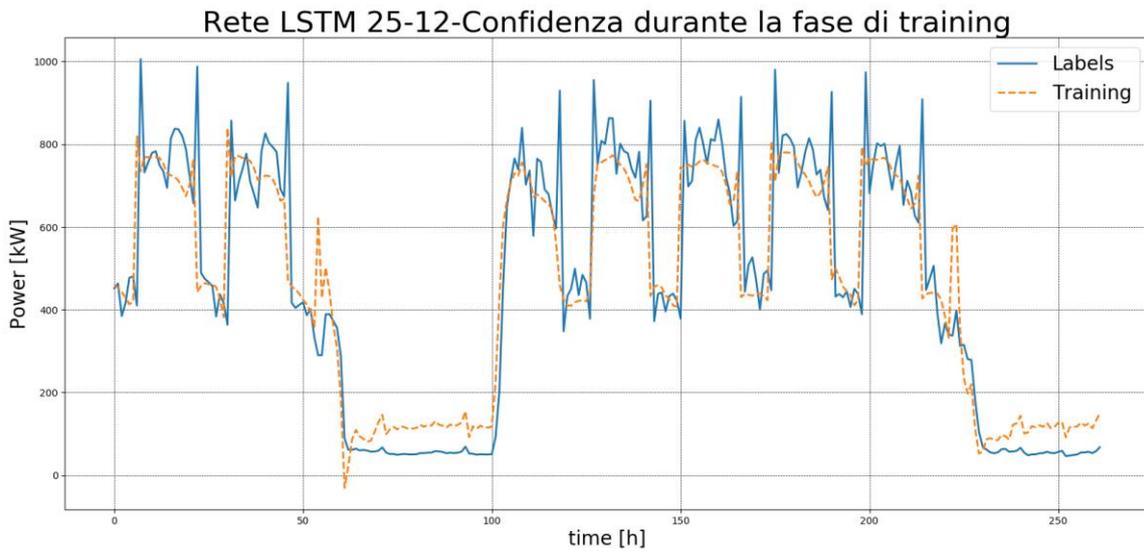


Figura 59:confidenza durante la fase di training

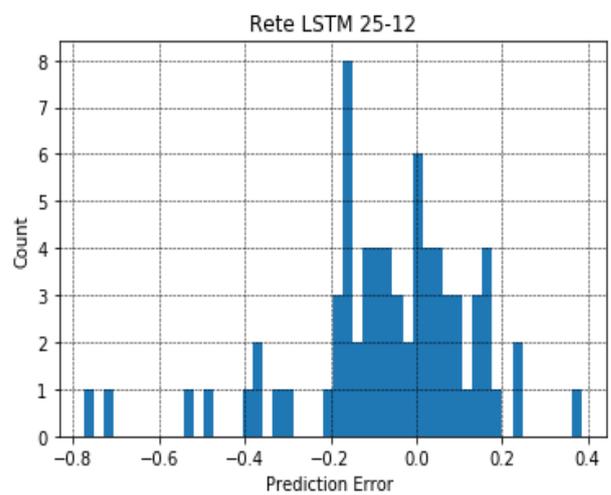
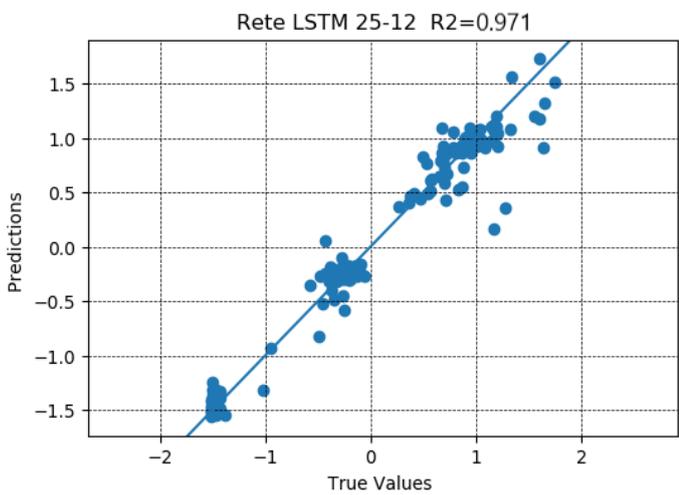


Figura 60: distribuzione errore

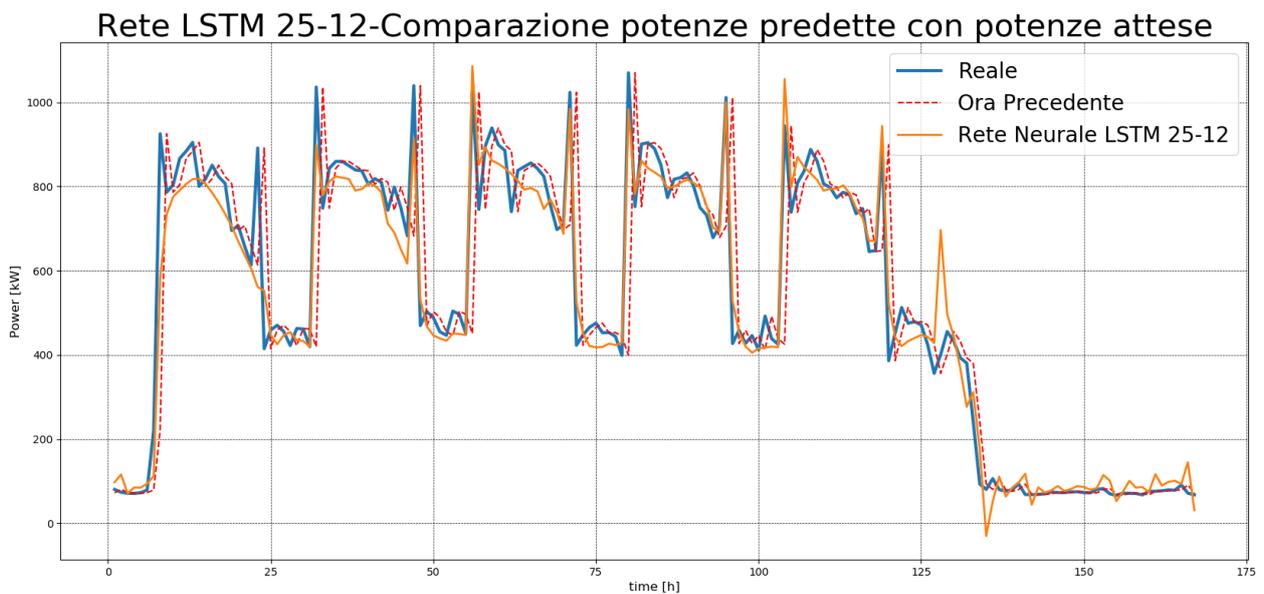


Figura 61: comparazione potenze previste con le potenze reali, rete LSTM oraria

La Figura 61 mostra il confronto fra la curva di carico reale e quella predetta dalla rete nell'arco di una settimana, si nota come anche per le reti LSTM il sistema fornisca una previsione che segue in maniera molto fedele l'andamento reale dei consumi, garantendo uno scarto nell'ordine di qualche decina di kW_{el}. Si nota anche come la curva delle previsioni abbia una varianza minore dei valori rispetto all'andamento reale, sottostimando i picchi di potenza e sovrastimando i punti di minimo.

3.3.9 Rete LSTM per la previsione giornaliera

Creazione della rete

Una volta che i dati sono stati preparati adeguatamente si può procedere alla realizzazione della struttura della rete. Il secondo modello ha l'obiettivo di prevedere la potenza elettrica nell'ora successiva basandosi sul valore (reale) precedente.

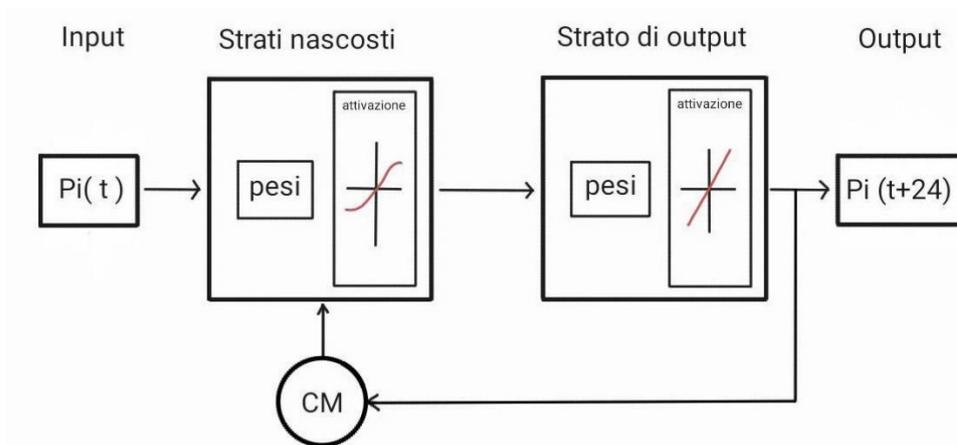


Figura 62: funzionamento della rete LSTM per le previsioni giornaliere

La rete può essere schematizzata come due blocchi rappresentanti rispettivamente gli strati nascosti e lo strato di output e da una cella di memoria (CM) che riceve le informazioni in output e le memorizza per confrontarle con l'input successivo. Fornendo la potenza attuale $P(t)$ la rete restituisce la potenza predetta per l'ora successiva $P'(t+1)$.

La sintassi per la creazione della struttura della rete è molto compatta. Il numero di nodi che compongono gli strati nascosti della rete può essere agevolmente per testare varie configurazioni e trovare quella più performante in base allo studio effettuato, mentre il numero di neuroni dello strato di output deve necessariamente essere pari al numero di output che si vuole ottenere, in questo caso 24 valori.

```

#=====
model = keras.Sequential([
    keras.layers.Dense(50, activation=tf.nn.relu,
                       input_shape=(train_x.shape[1],)),
    keras.layers.Dropout(0.2),
    keras.layers.Dense(30, activation=tf.nn.relu),
    keras.layers.Dropout(0.2),
    keras.layers.Dense(24)
])

model.compile(loss='mse',
              optimizer='Adam',
              metrics=['mae'])
#=====

```

Si sono utilizzati due strati densi nascosti e uno di output, alla fine di ogni strato denso è stato applicato un layer con funzione dropout per prevenire fenomeni di overfitting. Gli strati nascosti utilizzano la funzione di

attivazione *relu* (vedi 2.2.1). Successivamente si è definita la funzione per compilare la rete definendo i criteri di addestramento e valutazione. La funzione obiettivo utilizzata è l'errore quadratico medio (Mean Squared Error-MSE), poiché nelle previsioni regressive è un parametro significativo, ottimizzato attraverso l'algoritmo AdamOptimizer. La funzione metrica è utilizzata per visualizzare la progressione del processo di apprendimento è l'errore medio assoluto (Mean Absolute Error-MAE).

La struttura della rete è riassunta nella Tabella 18.

Tabella 15: capisaldi rete LSTM giornaliera

Rete	Layer	Tipo di layer	N° di nodi
Long Short Term Memory	Nascosto	LSTM	50
	Nascosto	LSTM	30
	Output	Denso	24

Il modello utilizza un set di 24 valori (tutte le ore di una giornata) e prevede simultaneamente 24 valori, corrispondenti alle 24 ore della giornata successiva, come schematizzato di seguito.



```
#=====
early_stop = keras.callbacks.EarlyStopping(monitor='val_loss', patience=5)

EPOCHS=200
BATCH_SIZE=24
history = model.fit(train_x, train_y,
                    batch_size=BATCH_SIZE, epochs=EPOCHS, validation_split=0.1,
                    callbacks=[early_stop], verbose=1)
#=====
```

Infine sono state definite le modalità di addestramento, in questo caso è stato impostato un valore limite di epoche molto alto, ma grazie all'implementazione di una funzione di Early Stopping il modello è in grado di fermare l'apprendimento nel momento in cui non si otterrebbero ulteriori miglioramenti, evitando di incorrere in fenomeni di overfitting.

Organizzazione dei dati

Per questo nuovo modello si utilizza lo stesso set di dati usato in precedenza, tuttavia le funzioni atte a pre processare i dati variano leggermente.

Risultati e prestazioni

Sono state provate varie configurazioni della rete variando il numero di nodi che compongono gli strati nascosti per valutare la configurazione più performante. In tutti i casi lo script è impostato per fermare automaticamente l'apprendimento una volta che la funzione Loss selezionata (in questo caso si è adoperato l'errore quadratico medio) non migliori dopo 5 epoche.

I risultati relativi alle reti LSTM sono illustrati nella Tabella 19. Data la natura stocastica del funzionamento delle reti neurali, per ogni configurazione è stato ripetuto il processo di training varie volte ottenendo risultati leggermente diversi ma con uno scostamento dal valore medio non significativo.

Tabella 16: prestazioni dell'addestramento rete LSTM giornaliera

Struttura	Training		Validation		R ²	MAE [kW]	epoche
	MSE	MAE	MSE	MAE			
35-30	0.808	0.795	0.757	0.715	0.693	129.9	60
	0.813	0.784	0.757	0.718	0.702	124.2	63
	0.806	0.783	0.753	0.713	0.690	132.1	63
	0.802	0.784	0.749	0.714	0.704	121.0	61
50-30	0.788	0.756	0.722	0.695	0.725	114.9	56
	0.784	0.754	0.724	0.698	0.728	116.2	55
	0.786	0.751	0.729	0.693	0.725	115.3	55
	0.771	0.754	0.726	0.695	0.725	114.0	54
70-50	0.788	0.756	0.722	0.695	0.720	111.0	42
	0.775	0.754	0.730	0.698	0.725	115.3	42
	0.793	0.751	0.726	0.692	0.725	115.1	40
	0.781	0.754	0.720	0.705	0.722	113.7	44
100-60	0.782	0.758	0.723	0.682	0.725	115.3	39
	0.773	0.759	0.713	0.687	0.728	116.5	38
	0.781	0.750	0.726	0.681	0.725	114.4	41
	0.783	0.753	0.724	0.694	0.728	116.8	40
Giornata Precedente	-	-	-	-	0.11	170.2	-

In questo caso le prestazioni della rete risultano sensibilmente inferiori rispetto ai casi precedenti, come si nota nella Figura 70, il modello sottostima in maniera considerevole il carico elettrico. Anche tralasciando i picchi, che sono soddisfatti tramite la rete elettrica nazionale, il modello farebbe funzionare il cogeneratore ben al di sotto delle condizioni ottimali, diminuendo l'efficienza e, di conseguenza, l'erogazione dei titoli di Efficienza Energetica. Per le previsioni giornaliere risulta quindi più performante una rete di tipo Feed Forward. Si confrontato il risultato della previsione del modello rispetto alle prestazioni che si avrebbero utilizzando i profili della giornata precedente.

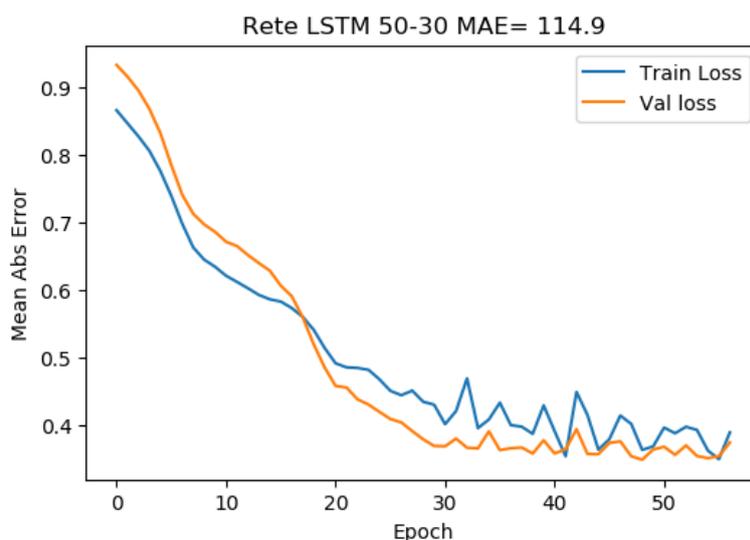


Figura 63: andamento dell'addestramento

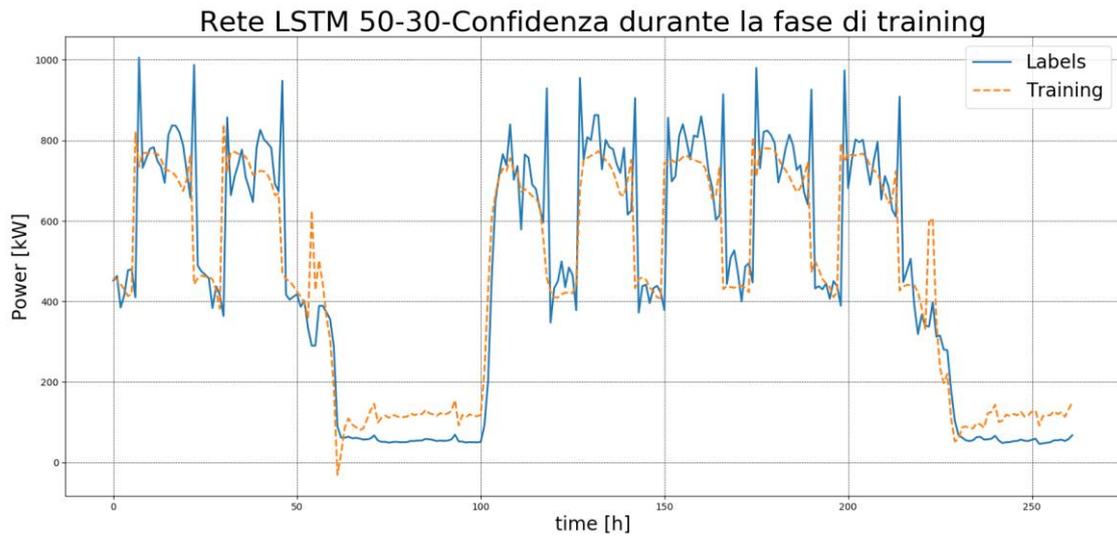


Figura 64: confidenza durante la fase di training

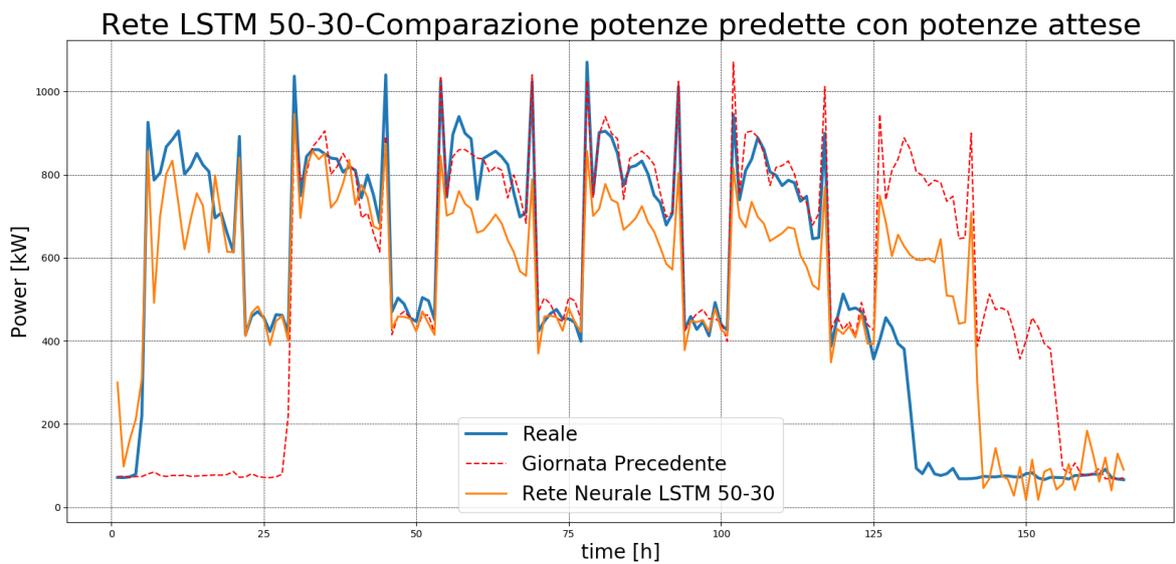
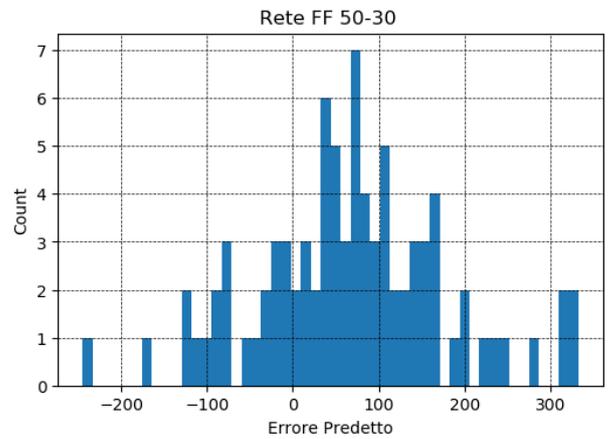
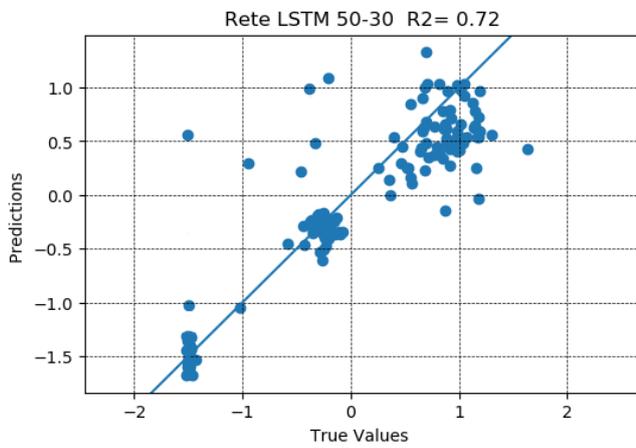


Figura 65: comparazione potenze previste con le potenze reali rete LSTM giornaliera

Anche in questo caso, come per la rete equivalente Feed Forward, il modello non è in grado di identificare i giorni del fine settimana e si è proceduto con lo sviluppo di un ulteriore modello in grado di identificare il sabato e la domenica.

3.3.10 Confronto fra i modelli

Lo sviluppo di modelli previsionali basati sulle reti neurali ha prodotto dei buoni risultati nello studio di questo stabilimento. Lo sviluppo in parallelo di due tipologie di rete (Feed Forward e Long Short Term Memory) ha fornito risultati molto simili, ed entrambe costituiscono una valida opzione per questa tipologia di problemi. Di seguito sono riportati i valori medi delle prestazioni per ogni rete e paragonati ai risultati ottenibili effettuando una semplice traslazione dei consumi precedenti coerentemente con l'intervallo di previsione adottata (un'ora, una giornata e una settimana).

Tabella 17: valori medi degli indici di prestazione

Previsione oraria			Previsione giornaliera			Previsione settimanale		
Rete	R ²	MAE [kW]	Rete	R ²	MAE [kW]	Rete	R ²	MAE [kW]
FF	0,925	53	FF	0,672	130	FF	0,945	45
LSTM	0,95	33	LSTM	0,723	115	LSTM	0,951	44
Rispetto all'ora precedente	0,721	79	Rispetto al giorno precedente	0,11	170	Rispetto alla settimana precedente	0,932	48

La previsione di una singola ora risulta molto accurata, anche se l'effettiva utilità a fini pratici è limitata; una semplice traslazione dell'ora precedente non è soddisfacente poiché i consumi sono caratterizzati da numerosi picchi di breve durata. A causa dell'interruzione della produzione durante il fine settimana i modelli di previsione giornaliera risultano inaffidabili. I modelli di previsione settimanale, pur essendo più complessi da sviluppare consentono di ottenere previsioni estremamente precise ed affidabili; questo modello può essere comunque utilizzato per prevedere una singola giornata, risultano molto flessibile nell'utilizzo operativo.

3.3.11 Rete LSTM per la previsione settimanale

Creazione della rete

Una volta che i dati sono stati preparati adeguatamente si può procedere alla realizzazione della struttura della rete. Il secondo modello ha l'obiettivo di prevedere la potenza elettrica nell'ora successiva basandosi sul valore (reale) precedente.

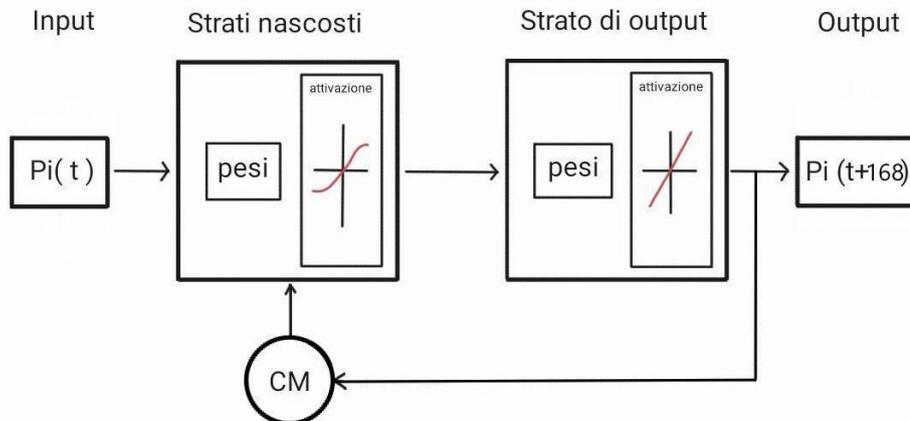


Figura 66: funzionamento della rete LSTM per le previsioni settimanali

La rete può essere schematizzata come due blocchi rappresentanti rispettivamente gli strati nascosti e lo strato di output e da una cella di memoria (CM) che riceve le informazioni in output e le memorizza per confrontarle con l'input successivo. Fornendo la potenza di un'intera settimana $P_i(t)$ la rete restituisce la potenza predetta per la settimana successiva $P'(t+1)$.

La sintassi per la creazione della struttura della rete è molto compatta. Il numero di nodi che compongono gli strati nascosti della rete può essere agevolmente per testare varie configurazioni e trovare quella più performante in base allo studio effettuato, mentre il numero di neuroni dello strato di output deve necessariamente essere pari al numero di output che si vuole ottenere, in questo caso 168 valori.

```
#=====
model = keras.Sequential([
    keras.layers.Dense(50, activation=tf.nn.relu,
                       input_shape=(train_x.shape[1],)),
    keras.layers.Dropout(0.2)
    keras.layers.Dense(30, activation=tf.nn.relu),
    keras.layers.Dropout(0.2)
    keras.layers.Dense(168)
])

model.compile(loss='mse',
              optimizer='Adam',
              metrics=['mae'])
#=====
```

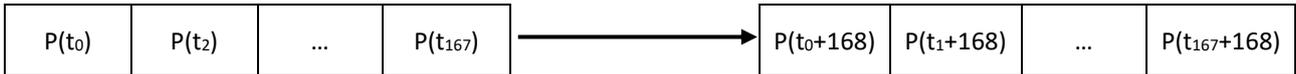
Si sono utilizzati due strati densi nascosti e uno di output, alla fine di ogni strato denso è stato applicato un layer con funzione dropout per prevenire fenomeni di overfitting. Gli strati nascosti utilizzano la funzione di attivazione *relu* (vedi 2.2.1). Successivamente si è definita la funzione per compilare la rete definendo i criteri di addestramento e valutazione. La funzione obiettivo utilizzata è l'errore quadratico medio (Mean Squared Error-MSE), poiché nelle previsioni regressive è un parametro significativo, ottimizzato attraverso l'algoritmo AdamOptimizer. La funzione metrica è utilizzata per visualizzare la progressione del processo di apprendimento è l'errore medio assoluto (Mean Absolute Error-MAE).

La struttura della rete è riassunta nella Tabella 18.

Tabella 18: capisaldi rete LSTM giornaliera

Rete	Layer	Tipo di layer	N° di nodi
Long Short Term Memory	Nascosto	LSTM	50
	Nascosto	LSTM	30
	Output	Denso	168

Il modello utilizza un set di 24 valori (tutte le ore di una giornata) e prevede simultaneamente 24 valori, corrispondenti alle 24 ore della giornata successiva, come schematizzato di seguito.



```

#=====
early_stop = keras.callbacks.EarlyStopping(monitor='val_loss', patience=5)

EPOCHS=200
BATCH_SIZE=1
history = model.fit(train_x, train_y,
                    batch_size=BATCH_SIZE, epochs=EPOCHS, validation_split=0.1,
                    callbacks=[early_stop], verbose=1)
#=====

```

Infine sono state definite le modalità di addestramento, in questo caso è stato impostato un valore limite di epoche molto alto, ma grazie all'implementazione di una funzione di Early Stopping il modello è in grado di fermare l'apprendimento nel momento in cui non si otterrebbero ulteriori miglioramenti, evitando di incorrere in fenomeni di overfitting.

Organizzazione dei dati

Per questo nuovo modello si utilizza lo stesso set di dati usato in precedenza, tuttavia le funzioni atte a preprocessare i dati variano leggermente.

A questo scopo si è utilizzato l'indice del dataset generato automaticamente per creare i due sottoinsiemi ed a entrambi si è applicata la funzione per pre-processare i dati:

- Il **Training Set**: comprende il 90% dei dati e viene utilizzato per addestrare il modello. A sua volta sarà suddiviso automaticamente da programma durante la fase di addestramento nel Training Subset e nel Validation Subset (VEDI) rispettivamente 10% e 80% del totale. Rispetto alla previsione oraria si è diminuita la dimensione del training set al fine di avere due giornate intere nel Test Set;
- Il **Test Set**: è composto dal 10% dei dati e viene utilizzato per verificare la bontà del modello testandolo su dai dati che non ha mai visto in precedenza.

Risultati e prestazioni

Sono state provate varie configurazioni della rete variando il numero di nodi che compongono gli strati nascosti per valutare la configurazione più performante. In tutti i casi lo script è impostato per fermare automaticamente l'apprendimento una volta che la funzione Loss selezionata (in questo caso si è adoperato l'errore quadratico medio) non migliori dopo 5 epoche.

I risultati relativi alle reti LSTM sono illustrati nella Tabella 19. Data la natura stocastica del funzionamento delle reti neurali, per ogni configurazione è stato ripetuto il processo di training varie volte ottenendo risultati leggermente diversi ma con uno scostamento dal valore medio non significativo.

Tabella 19: prestazioni dell'addestramento rete LSTM settimanale

Struttura	Training		Validation		R ²	MAE [kW]	epoche
	MSE	MAE	MSE	MAE			
35-30	0.808	0.795	0.757	0.715	0.894	59.4	39

	0.813	0.784	0.757	0.718	0.904	54.7	43
	0.806	0.783	0.753	0.713	0.891	62.6	43
	0.802	0.784	0.749	0.714	0.904	58.0	41
50-30	0.788	0.756	0.722	0.695	0.952	44.5	26
	0.784	0.754	0.724	0.698	0.952	46.6	25
	0.786	0.751	0.729	0.693	0.952	45.1	25
	0.771	0.754	0.726	0.695	0.952	44.4	24
70-50	0.788	0.756	0.722	0.695	0.949	41.3	22
	0.775	0.754	0.730	0.698	0.952	45.8	22
	0.793	0.751	0.726	0.692	0.952	45.6	21
	0.781	0.754	0.720	0.705	0.951	43.3	24
100-60	0.782	0.758	0.723	0.682	0.952	45.0	20
	0.773	0.759	0.713	0.687	0.952	46.4	21
	0.781	0.750	0.726	0.681	0.952	44.2	20
	0.783	0.753	0.724	0.694	0.952	46.0	20
Settimana Precedente	-	-	-	-	0.932	48.0	-

In questo caso le prestazioni della rete risultano un poco inferiori rispetto ai casi precedenti, come si nota nella Figura 69 e nella Figura 70, il modello sottostima in maniera considerevole il carico elettrico. Anche tralasciando i picchi, che sono soddisfatti tramite la rete elettrica nazionale, il modello farebbe funzionare il cogeneratore ben al di sotto delle condizioni ottimali, diminuendo l'efficienza e, di conseguenza, l'erogazione dei titoli di Efficienza Energetica. Per le previsioni giornaliere risulta quindi più performante una rete di tipo Feed Forward.

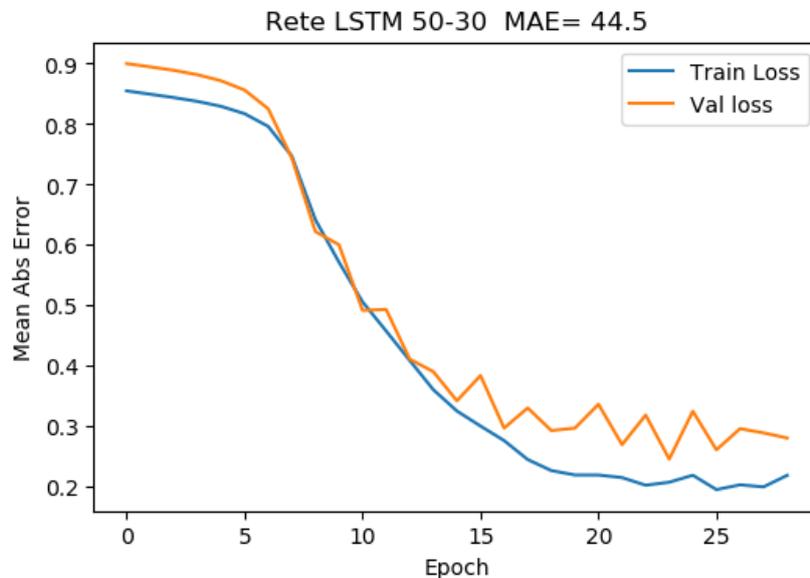


Figura 67: andamento dell'addestramento

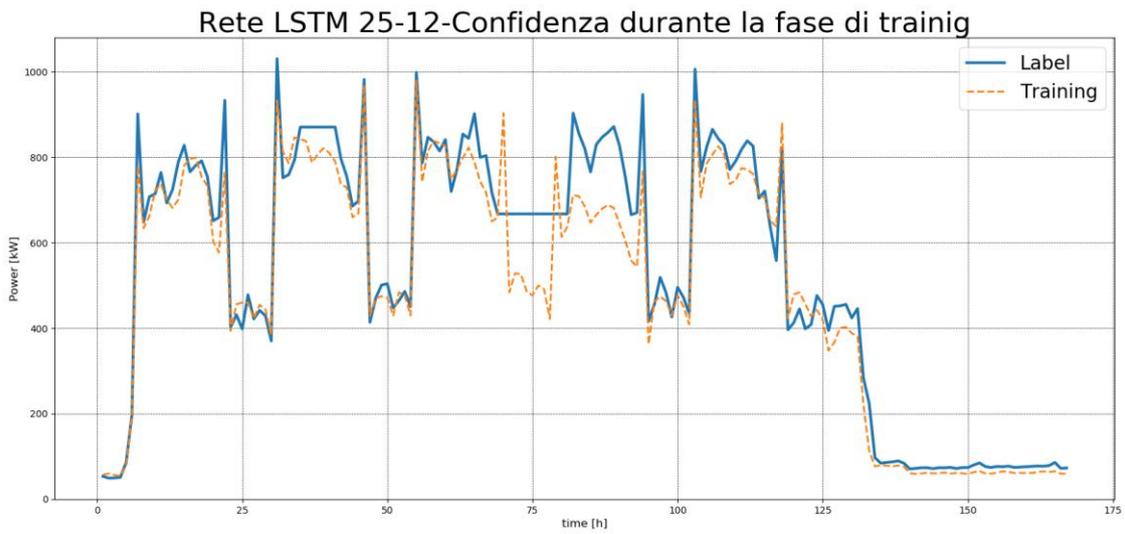


Figura 68: confidenza durante la fase di training

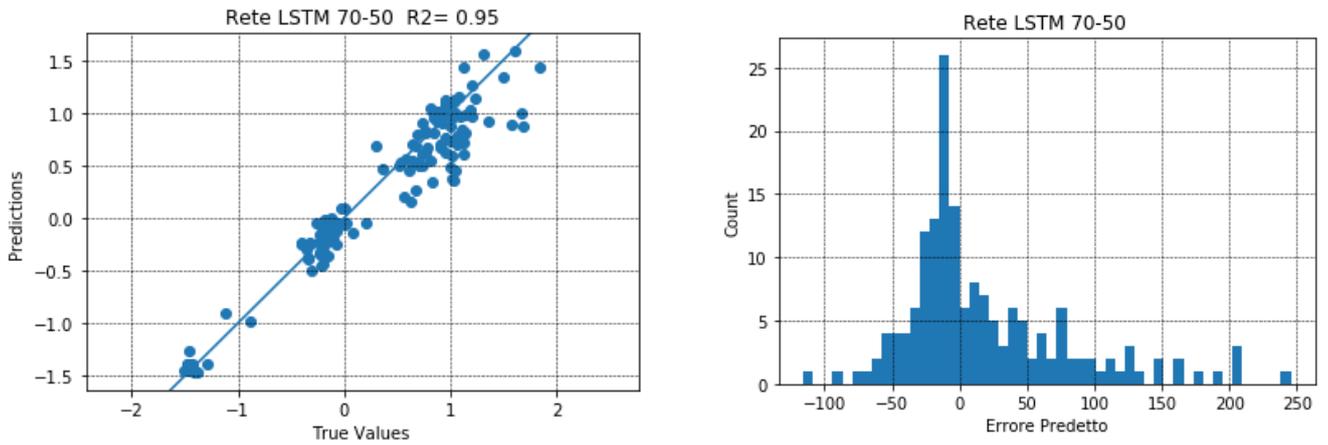


Figura 69: distribuzione dell'errore

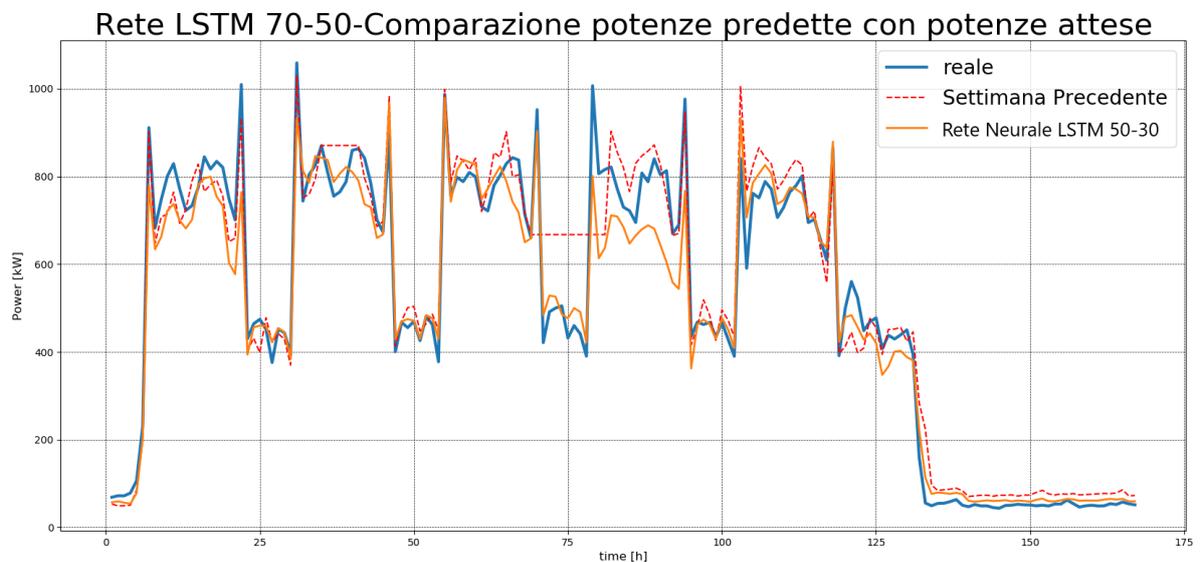


Figura 70: comparazione potenze previste con le potenze reali rete LSTM settimanale

3.4 Industria Alimentare

3.4.1 Gruppo di cogenerazione e impianto fotovoltaico

Il secondo stabilimento preso in oggetto è uno stabilimento alimentare, questa tipologia di stabilimento produttivo utilizza grandi quantità di acqua calda e vapore nei processi di lavorazione, rendendo particolarmente vantaggioso l'utilizzo di un impianto di cogenerazione.



Figura 71: gruppo di cogenerazione

Il modulo cogenerativo fornisce una potenza elettrica a pieno carico pari a 2680 kW_{el} e una potenza termica tramite cogenerazione di 3060 kW_{th}.

Il modulo cogenerativo ha un primo stadio di recupero termico all'interno del gruppo motore (attraverso il circuito dell'olio, il circuito acqua delle camicie motore e il primo stadio di interrefrigerazione) che consente un recupero di 190 kW_{th} in forma di acqua calda. Uno scambiatore di calore successivo recupera la potenza termica proveniente dal circuito di raffreddamento del motore e 300 kW_{th} della batteria di preriscaldamento installata nel circuito per il recupero dei fumi che produce acqua calda a circa 80 °C.

I fumi di combustione del motore passano attraverso uno scambiatore per il recupero fumi in grado di recuperare 2560 kW_{th}.

Tabella 20: dati di targa

PCI metano	9,5	kWh/Nm ³
Consumo di gas	630	Nm ³ /h
Potenza in ingresso	5970	kW
Potenza elettrica erogata	2680	kW
Potenza termica recuperabile	3060	kW
Rendimento elettrico	44,8	%
Rendimento termico	51,2	%
Rendimento complessivo	96,0	%

Con i dati utilizzati nello studio sono stati valutati gli indici di prestazione relativi alla cogenerazione (vedi 0) e riportati nella Tabella 6: indici mensili relativi al primo impianto di cogenerazione contenenti le stime di risparmio relative ad un anno di esercizio (si veda la normativa in allegato).

Tabella 21: indici relativi al secondo impianto di cogenerazione

PES	RISP [MWh/anno]	Certificati Bianchi annuali
0,17	12200	1370

Ne risulta che lo stabilimento rientra nella categoria CAR (Cogenerazione ad Alto Rendimento).

Lo stabilimento è alimentato anche da un parco fotovoltaico che consente una produzione elettrica di picco di 1 MW_{el}. La produzione elettrica da parte dell'impianto fotovoltaico ha la priorità di utilizzo rispetto a quella prodotta dal cogeneratore, il quale viene regolato di conseguenza, aumentando o diminuendo la produzione elettrica. Le apparecchiature di misurazione e controllo di questo stabilimento sono in grado di fornire i dati relativi alla quantità di energia prodotta dal parco fotovoltaico, ma l'azienda non fornisce informazioni sulla quota autoconsumata e la quota ceduta alla rete. Rispetto al caso precedente l'analisi si concentrerà sulla curva di carico del cogeneratore.

3.4.2 Apparecchiature per il monitoraggio

L'energia elettrica autoprodotta nell'impianto di cogenerazione viene contabilizzata tramite un apposito gruppo di misura fiscale installato subito a valle dell'alternatore. Il gruppo misura l'energia elettrica attiva di tipo trifase statico.



Figura 72: strumento PLC di misura dell'energia elettrica prodotta

L'energia elettrica prodotta, al netto della quota parte utilizzata per l'alimentazione delle apparecchiature ausiliarie del gruppo di generazione degli assorbimenti della centrale e le perdite dovute alla rete e alla trasformazione delle tensioni, è interamente usata per soddisfare il fabbisogno energetico dello stabilimento. Eventuali picchi di richiesta non soddisfatti dall'impianto di cogenerazione sono soddisfatti prelevando energia dalla rete elettrica nazionale

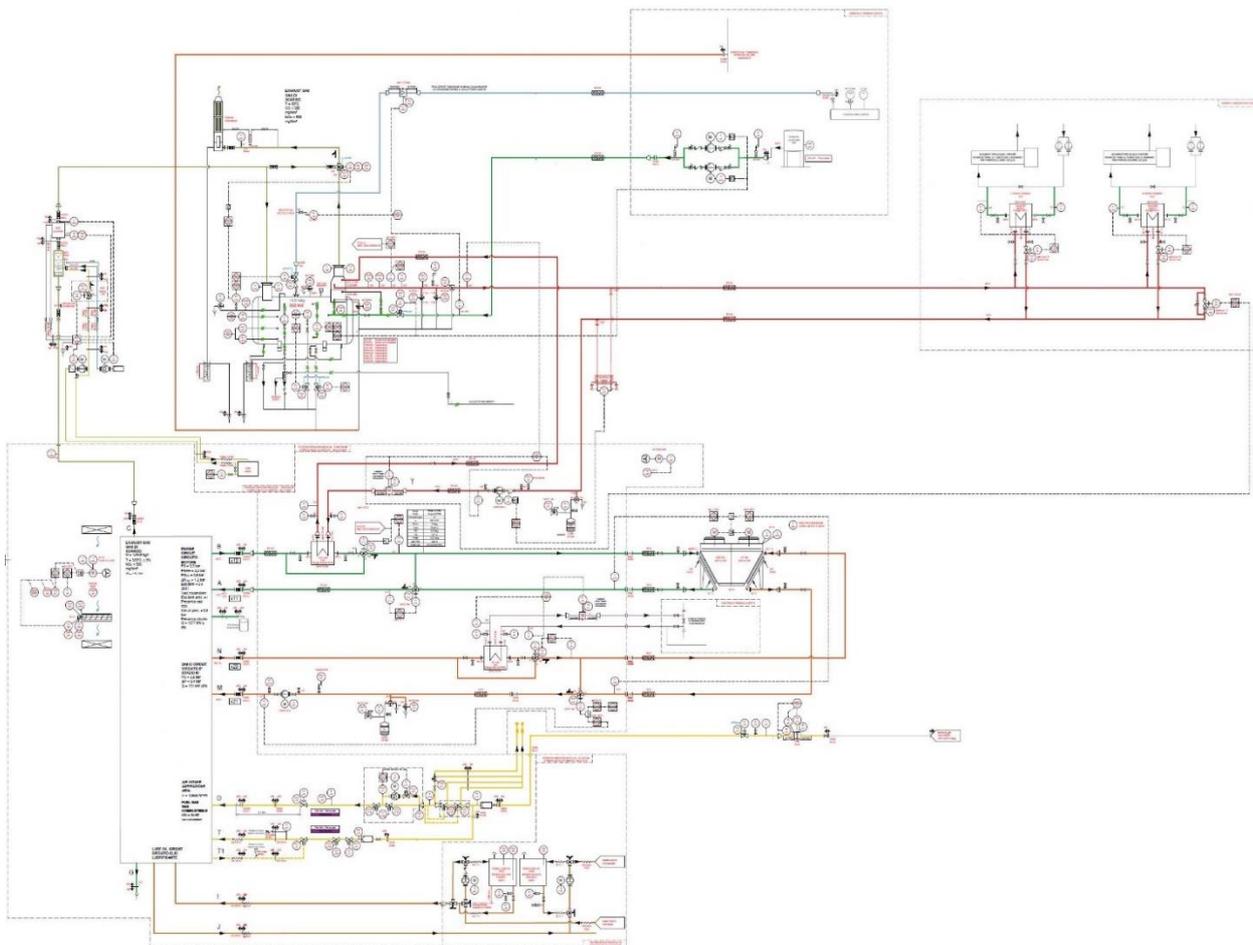


Figura 73: schema di funzionamento dell'impianto di cogenerazione

3.4.3 Preparazione e formattazione dei dati

I parametri di funzionamento caratteristici dell'impianto sono acquisiti attraverso una rete di monitoraggio attraverso delle apparecchiature SCADA ed inviati ad una piattaforma cloud di stoccaggio. Le misurazioni sono disponibili in un file .csv in cui è registrati, per ogni intervallo temporale la potenza prodotta dall'impianto di cogenerazione dello stabilimento (termica ed elettrica), la potenza assorbita dalla rete e i consumi di combustibile.

Timestamp	Potenza Elettrica CHP	Potenza Elettrica prelevata da rete	Potenza Termica in Vapore	Potenza Termica in Acqua Calda (AT+BT)	Consumi Gas Naturale
10/01/2018 00:59:59	2166	101	841	845	511
10/01/2018 01:59:59	2180	98	818	863	518
10/01/2018 02:59:59	2246	101	844	888	539
10/01/2018 03:59:59	2194	100	838	867	529
10/01/2018 04:59:59	2206	100	832	881	527
10/01/2018 05:59:59	2232	100	832	896	523
10/01/2018 06:59:59	2262	98	834	924	527
10/01/2018 07:59:59	2248	102	833	928	526
10/01/2018 08:59:59	2208	98	826	911	525
10/01/2018 09:59:59	2082	98	806	857	497
10/01/2018 10:59:59	2020	96	791	827	485
10/01/2018 11:59:59	1990	104	810	813	473

Figura 74: struttura originale dei dati acquisiti

Il formato .csv viene utilizzato poiché è molto leggero e consente una lettura più veloce da parte del programma rispetto ad un file .xlsx (file in formato Excel).

Il primo e più importante passo nella scrittura del programma consiste nell'organizzare opportunamente i dati a disposizione da far visionare alla rete per il processo di addestramento poiché senza una corretta formattazione la rete non sarà in grado di analizzare correttamente gli input che le vengono forniti.

3.4.4 Rete FF per la previsione oraria

Creazione della rete

Una volta che i dati sono stati preparati adeguatamente si può procedere alla realizzazione della struttura della rete. La prima tipologia di rete studiata è una rete Feed Forward, la tipologia più semplice di rete ma che risulta performante in una grande varietà di applicazioni.

Il primo modello ha l'obiettivo di prevedere la potenza elettrica nell'ora successiva fornendo il valore dell'ora attuale. La rete può essere schematizzata come due blocchi separati composti dagli strati nascosti e dallo strato finali di output, i quali ricevono l'input, lo elaborano in base ai pesi delle connessioni e alla rispettiva funzione di attivazione dello strato. Fornendo quindi le potenze precedenti reali $P(t)$ il modello restituisce la potenza prevista per l'ora successiva $P'(t+1)$.

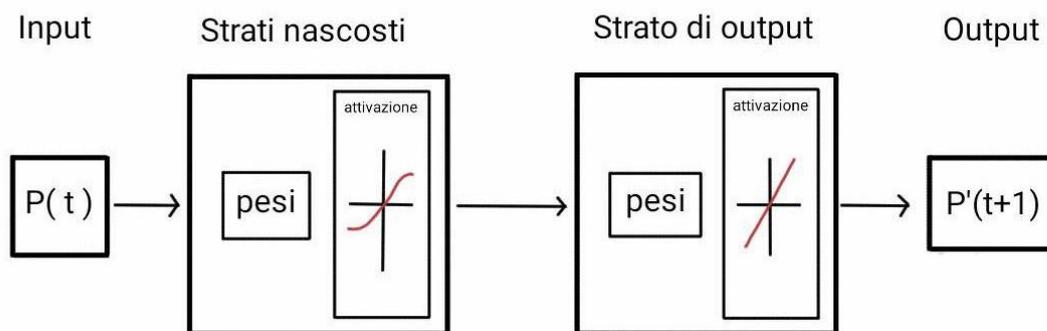


Figura 75: funzionamento della rete FF per le previsioni orarie

La sintassi per la creazione della struttura della rete è molto compatta. Il numero di nodi che compongono gli strati nascosti della rete può essere agevolmente per testare varie configurazioni e trovare quella più performante in base allo studio effettuato, mentre il numero di neuroni dello strato di output deve necessariamente essere pari al numero di output che si vuole ottenere, in questo caso un singolo valore.

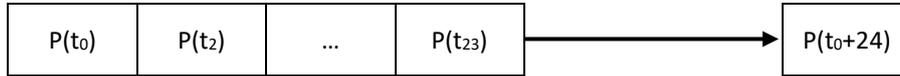
Si sono utilizzati due strati densi nascosti e uno di output, alla fine di ogni strato denso è stato applicato un layer con funzione dropout per prevenire fenomeni di overfitting. Gli strati nascosti utilizzano la funzione di attivazione *relu* (vedi 2.2.1). Successivamente si è definita la funzione per compilare la rete definendo i criteri di addestramento e valutazione. La funzione obiettivo utilizzata è l'errore quadratico medio (Mean Squared Error-MSE), poiché nelle previsioni regressive è un parametro significativo, ottimizzato attraverso l'algoritmo AdamOptimizer. La funzione metrica è utilizzata per visualizzare la progressione del processo di apprendimento è l'errore medio assoluto (Mean Absolute Error-MAE).

La struttura della rete è riassunta nella Tabella 22.

Tabella 22: capisaldi rete FF oraria

Rete	Layer	Tipo di layer	N° di nodi
Feed Forward	Nascosto	Denso	25
	Nascosto	Denso	12
	Output	Denso	1

La finestra di dati utilizzata comprende le 24 ore precedenti al valore calcolato, come schematizzato di seguito.



In fine sono state definite le modalità di addestramento, in questo caso è stato impostato un valore limite di epoche molto alto, ma grazie all'implementazione di una funzione di Early Stopping il modello è in grado di fermare l'apprendimento nel momento in cui non si otterrebbero ulteriori miglioramenti, evitando di incorrere in fenomeni di overfitting.

Organizzazione dei dati

Particolarmente importante è la normalizzazione delle grandezze utilizzate, poiché, sebbene la rete sia in grado di analizzare i dati anche con le loro grandezze "reali", applicando una normalizzazione la lettura risulta molto più veloce. In questo studio si è applicata una normalizzazione sul valore medio delle grandezze.

Risultati e prestazioni

Sono state provate varie configurazioni della rete variando il numero di nodi che compongono gli strati nascosti per valutare la configurazione più performante. In tutti i casi lo script è impostato per fermare automaticamente l'apprendimento una volta che la funzione Loss selezionata (in questo caso si è adoperato l'errore quadratico medio) non migliori dopo 5 epoche.

I risultati relativi alle reti Feed Forward sono illustrati nella Tabella 23. Data la natura stocastica del funzionamento delle reti neurali, per ogni configurazione è stato ripetuto il processo di training varie volte ottenendo risultati leggermente diversi ma con uno scostamento dal valore medio non significativo.

Tabella 23: prestazioni dell'addestramento rete FF oraria

Struttura	Training		Validation		R ²	MAE [kW]	epoche
	MSE	MAE	MSE	MAE			
10-4	0.245	0.316	0.461	0.444	0.824	53.1	31
	0.247	0.315	0.461	0.444	0.824	52.9	28
	0.246	0.319	0.463	0.446	0.813	52.1	30
	0.243	0.325	0.460	0.442	0.823	53.2	30
25-12	0.221	0.281	0.4244	0.424	0.856	47.8	17
	0.221	0.283	0.4256	0.425	0.856	47.4	17
	0.221	0.284	0.4264	0.426	0.845	49.3	17
	0.222	0.282	0.4238	0.428	0.856	48.0	16
50-25	0.225	0.290	0.431	0.436	0.849	50.6	16
	0.226	0.289	0.430	0.437	0.856	47.1	17
	0.224	0.291	0.432	0.435	0.856	48.6	18
	0.227	0.291	0.432	0.436	0.856	47.0	18
100-50	0.227	0.293	0.433	0.436	0.856	46.9	15
	0.227	0.292	0.435	0.432	0.856	48.1	16
	0.226	0.296	0.436	0.438	0.865	47.2	17
	0.228	0.298	0.455	0.435	0.856	47.8	16
Ora Precedente	-	-	-	-	0.517	108	-

Si noti come le prestazioni delle varie reti siano molto simili fra loro, l'unico elemento di differenza è, come prevedibile, il numero di epoche: una rete più semplice impiegherà un tempo maggiore per imparare i pattern caratteristici del dataframe. Le differenze rispetto gli altri indici caratteristici (errore medio assoluto (MAE) sul training set e sul validation set, R^2 e l'errore medio sui valori denormalizzati) sono trascurabili e assumono tutti ottimi risultati. Per questa analisi un buon compromesso risulta essere la configurazione a tre strati rispettivamente composti da 25,12 e 1 neurone. Si ricorda che il numero di neuroni dello strato finale deve essere uguale al numero di output attesi, in questo caso solamente uno essendo la previsione su di una singola ora.

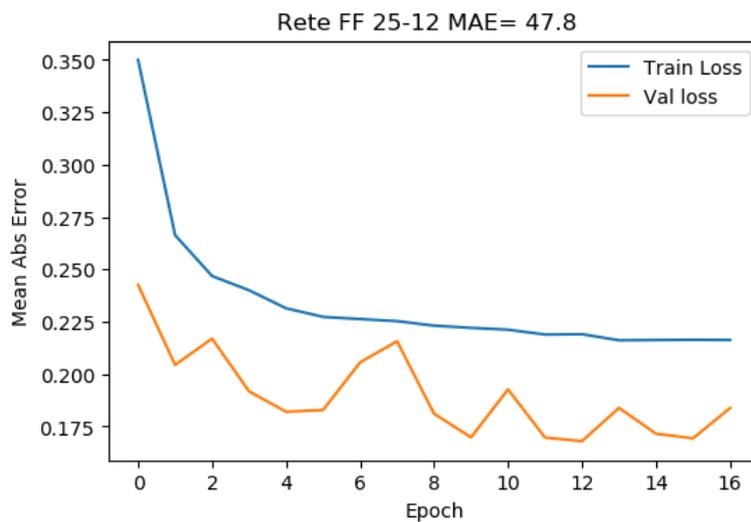


Figura 76: andamento dell'addestramento

La progressione dell'apprendimento del modello durante la fase di training è mostrata nella Figura 42, si osserva l'andamento iperbolico dell'errore in funzione delle epoche. L'errore del Training Set e del Validation Set mantengono valori molto simili durante tutto il processo e alla fine tendono a stabilizzarsi sullo stesso risultato. I dati utilizzati per l'addestramento compongono il 95% del totale (Training Set), il restante 5% è stato utilizzato per effettuare le previsioni e compararle con i valori reali (Test Set).

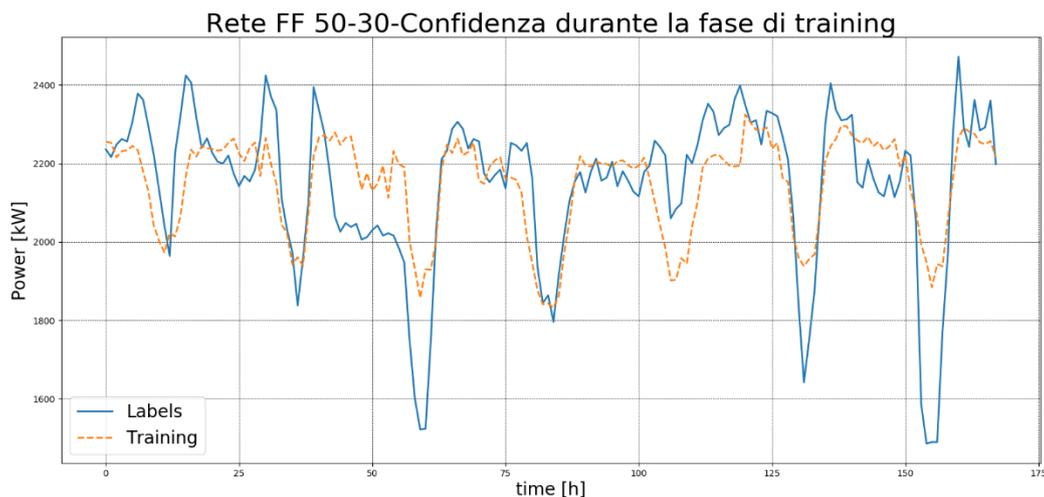


Figura 77: confidenza durante la fase di training

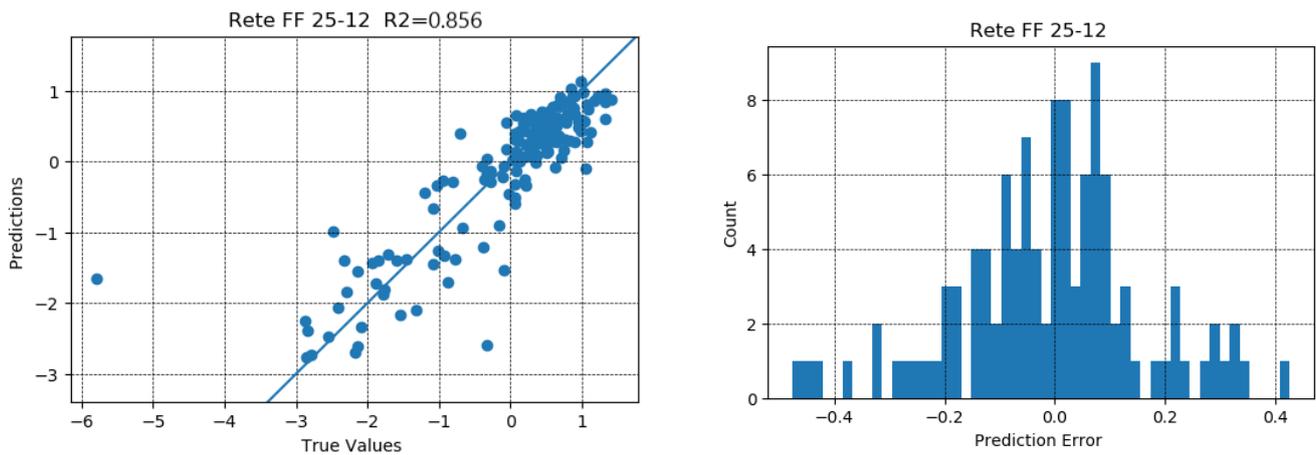


Figura 78: distribuzione dell'errore

Dalla Figura 78 si osserva come le previsioni della rete abbiano un errore contenuto rispetto ai valori reali e con una distribuzione che tende alla distribuzione normale.

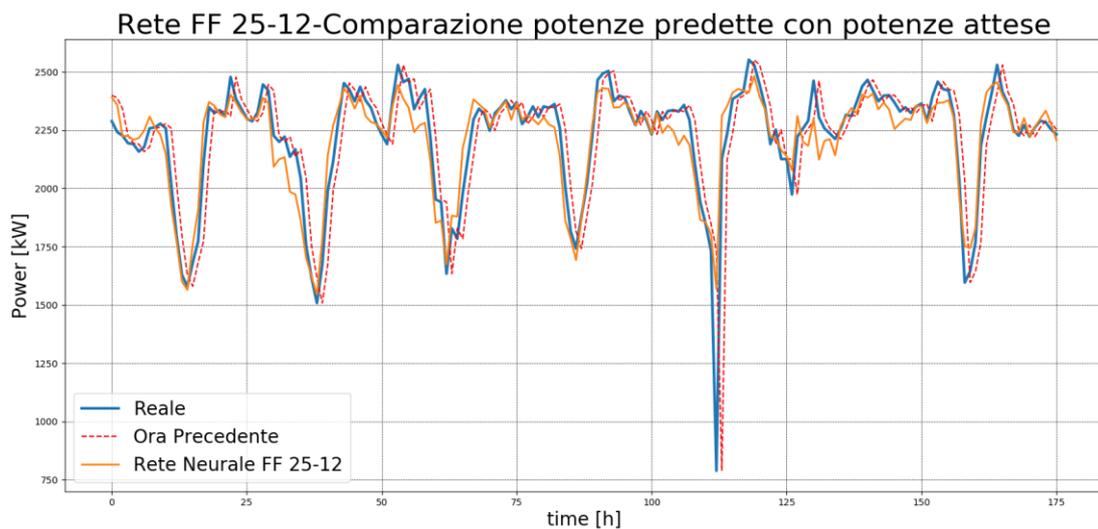


Figura 79: comparazione potenze previste con le potenze reali, rete FF oraria

La Figura 79 mostra l'andamento delle potenze predette dalla rete rispetto alle potenze reali nell'arco di una settimana. Si osserva una notevole capacità della rete di fornire previsioni che seguano l'andamento reale dovuta all'andamento molto regolare dei profili di consumo dello stabilimento, anche se le potenze predette tendono ad essere lievemente inferiori rispetto a quelle reali pur mantenendosi in uno scarto di poche decine di kW_{el}.

3.4.5 Rete FF per la previsione giornaliera

Creazione della rete

Una volta che i dati sono stati preparati adeguatamente si può procedere alla realizzazione della struttura della rete. Il secondo modello ha l'obiettivo di prevedere la potenza elettrica di un'intera giornata basandosi sulle misurazioni della giornata precedente. La rete può essere schematizzata come due blocchi separati composti dagli strati nascosti e dallo strato finali di output, i quali ricevono l'input, lo elaborano in base ai pesi delle connessioni e alla rispettiva funzione di attivazione dello strato. Fornendo quindi tutte le potenze nell'arco di una giornata $P_i(t)$ il modello restituisce la potenza prevista per la stessa ora nella giornata successiva $P_i'(t+24)$.

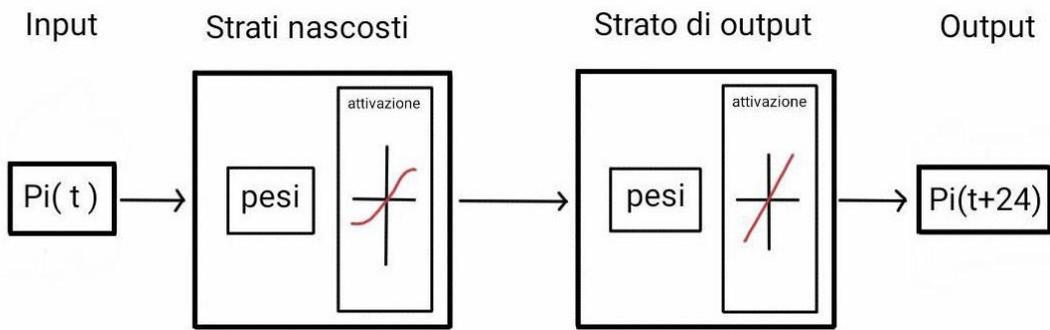


Figura 80: funzionamento della rete FF per le previsioni giornaliere

La sintassi per la creazione della struttura della rete è molto compatta. Il numero di nodi che compongono gli strati nascosti della rete può essere agevolmente per testare varie configurazioni e trovare quella più performante in base allo studio effettuato, mentre il numero di neuroni dello strato di output deve necessariamente essere pari al numero di output che si vuole ottenere, in questo caso 24 valori.

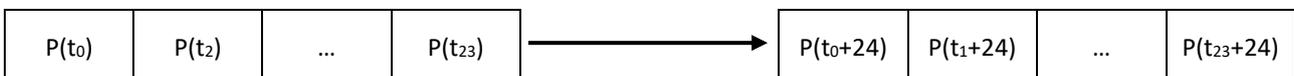
Si sono utilizzati due strati densi nascosti e uno di output, alla fine di ogni strato denso è stato applicato un layer con funzione dropout per prevenire fenomeni di overfitting. Gli strati nascosti utilizzano la funzione di attivazione *relu* (vedi 2.2.1). Successivamente si è definita la funzione per compilare la rete definendo i criteri di addestramento e valutazione. La funzione obiettivo utilizzata è l'errore quadratico medio (Mean Squared Error-MSE), poiché nelle previsioni regressive è un parametro significativo, ottimizzato attraverso l'algoritmo AdamOptimizer. La funzione metrica è utilizzata per visualizzare la progressione del processo di apprendimento è l'errore medio assoluto (Mean Absolute Error-MAE).

La struttura della rete è riassunta nella Tabella 24.

Tabella 24: capisaldi rete FF giornaliera

Rete	Layer	Tipo di layer	N° di nodi
Feed Forward	Nascosto	Denso	70
	Nascosto	Denso	50
	Output	Denso	24

Il modello utilizza un set di 24 valori (tutte le ore di una giornata) e prevede simultaneamente 24 valori, corrispondenti alle 24 ore della giornata successiva, come schematizzato di seguito.



Infine sono state definite le modalità di addestramento, in questo caso è stato impostato un valore limite di epoche molto alto, ma grazie all'implementazione di una funzione di Early Stopping il modello è in grado di fermare l'apprendimento nel momento in cui non si otterrebbero ulteriori miglioramenti, evitando di incorrere in fenomeni di overfitting.

Organizzazione dei dati

Per questo nuovo modello si utilizza lo stesso set di dati usato in precedenza, tuttavia le funzioni utilizzate per preprocessare i dati variano leggermente. A questo scopo si è utilizzato l'indice del dataset generato automaticamente per creare i due sotto insiemi ed a entrambi si è applicata la funzione per pre-processare i dati.

Risultati e prestazioni

Sono state provate varie configurazioni della rete variando il numero di nodi che compongono gli strati nascosti per valutare la configurazione più performante. In tutti i casi lo script è impostato per fermare automaticamente l'apprendimento una volta che la funzione Loss selezionata (in questo caso si è adoperato l'errore quadratico medio) non migliori dopo 5 epoche.

I risultati relativi alle reti Feed Forward sono illustrati nella Tabella 25. Data la natura stocastica del funzionamento delle reti neurali, per ogni configurazione è stato ripetuto il processo di training varie volte ottenendo risultati leggermente diversi ma con uno scostamento dal valore medio non significativo.

Tabella 25: prestazioni dell'addestramento rete FF giornaliera

Struttura	Training		Validation		R ²	MAE [kW]	epoche
	MSE	MAE	MSE	MAE			
35-30	0.455	0.474	0.429	0.431	0.309	160	21
	0.444	0.468	0.429	0.432	0.310	162	20
	0.448	0.470	0.430	0.431	0.309	161	19
	0.465	0.471	0.430	0.431	0.309	160	20
50-30	0.445	0.464	0.424	0.424	0.351	151	15
	0.444	0.465	0.423	0.424	0.349	157	15
	0.448	0.466	0.425	0.424	0.349	155	14
	0.445	0.467	0.425	0.425	0.358	152	16
70-50	0.443	0.456	0.427	0.428	0.339	151	13
	0.444	0.458	0.428	0.428	0.351	142	15
	0.444	0.457	0.427	0.427	0.351	151	14
	0.447	0.457	0.426	0.429	0.349	156	14
100-60	0.449	0.443	0.422	0.436	0.349	147	14
	0.450	0.442	0.423	0.437	0.351	151	15
	0.451	0.443	0.423	0.436	0.338	158	13
	0.449	0.443	0.423	0.436	0.351	144	15
Giornata Precedente	-	-	-	-	0.241	168	-

Anche in questo caso i risultati ottenuti variano molto poco al variare della complessità della rete, tranne che per le epoche necessarie per addestrare il modello. Rispetto al modello preliminare si osserva come la precisione sia lievemente inferiore pur restando molto buona, in particolare R² si mantiene su un valore di circa 0.70. Le prestazioni risultano inferiori rispetto al modello preliminare, ma bisogna tenere conto che, in questo caso la previsione è molto più complessa. Per questa analisi un buon compromesso risulta essere la configurazione a tre strati rispettivamente composti da 70,50 e 24 neuroni. Nella Figura 82: andamento dell'addestramento è rappresentato l'andamento dell'errore durante la fase di addestramento; si noti come sia per il Validation Set sia per il Training Set l'errore si mantenga simile durante tutto il processo e tenda a stabilizzarsi su un valore quasi coincidente negli step finali.

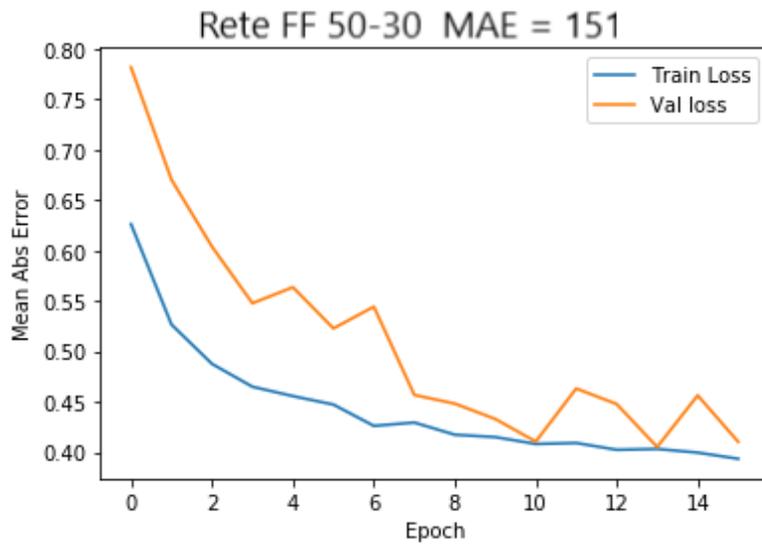


Figura 82: andamento dell'addestramento

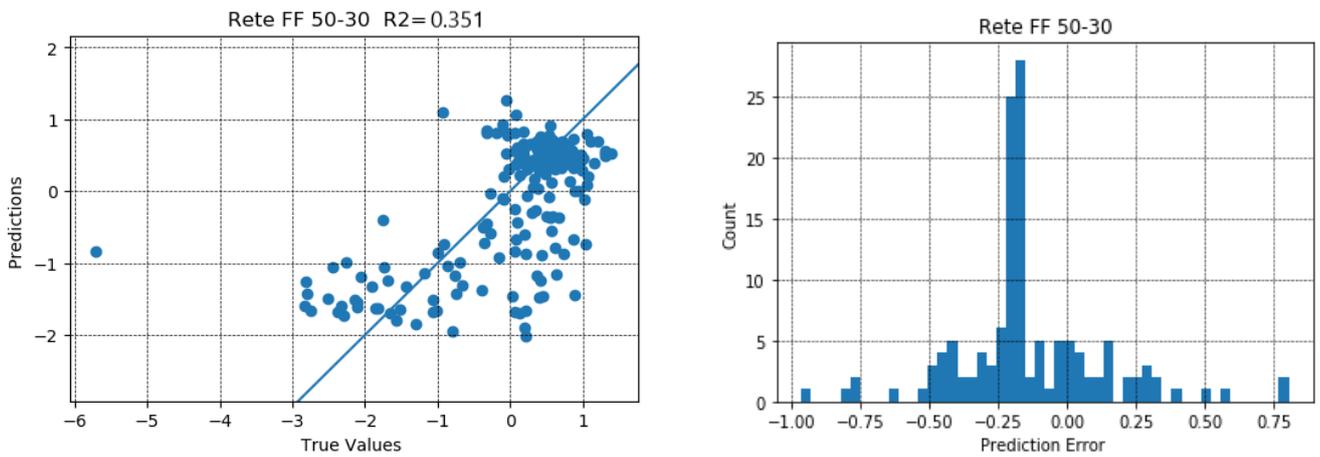


Figura 81: distribuzione dell'errore

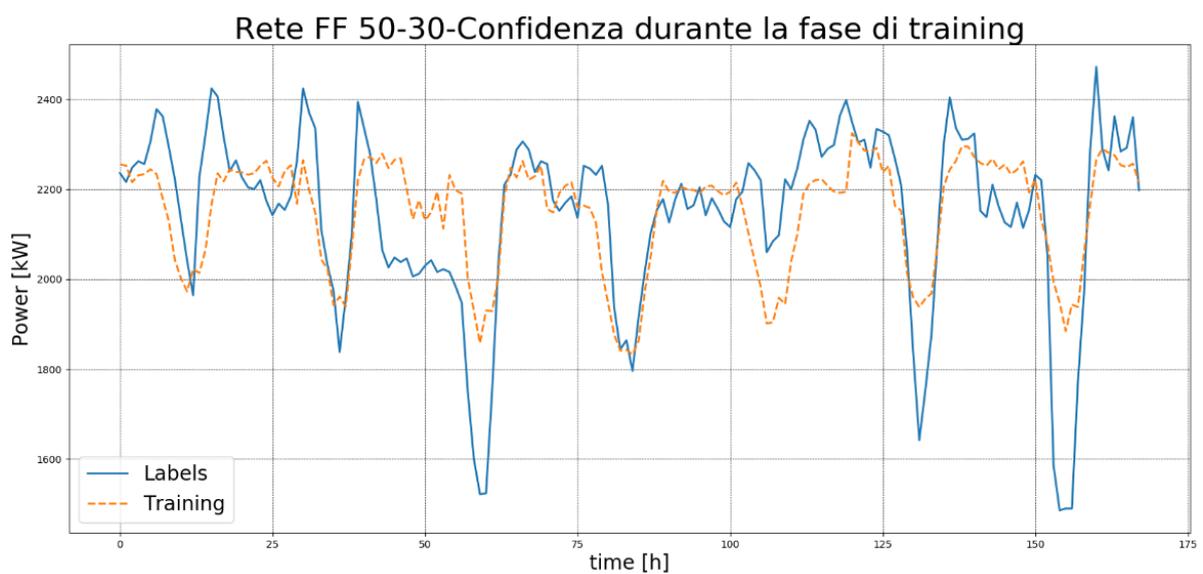


Figura 83: confidenza durante la fase di training

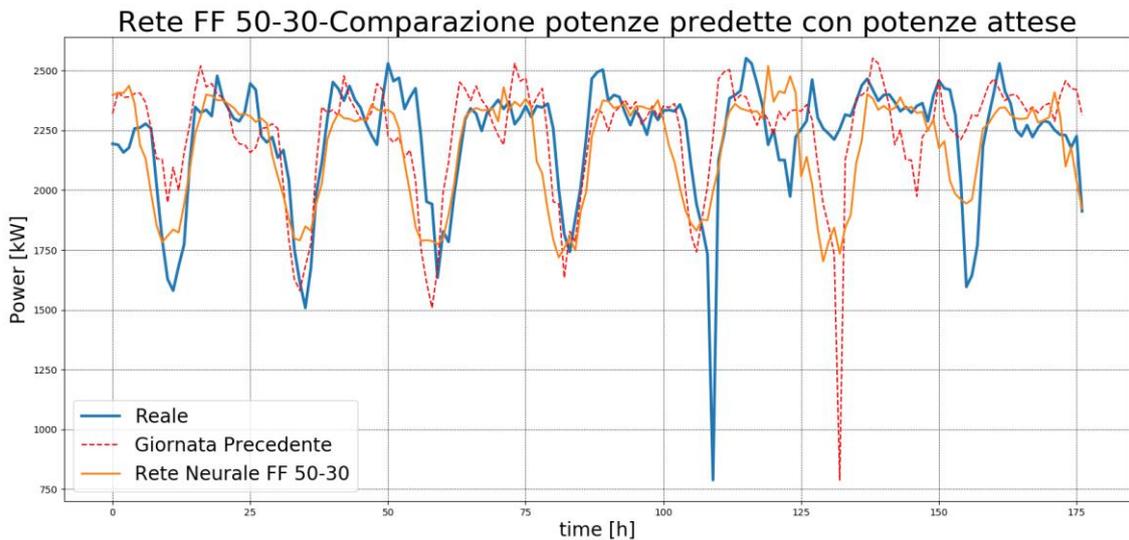


Figura 84: comparazione potenze previste con le potenze reali, rete FF giornaliera

Osservando la comparazione fra la curva delle previsioni e quella dei dati reali nel corso di una settimana nella Figura 55, si nota come la previsione di un'intera giornata sia meno precisa rispetto a alla previsione di una singola ora. In questo caso il modello non riesce a seguire fedelmente la curva di carico reale poiché essa è pesantemente influenzata dalla presenza dell'impianto fotovoltaico, la cui produzione dipende esclusivamente dalle condizioni atmosferiche.

3.4.6 Rete FF per la previsione settimanale

Creazione della rete

Una volta che i dati sono stati preparati adeguatamente si può procedere alla realizzazione della struttura della rete. Il secondo modello ha l'obiettivo di prevedere la potenza elettrica di un'intera settimana basandosi sulle misurazioni della settimana precedente. La rete può essere schematizzata come due blocchi separati composti dagli strati nascosti e dallo strato finali di output, i quali ricevono l'input, lo elaborano in base ai pesi delle connessioni e alla rispettiva funzione di attivazione dello strato. Fornendo quindi tutte le potenze nell'arco di una settimana $P_i(t)$ il modello restituisce la potenza prevista per la stessa ora nella settimana successiva $P_i(t+168)$.

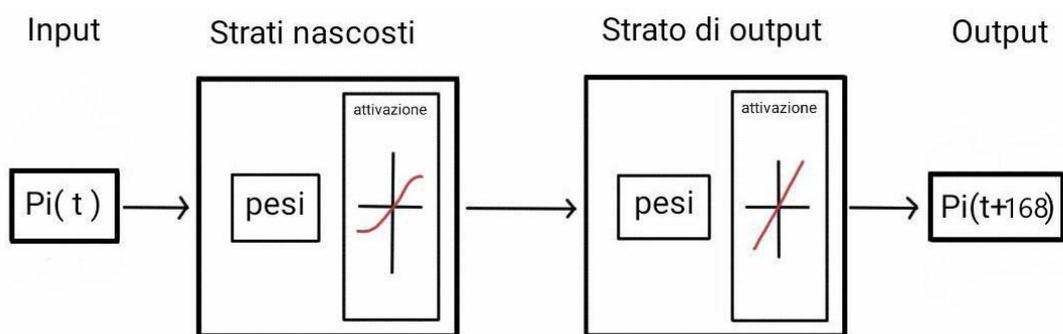


Figura 85: funzionamento della rete FF per le previsioni settimanale

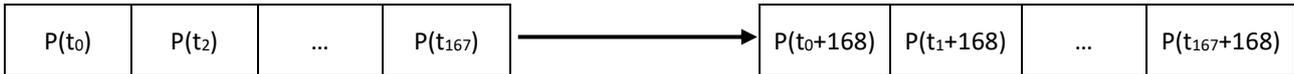
La sintassi per la creazione della struttura della rete è molto compatta. Il numero di nodi che compongono gli strati nascosti della rete può essere agevolmente per testare varie configurazioni e trovare quella più performante in base allo studio effettuato, mentre il numero di neuroni dello strato di output deve necessariamente essere pari al numero di output che si vuole ottenere, in questo caso 168 valori.

La struttura della rete è riassunta nella Tabella 11.

Tabella 26: capisaldi rete FF giornaliera

Rete	Layer	Tipo di layer	N° di nodi
Feed Foreward	Nascosto	Denso	70
	Nascosto	Denso	50
	Output	Denso	168

Il modello utilizza un set di 24 valori (tutte le ore di una giornata) e prevede simultaneamente 24 valori, corrispondenti alle 24 ore della giornata successiva, come schematizzato di seguito.



Si sono utilizzati due strati densi nascosti e uno di output, alla fine di ogni strato denso è stato applicato un layer con funzione dropout per prevenire fenomeni di overfitting. Gli strati nascosti utilizzano la funzione di attivazione *relu* (vedi 2.2.1). Successivamente si è definita la funzione per compilare la rete definendo i criteri di addestramento e valutazione. La funzione obiettivo utilizzata è l'errore quadratico medio (Mean Squared Error-MSE), poiché nelle previsioni regressive è un parametro significativo, ottimizzato attraverso l'algoritmo AdamOptimizer. La funzione metrica è utilizzata per visualizzare la progressione del processo di apprendimento è l'errore medio assoluto (Mean Absolute Error-MAE). In fine sono state definite le modalità di addestramento, in questo caso è stato impostato un valore limite di epoche molto alto, ma grazie all'implementazione di una funzione di Early Stopping il modello è in grado di fermare l'apprendimento nel momento in cui non si otterrebbero ulteriori miglioramenti, evitando di incorrere in fenomeni di overfitting.

Organizzazione dei dati

Per questo nuovo modello si utilizza lo stesso set di dati usato in precedenza, tuttavia le funzioni atte a preprocessare i dati variano leggermente. In questo caso, infatti, i dati in input e in output sono tutte e 168 le ore della settimana.

Risultati e prestazioni

Sono state provate varie configurazioni della rete variando il numero di nodi che compongono gli strati nascosti per valutare la configurazione più performante. In tutti i casi lo script è impostato per fermare automaticamente l'apprendimento una volta che la funzione Loss selezionata (in questo caso si è adoperato l'errore quadratico medio) non migliora dopo 5 epoche. I risultati relativi alle reti Feed Foreward sono illustrati nella Tabella 12. Data la natura stocastica del funzionamento delle reti neurali, per ogni configurazione è stato ripetuto il processo di training varie volte ottenendo risultati leggermente diversi ma con uno scostamento dal valore medio non significativo.

Tabella 27: prestazioni dell'addestramento rete LSTM settimanale

Struttura	Training		Validation		R ²	MAE [kW]	epoche
	MSE	MAE	MSE	MAE			
35-30	0.708	0.695	0.657	0.615	0.348	161	39
	0.713	0.684	0.657	0.618	0.351	157	43
	0.706	0.683	0.653	0.613	0.347	163	43
	0.702	0.684	0.634	0.614	0.348	161	41
50-30	0.488	0.456	0.522	0.495	0.413	143	16
	0.401	0.454	0.545	0.498	0.408	141	15
	0.486	0.451	0.529	0.493	0.410	148	15
	0.452	0.454	0.526	0.495	0.413	144	14
70-50	0.488	0.456	0.522	0.495	0.415	140	12
	0.475	0.454	0.530	0.498	0.420	136	12
	0.493	0.451	0.526	0.492	0.417	148	11
	0.481	0.454	0.520	0.405	0.413	142	14
100-60	0.482	0.458	0.523	0.482	0.416	139	10
	0.473	0.459	0.513	0.487	0.415	140	11
	0.481	0.450	0.526	0.481	0.414	144	10
	0.483	0.453	0.524	0.494	0.414	143	10
Settimana Precedente	-	-	-	-	0.465	131.2	-

In questo caso le prestazioni della rete risultano un poco inferiori rispetto ai casi precedenti, come si nota nella Figura 69 e nella Figura 70, il modello sottostima in maniera considerevole il carico elettrico. Anche tralasciando i picchi, che sono soddisfatti tramite la rete elettrica nazionale, il modello farebbe funzionare il cogeneratore ben al di sotto delle condizioni ottimali, diminuendo l'efficienza e, di conseguenza, l'erogazione dei titoli di Efficienza Energetica. Per le previsioni giornaliere risulta quindi più performante una rete di tipo Feed Forward.

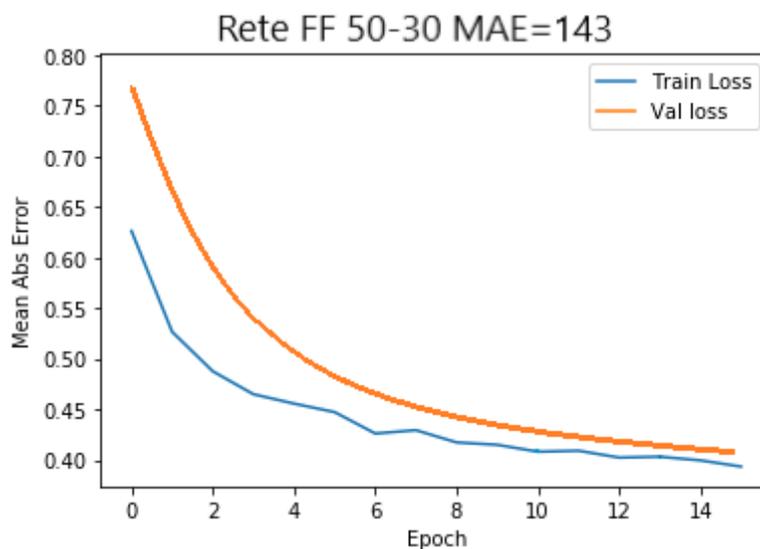


Figura 86: andamento dell'addestramento

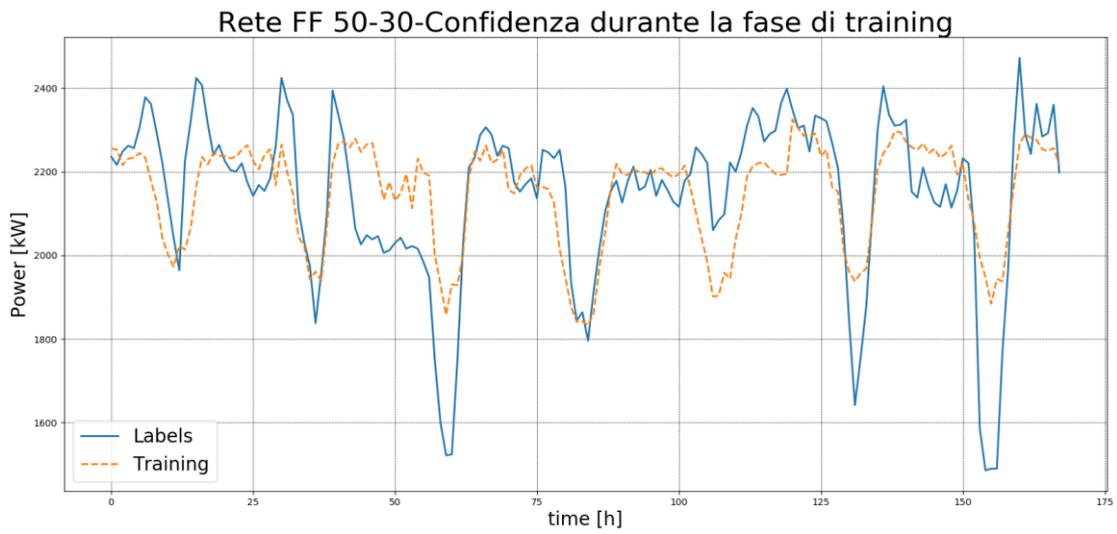


Figura 87: confidenza durante la fase di training

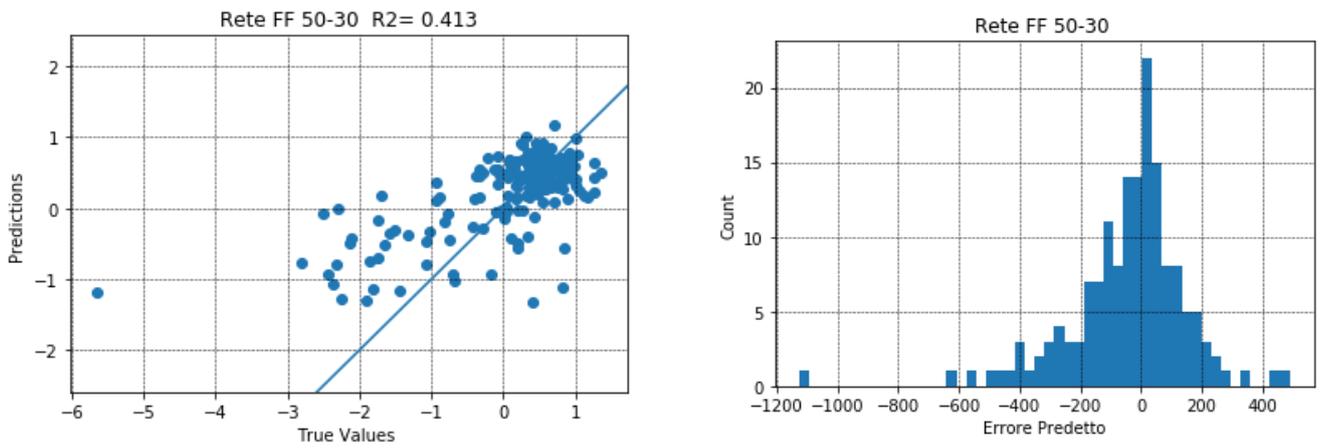


Figura 88: distribuzione dell'errore

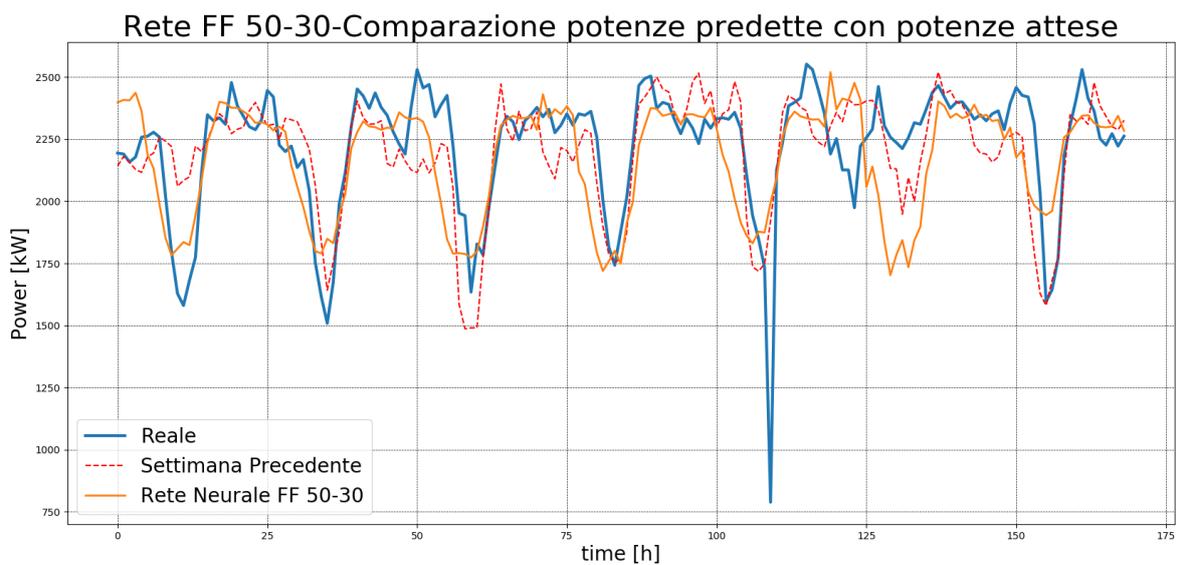


Figura 89: comparazione potenze previste con le potenze reali rete FF settimanale

3.4.7 Rete LSTM per la previsione oraria

Creazione della rete

Una volta che i dati sono stati preparati adeguatamente si può procedere alla realizzazione della struttura della rete. La rete può essere schematizzata come due blocchi separati composti dagli strati nascosti e dallo strato finali di output, i quali ricevono l'input, lo elaborano in base ai pesi delle connessioni e alla rispettiva funzione di attivazione dello strato. Fornendo quindi le potenze precedenti reali $P(t)$ il modello restituisce la potenza prevista per l'ora successiva $P'(t+1)$.

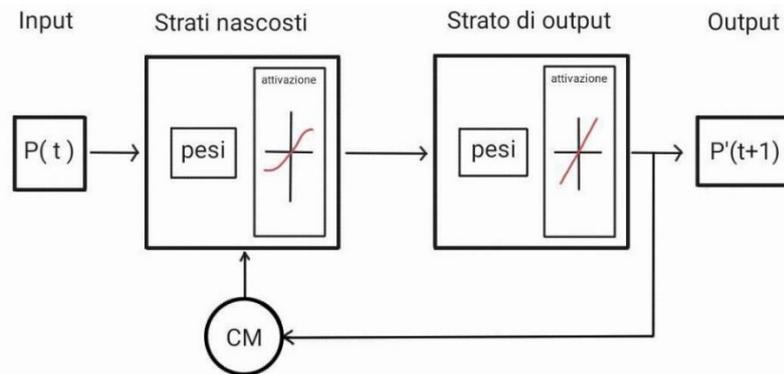


Figura 90: rete LSTM per la previsione oraria

La sintassi per la creazione della struttura della rete è molto compatta. Il numero di nodi che compongono gli strati nascosti della rete può essere agevolmente per testare varie configurazioni e trovare quella più performante in base allo studio effettuato, mentre il numero di neuroni dello strato di output deve necessariamente essere pari al numero di output che si vuole ottenere, in questo caso un solo valore.

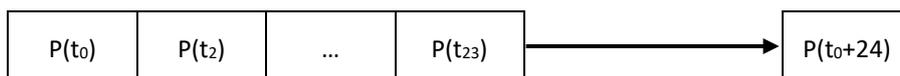
Si sono utilizzati due strati densi nascosti LSTM e uno di output di tipo Dense, alla fine di ogni strato è stato applicato un layer con funzione dropout per prevenire fenomeni di overfitting. Gli strati nascosti utilizzano la funzione di attivazione *relu* (vedi 2.2.1). Successivamente si è definita la funzione per compilare la rete definendo i criteri di addestramento e valutazione. La funzione obiettivo utilizzata è l'errore quadratico medio (Mean Squared Error-MSE), poiché nelle previsioni regressive è un parametro significativo, ottimizzato attraverso l'algoritmo AdamOptimizer. La funzione metrica è utilizzata per visualizzare la progressione del processo di apprendimento è l'errore medio assoluto (Mean Absolute Error-MAE).

La struttura della rete è riassunta nella Tabella 28.

Tabella 28: capisaldi rete LSTM oraria

Rete	Layer	Tipo di layer	N° di nodi
Long short Term Memory	Nascosto	CuDNNLSTM	25
	Nascosto	CuDNNLSTM	12
	Output	Denso	1

La finestra di dati utilizzata comprende le 24 ore precedenti al valore calcolato, come schematizzato di seguito.



In fine sono state definite le modalità di addestramento, in questo caso è stato impostato un valore limite di epoche molto alto, ma grazie all'implementazione di una funzione di Early Stopping il modello è in grado di fermare l'apprendimento nel momento in cui non si otterrebbero ulteriori miglioramenti, evitando di incorrere in fenomeni di overfitting.

Organizzazione dei dati

La fase di preparazione dei dati è analoga a quanto fatto nel paragrafo 3.3.3, con la creazione di una colonna contenente le potenze sfalsate di un'ora che fungerà da label in fase di addestramento.

Risultati e prestazioni

Sono state provate varie configurazioni della rete variando il numero di nodi che compongono gli strati nascosti per valutare la configurazione più performante (si sono utilizzate le stesse configurazioni adoperate nella rete Feed Forward per la previsione oraria al fine di permettere una migliore comparazione delle prestazioni fra le due tipologie di rete). In tutti i casi lo script è impostato per fermare automaticamente l'apprendimento una volta che la funzione Loss selezionata (in questo caso si è adoperato l'errore quadratico medio) non migliora dopo 5 epoche.

I risultati relativi alle reti Feed Forward sono illustrati nella Tabella 29. Data la natura stocastica del funzionamento delle reti neurali, per ogni configurazione è stato ripetuto il processo di training varie volte ottenendo risultati leggermente diversi ma con uno scostamento dal valore medio non significativo.

Tabella 29: prestazioni dell'addestramento rete LSTM per previsioni orarie

Struttura	Training		Validation		R ²	MAE [kW]	epoche
	MSE	MAE	MSE	MAE			
10-4	0.314	0.373	0.246	0.216	0.857	93.3	29
	0.308	0.374	0.248	0.213	0.858	91.0	31
	0.312	0.378	0.249	0.214	0.868	89.3	28
	0.311	0.372	0.241	0.246	0.874	83.4	33
25-12	0.276	0.359	0.211	0.174	0.893	53.2	13
	0.274	0.340	0.214	0.178	0.881	55.6	16
	0.276	0.354	0.213	0.175	0.893	52.8	18
	0.273	0.358	0.214	0.180	0.880	57.3	14
50-25	0.273	0.359	0.221	0.182	0.879	58.7	15
	0.276	0.354	0.224	0.184	0.870	61.0	14
	0.274	0.357	0.226	0.183	0.881	55.3	12
	0.273	0.353	0.220	0.182	0.870	62.8	16
100-50	0.277	0.349	0.224	0.188	0.893	52.5	13
	0.273	0.356	0.225	0.189	0.894	50.3	11
	0.274	0.354	0.227	0.190	0.880	57.4	12
	0.275	0.352	0.223	0.189	0.891	52.4	11
Ora Precedente	-	-	-	-	0.517	108	-

Nel caso delle reti LSTM le prestazioni non subiscono grandi oscillazioni al variare della complessità della rete, tranne che per il numero di epoche necessarie ad effettuare l'addestramento. Rispetto alle reti Feed Forward il valore di R² ha un tende ad avere un valore migliore (≈ 0.90 circa contro ≈ 0.80) anche se in entrambi i casi lo scarto fra le previsioni e i dati reali è molto contenuto in termini assoluti. Anche con l'utilizzo di questa tipologia di rete la configurazione 25-12 risulta essere il compromesso migliore, infatti andando ad aumentare il numero di nodi e la complessità della rete non si ottengono miglioramenti significativi, mentre la rete 10-4 ha prestazioni decisamente inferiori.

La Figura 58 mostra l'andamento della precisione del modello in funzione delle epoche durante la fase di addestramento, rispetto alla rete Feed Forward di pari complessità la rete LSTM impiega circa la metà delle epoche per stabilizzarsi sugli stessi valori di precisione.

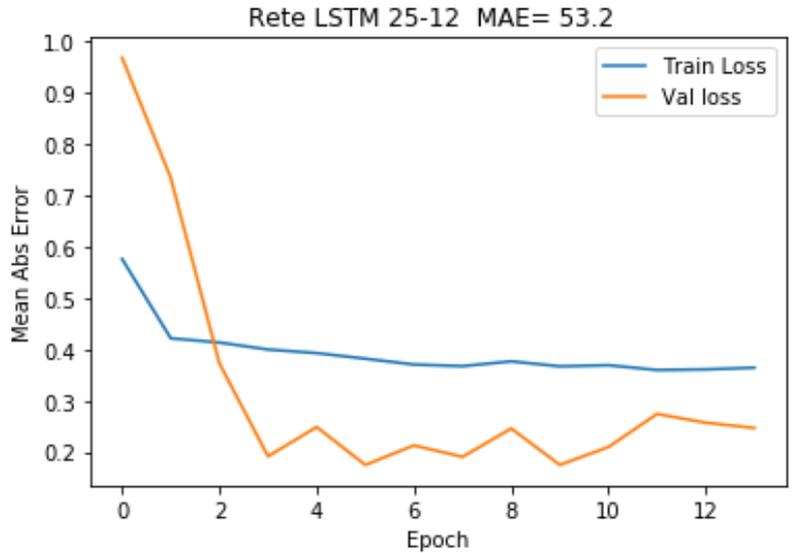


Figura 92: andamento dell'addestramento

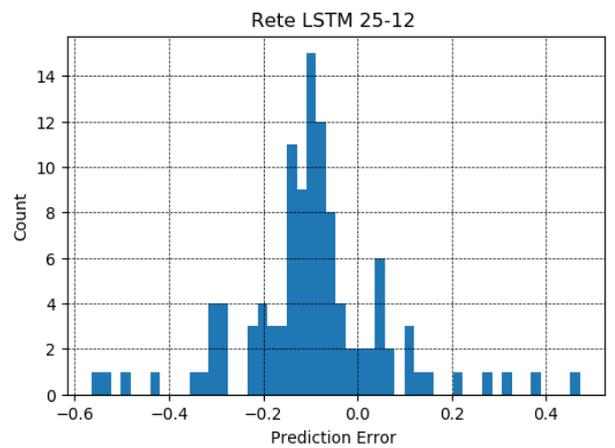
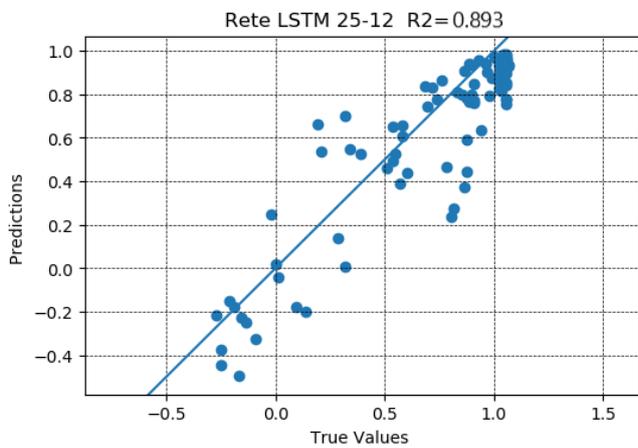


Figura 91: distribuzione errore

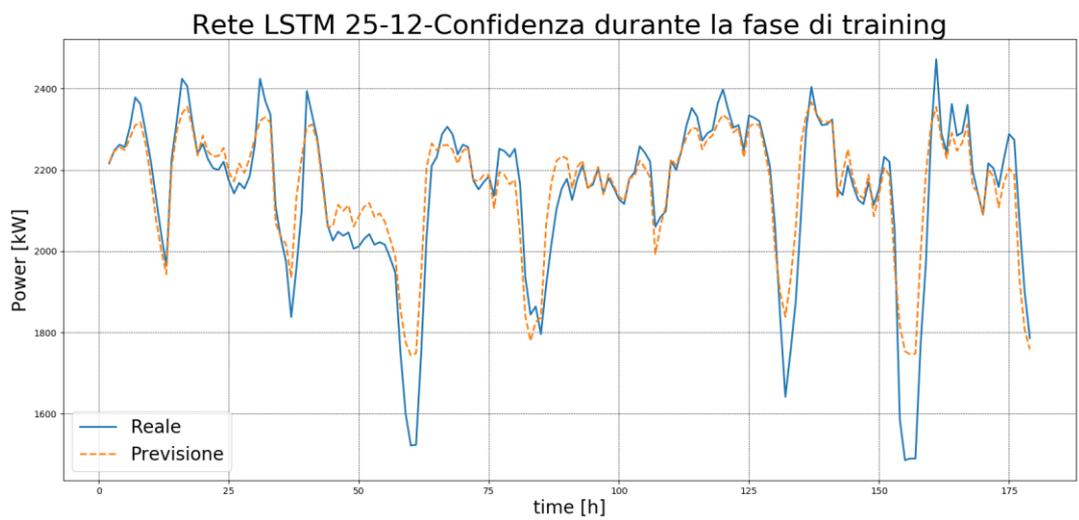


Figura 93: confidenza durante la fase di training

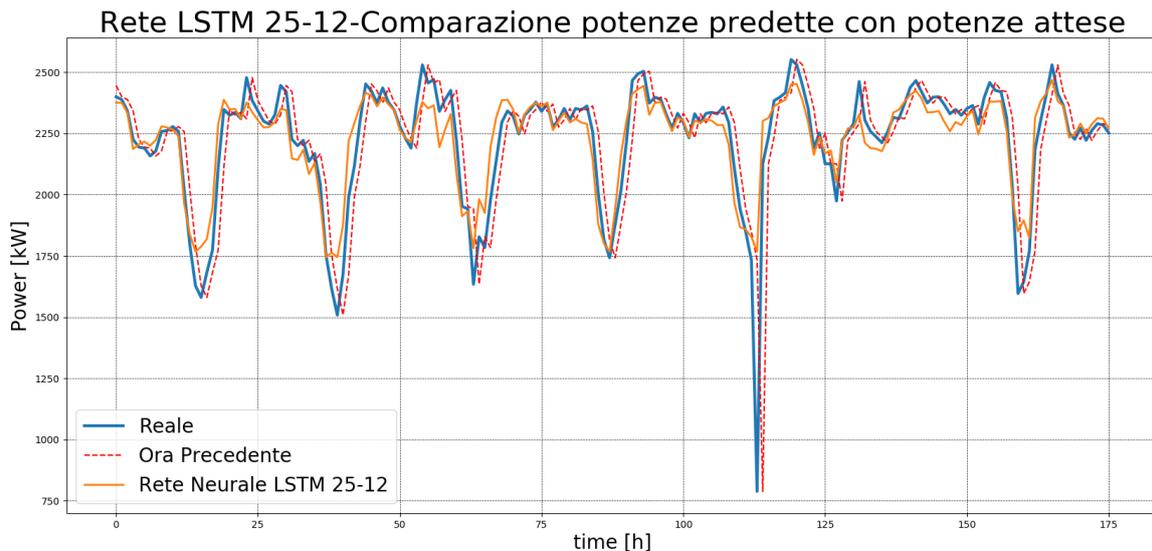


Figura 94: comparazione potenze previste con le potenze reali, rete LSTM oraria

La Figura 94 mostra il confronto fra la curva di carico reale e quella predetta dalla rete, si nota come anche per le reti LSTM il sistema fornisca una previsione che segue in maniera molto fedele l'andamento reale dei consumi durante una settimana, garantendo uno scarto nell'ordine di qualche decina di kW_{el}. Si nota anche come la curva delle previsioni abbia una varianza minore dei valori rispetto all'andamento reale, sottostimando i picchi di potenza e sovrastimando i punti di minimo.

3.4.8 Rete LSTM per la previsione giornaliera

Creazione della rete

Una volta che i dati sono stati preparati adeguatamente si può procedere alla realizzazione della struttura della rete. Il secondo modello ha l'obiettivo di prevedere la potenza elettrica di un'intera giornata, basandosi sui valori (reali) relativi alla giornata precedente.

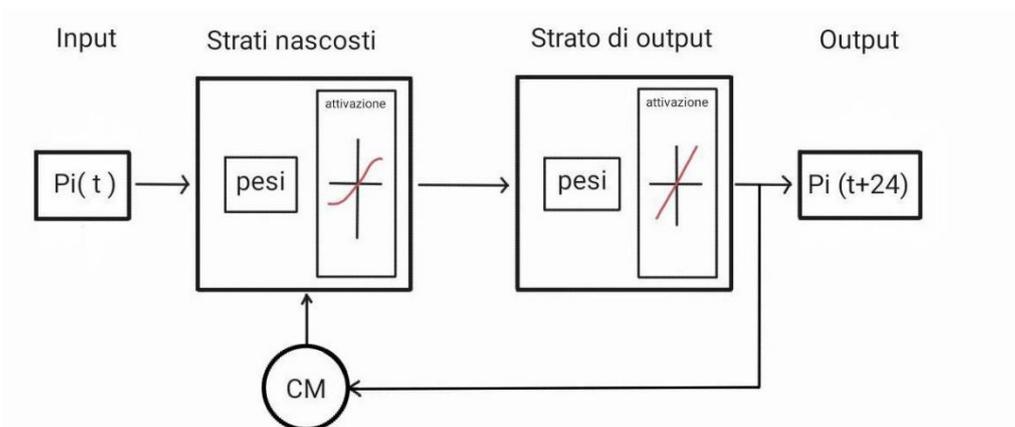


Figura 95: funzionamento della rete LSTM per le previsioni giornaliere

La rete può essere schematizzata come due blocchi rappresentanti rispettivamente gli strati nascosti e lo strato di output e da una cella di memoria (CM) che riceve le informazioni in output e le memorizza per confrontarle con l'input successivo. Fornendo la potenza attuale $P(t)$ la rete restituisce la potenza predetta per l'ora successiva $P'(t+1)$.

La sintassi per la creazione della struttura della rete è molto compatta. Il numero di nodi che compongono gli strati nascosti della rete può essere agevolmente per testare varie configurazioni e trovare quella più performante in base allo studio effettuato, mentre il numero di neuroni dello strato di output deve necessariamente essere pari al numero di output che si vuole ottenere, in questo caso 24 valori.

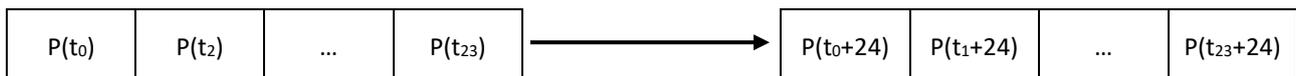
Si è utilizzati due strati nascosti, uno LSTM e uno denso, ed uno di output, alla fine di ogni strato nascosto è stato applicato un layer con funzione dropout ed uno di normalizzazione per prevenire fenomeni di overfitting. In questo modello le celle LSTM sono lievemente diverse rispetto a quelle utilizzate nel capitolo precedente: la struttura e le caratteristiche computazionali sono le medesime, ma permettono di utilizzare in maniera più efficiente l'accelerazione del processore grafico per mezzo dei CuDA cores. Gli strati nascosti utilizzano la funzione di attivazione *relu* (vedi 2.2.1). Successivamente si è definita la funzione per compilare la rete definendo i criteri di addestramento e valutazione. La funzione obiettivo utilizzata è l'errore quadratico medio (Mean Squared Error-MSE), poiché nelle previsioni regressive è un parametro significativo, ottimizzato attraverso l'algoritmo AdamOptimizer. La funzione metrica è utilizzata per visualizzare la progressione del processo di apprendimento è l'errore medio assoluto (Mean Absolute Error-MAE).

La struttura della rete è riassunta nella Tabella 30.

Tabella 30: capisaldi rete LSTM giornaliera

Rete	Layer	Tipo di layer	N° di nodi
Long short Term Memory	Nascosto	CuDNNLSTM	70
	Nascosto	Denso	50
	Output	Denso	1

Il modello utilizza un set di 24 valori (tutte le ore di una giornata) e prevede simultaneamente 24 valori, corrispondenti alle 24 ore della giornata successiva, come schematizzato di seguito.



In fine sono state definite le modalità di addestramento, in questo caso è stato impostato un valore limite di epoche molto alto, ma grazie all'implementazione di una funzione di Early Stopping il modello è in grado di fermare l'apprendimento nel momento in cui non si otterrebbero ulteriori miglioramenti, evitando di incorrere in fenomeni di overfitting.

Organizzazione dei dati

Per questo nuovo modello si utilizza lo stesso set di dati usato in precedenza, tuttavia le funzioni atte a preprocessare i dati variano leggermente.

Risultati e prestazioni

Sono state provate varie configurazioni della rete variando il numero di nodi che compongono gli strati nascosti per valutare la configurazione più performante. In tutti i casi lo script è impostato per fermare automaticamente l'apprendimento una volta che la funzione Loss selezionata (in questo caso si è adoperato l'errore quadratico medio) non migliori dopo 5 epoche.

I risultati relativi alle reti Feed Forward sono illustrati nella Tabella 31. Data la natura stocastica del funzionamento delle reti neurali, per ogni configurazione è stato ripetuto il processo di training varie volte ottenendo risultati leggermente diversi ma con uno scostamento dal valore medio non significativo.

Tabella 31: prestazioni dell'addestramento rete LSTM giornaliera

Struttura	Training		Validation		R ²	MAE [kW]	epoche
	MSE	MAE	MSE	MAE			
35-30	0.526	0.551	0.428	0.447	0.346	170	25
	0.523	0.548	0.425	0.444	0.346	171	24
	0.524	0.549	0.427	0.450	0.335	168	25
	0.519	0.558	0.428	0.451	0.346	173	28
50-30	0.477	0.493	0.402	0.415	0.465	130	9
	0.445	0.496	0.398	0.412	0.465	130	10
	0.442	0.496	0.401	0.414	0.464	134	9
	0.447	0.498	0.401	0.413	0.465	132	8
70-50	0.441	0.487	0.403	0.409	0.465	131	10
	0.445	0.490	0.405	0.405	0.471	120	8
	0.444	0.498	0.411	0.410	0.465	133	10
	0.447	0.496	0.402	0.407	0.455	132	11
100-60	0.452	0.496	0.426	0.429	0.455	130	11
	0.453	0.497	0.426	0.426	0.455	131	10
	0.458	0.488	0.428	0.432	0.464	135	8
	0.444	0.493	0.426	0.428	0.455	132	10
Giornata Precedente	-	-	-	-	0.241	168	-

Come nel capitolo precedente, la rete LSTM per la previsione oraria fornisce dei risultati significativamente inferiori rispetto alla controparte Feed Forward; il valore di R² si attesta su circa il 25% e l'errore medio nell'ordine dei 100 kW_{el}. Come si nota nella Figura 99, il modello tende a fornire buoni risultati per quanto riguarda i valori medi, ma non è in grado di prevedere in maniera efficace eventuali cali nella produzione da parte del cogeneratore poiché esso è condizionato dalla produzione fotovoltaica, la quale dipende esclusivamente da fattori climatici e non dalla produzione dello stabilimento.

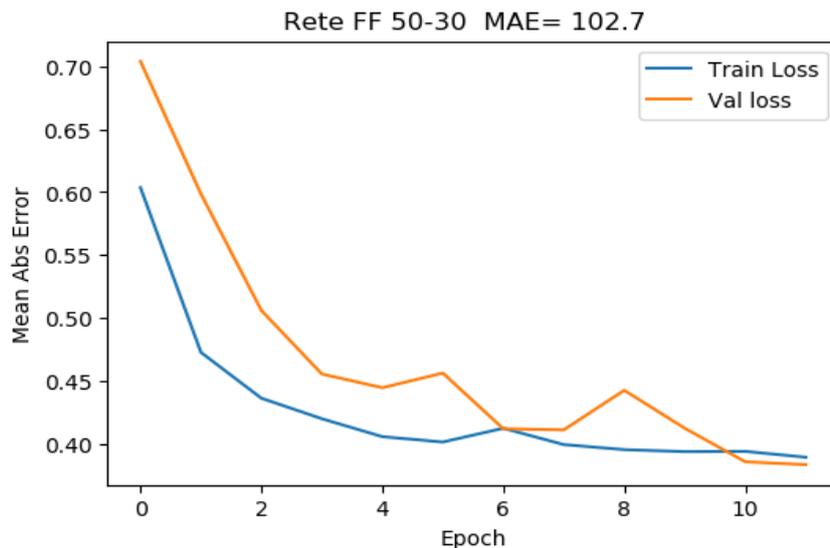


Figura 96: andamento dell'addestramento

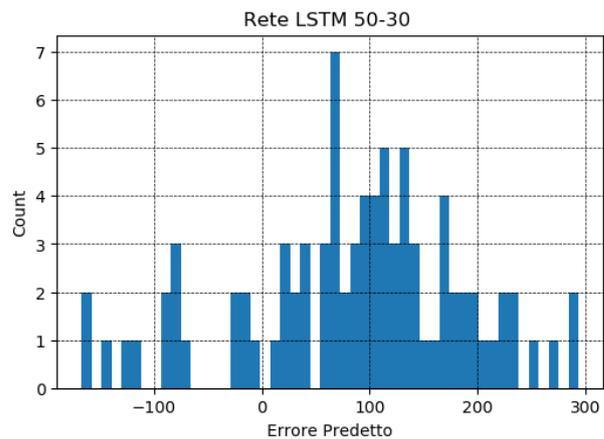
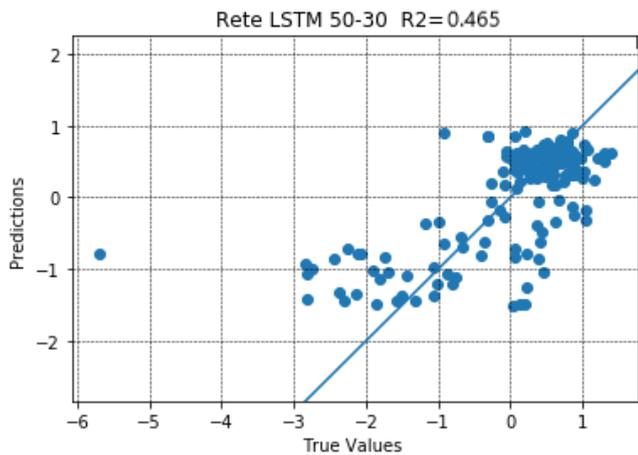


Figura 97: distribuzione errore

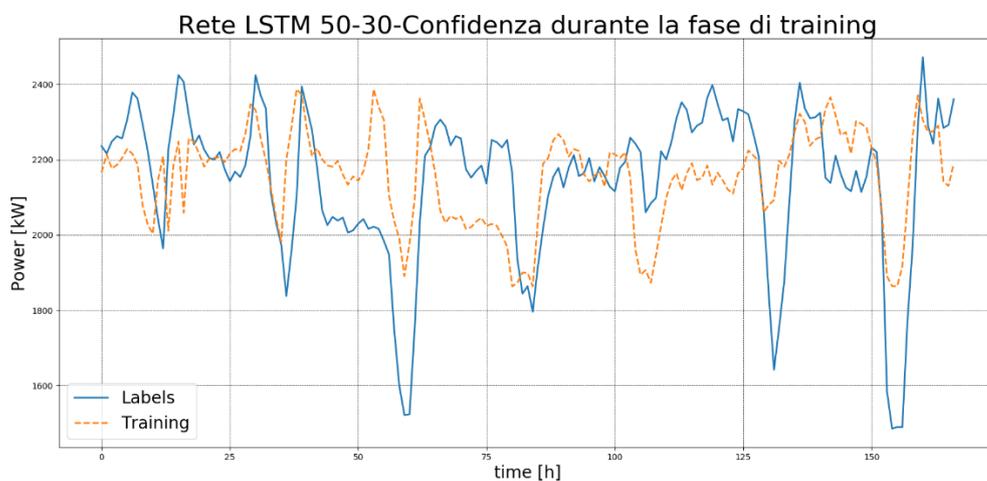


Figura 98: confidenza durante la fase di training

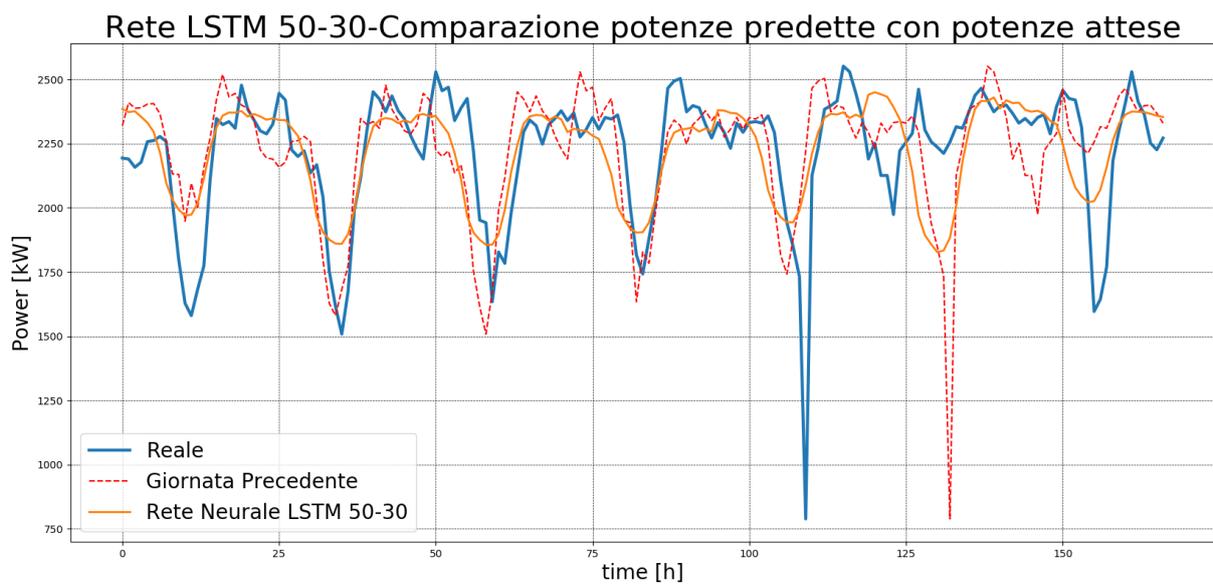


Figura 99: comparazione potenze previste con le potenze reali, rete LSTM giornaliera

3.4.9 Rete LSTM per la previsione settimanale

Creazione della rete

Una volta che i dati sono stati preparati adeguatamente si può procedere alla realizzazione della struttura della rete. Il secondo modello ha l'obiettivo di prevedere la potenza elettrica nell'ora successiva basandosi sul valore (reale) precedente.

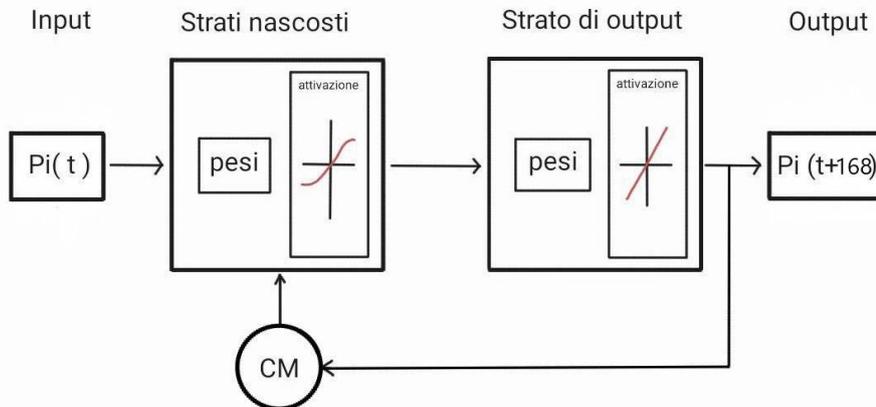


Figura 100: funzionamento della rete LSTM per le previsioni settimanali

La rete può essere schematizzata come due blocchi rappresentanti rispettivamente gli strati nascosti e lo strato di output e da una cella di memoria (CM) che riceve le informazioni in output e le memorizza per confrontarle con l'input successivo. Fornendo la potenza di un'intera settimana $P_i(t)$ la rete restituisce la potenza predetta per la settimana successiva $P'(t+1)$.

La sintassi per la creazione della struttura della rete è molto compatta. Il numero di nodi che compongono gli strati nascosti della rete può essere agevolmente per testare varie configurazioni e trovare quella più performante in base allo studio effettuato, mentre il numero di neuroni dello strato di output deve necessariamente essere pari al numero di output che si vuole ottenere, in questo caso 168 valori.

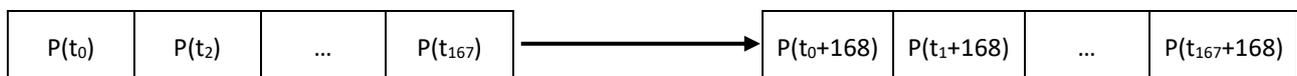
Si sono utilizzati due strati densi nascosti e uno di output, alla fine di ogni strato denso è stato applicato un layer con funzione dropout per prevenire fenomeni di overfitting. Gli strati nascosti utilizzano la funzione di attivazione *relu* (vedi 2.2.1). Successivamente si è definita la funzione per compilare la rete definendo i criteri di addestramento e valutazione. La funzione obiettivo utilizzata è l'errore quadratico medio (Mean Squared Error-MSE), poiché nelle previsioni regressive è un parametro significativo, ottimizzato attraverso l'algoritmo AdamOptimizer. La funzione metrica è utilizzata per visualizzare la progressione del processo di apprendimento è l'errore medio assoluto (Mean Absolute Error-MAE).

La struttura della rete è riassunta nella Tabella 18.

Tabella 32: capisaldi rete LSTM settimanale

Rete	Layer	Tipo di layer	N° di nodi
Long Short Term Memory	Nascosto	LSTM	50
	Nascosto	LSTM	30
	Output	Denso	168

Il modello utilizza un set di 24 valori (tutte le ore di una giornata) e prevede simultaneamente 24 valori, corrispondenti alle 24 ore della giornata successiva, come schematizzato di seguito.



Infine sono state definite le modalità di addestramento, in questo caso è stato impostato un valore limite di epoche molto alto, ma grazie all'implementazione di una funzione di Early Stopping il modello è in grado di

fermare l'apprendimento nel momento in cui non si otterrebbero ulteriori miglioramenti, evitando di incorrere in fenomeni di overfitting.

Organizzazione dei dati

Per questo nuovo modello si utilizza lo stesso set di dati usato in precedenza, tuttavia le funzioni atte a preprocessare i dati variano leggermente.

Risultati e prestazioni

Sono state provate varie configurazioni della rete variando il numero di nodi che compongono gli strati nascosti per valutare la configurazione più performante. In tutti i casi lo script è impostato per fermare automaticamente l'apprendimento una volta che la funzione Loss selezionata (in questo caso si è adoperato l'errore quadratico medio) non migliori dopo 5 epoche.

I risultati relativi alle reti LSTM sono illustrati nella Tabella 19. Data la natura stocastica del funzionamento delle reti neurali, per ogni configurazione è stato ripetuto il processo di training varie volte ottenendo risultati leggermente diversi ma con uno scostamento dal valore medio non significativo.

Tabella 33: prestazioni dell'addestramento rete LSTM settimanale

Struttura	Training		Validation		R ²	MAE [kW]	epoche
	MSE	MAE	MSE	MAE			
35-30	0.708	0.695	0.657	0.615	0.348	161	40
	0.710	0.684	0.654	0.618	0.351	157	43
	0.708	0.683	0.623	0.613	0.347	163	43
	0.702	0.684	0.634	0.614	0.348	161	42
50-30	0.488	0.456	0.522	0.495	0.409	145	16
	0.401	0.444	0.545	0.496	0.408	144	15
	0.476	0.458	0.529	0.493	0.400	150	16
	0.452	0.454	0.526	0.495	0.402	147	14
70-50	0.488	0.451	0.522	0.495	0.415	150	12
	0.475	0.454	0.531	0.498	0.421	156	12
	0.493	0.459	0.526	0.492	0.416	157	13
	0.481	0.454	0.520	0.405	0.414	152	14
100-60	0.482	0.458	0.523	0.482	0.416	149	11
	0.473	0.459	0.583	0.483	0.415	151	11
	0.483	0.450	0.526	0.486	0.413	155	10
	0.467	0.453	0.524	0.494	0.415	152	10
Settimana Precedente	-	-	-	-	0.465	131.2	-

In questo caso le prestazioni della rete risultano un poco inferiori rispetto ai casi precedenti, come si nota nella Figura 69 e nella Figura 70, il modello sottostima in maniera considerevole il carico elettrico. Anche tralasciando i picchi, che sono soddisfatti tramite la rete elettrica nazionale, il modello farebbe funzionare il cogeneratore ben al di sotto delle condizioni ottimali, diminuendo l'efficienza e, di conseguenza, l'erogazione dei titoli di Efficienza Energetica. Per le previsioni giornaliere risulta quindi più performante una rete di tipo Feed Forward.

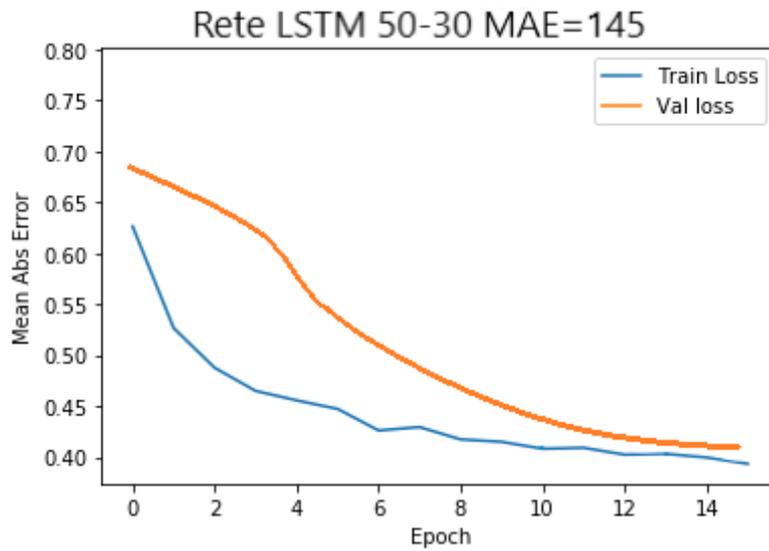


Figura 101: andamento dell'addestramento

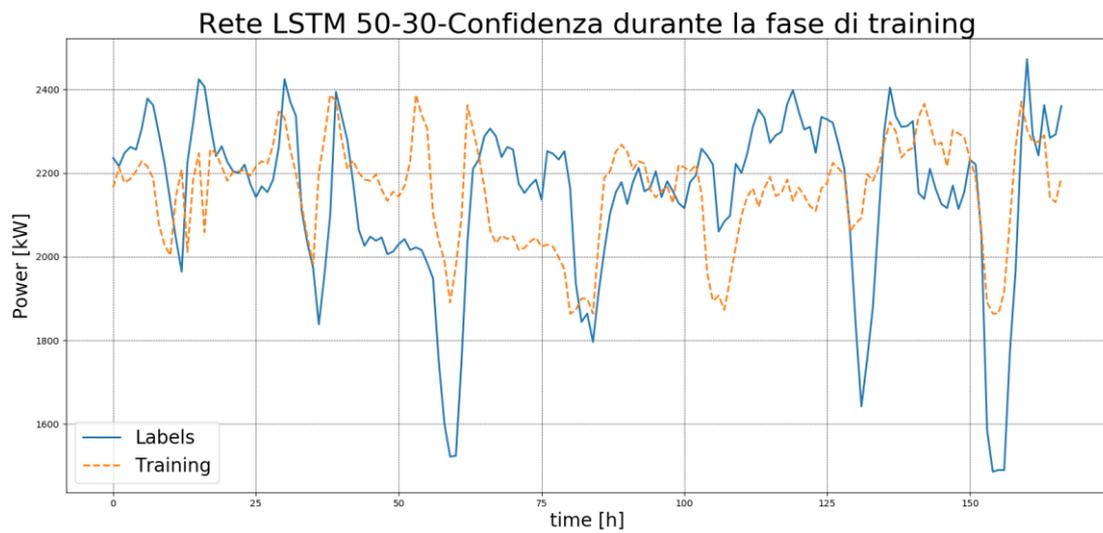


Figura 102: confidenza durante la fase di training

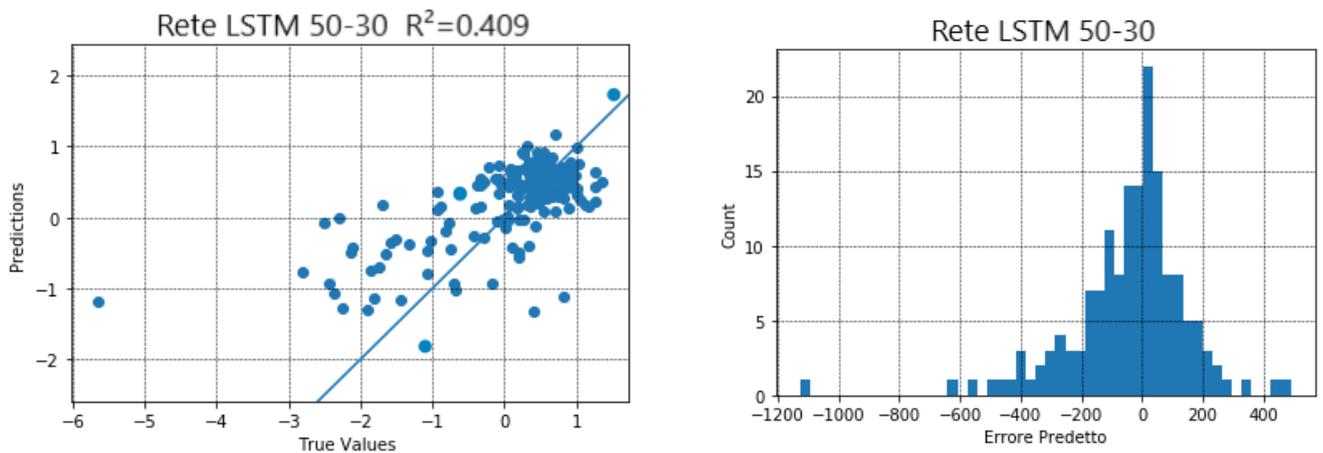


Figura 103: distribuzione dell'errore

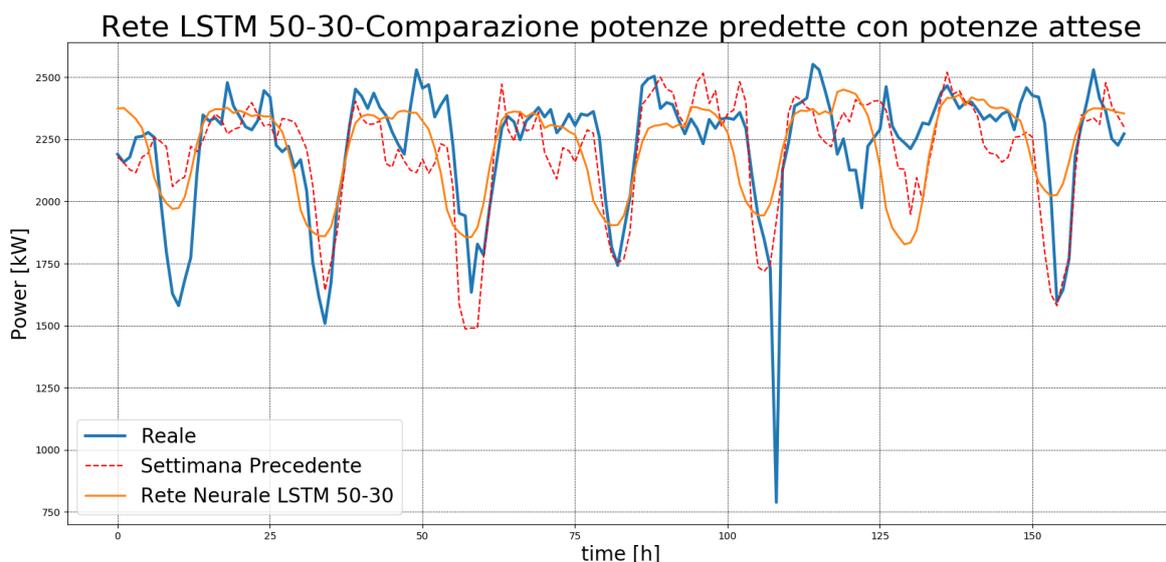


Figura 104: comparazione potenze previste con le potenze reali rete LSTM settimanale

3.4.10 Confronto fra i modelli

Lo sviluppo di modelli previsionali basati sulle reti neurali ha prodotto dei buoni risultati nello studio di questo stabilimento. Lo sviluppo in parallelo di due tipologie di rete (Feed Forward e Long Short Term Memory) ha fornito risultati molto simili, ed entrambe costituiscono una valida opzione per questa tipologia di problemi. Di seguito sono riportati i valori medi delle prestazioni per ogni rete e paragonati ai risultati ottenibili effettuando una semplice traslazione dei consumi precedenti coerentemente con l'intervallo di previsione adottata (un'ora, una giornata e una settimana).

Tabella 34: valori medi degli indici di prestazione

Previsione oraria			Previsione giornaliera			Previsione settimanale		
Rete	R ²	MAE [kW]	Rete	R ²	MAE [kW]	Rete	R ²	MAE [kW]
FF	0,855	48	FF	0,348	151	FF	0,415	142
LSTM	0,887	58	LSTM	0,465	131	LSTM	0,412	144
Rispetto all'ora precedente	0,517	108	Rispetto al giorno precedente	0,241	168	Rispetto alla settimana precedente	0,465	131

Anche in questo secondo caso i modelli di previsione oraria garantiscono ottimi risultati ma la loro reale utilità è limitata dal breve orizzonte temporale, risultando comunque significativamente migliori di una semplice traslazione rispetto all'ora precedente.

In questo secondo stabilimento la produzione è continua durante tutto l'arco della settimana, di conseguenza i risultati ottenuti dai modelli di previsione giornaliera e settimanale risultano molto simili. Tutti i modelli studiati forniscono prestazioni inferiori rispetto al caso precedente poiché non è stata implementata una grandezza che tenga conto della produzione da parte dell'impianto fotovoltaico.

L'esperienza maturata nello sviluppo di questi modelli ha consentito di elaborare un modello più complesso in grado di valutare la produzione del cogeneratore tenendo conto anche di una stima della produzione fotovoltaica.

3.4.11 Rete LSTM per previsione settimanale con input esogeni

La caratteristica principale di questo impianto è la presenza simultanea di un motore cogenerativo e un impianto fotovoltaico per la generazione di energia elettrica. La produzione di energia da parte dell'impianto fotovoltaico dipende esclusivamente dalle condizioni metereologiche del sito, le quali non sono controllabili, questo comporta una produzione estremamente discontinua e non regolabile. Per questo motivo l'energia generata dal parco fotovoltaico ha la priorità rispetto a quella prodotta tramite l'impianto di cogenerazione, il quale viene regolato in base alla produzione da fonte rinnovabile. La conseguenza di questo accoppiamento è che il profilo di carico del cogeneratore (la grandezza che si vuole prevedere tramite la rete neurale artificiale) risulta molto irregolare e dipendente sia dalla produzione sia da condizioni esterne. Ne risulta che lo sviluppo di un modello che tenesse in considerazione esclusivamente il carico elettrico del cogeneratore, come nel caso precedente, potrebbe risultare inadeguato; si è dunque ricercato quali possano essere le variabili atmosferiche che influenzino in modo diretto la produzione fotovoltaica e, di conseguenza, la produzione del cogeneratore. Di seguito nella Figura 105 sono rappresentate alcune grandezze in relazione alla produzione del cogeneratore. Ne risulta che l'unica grandezza strettamente correlata con la curva di carico del cogeneratore è l'irradianza solare, eventuali correlazioni con temperatura atmosferica o copertura delle nubi risultano molto deboli e di conseguenza non sono stati inseriti fra le variabili analizzate dalla rete neurale.

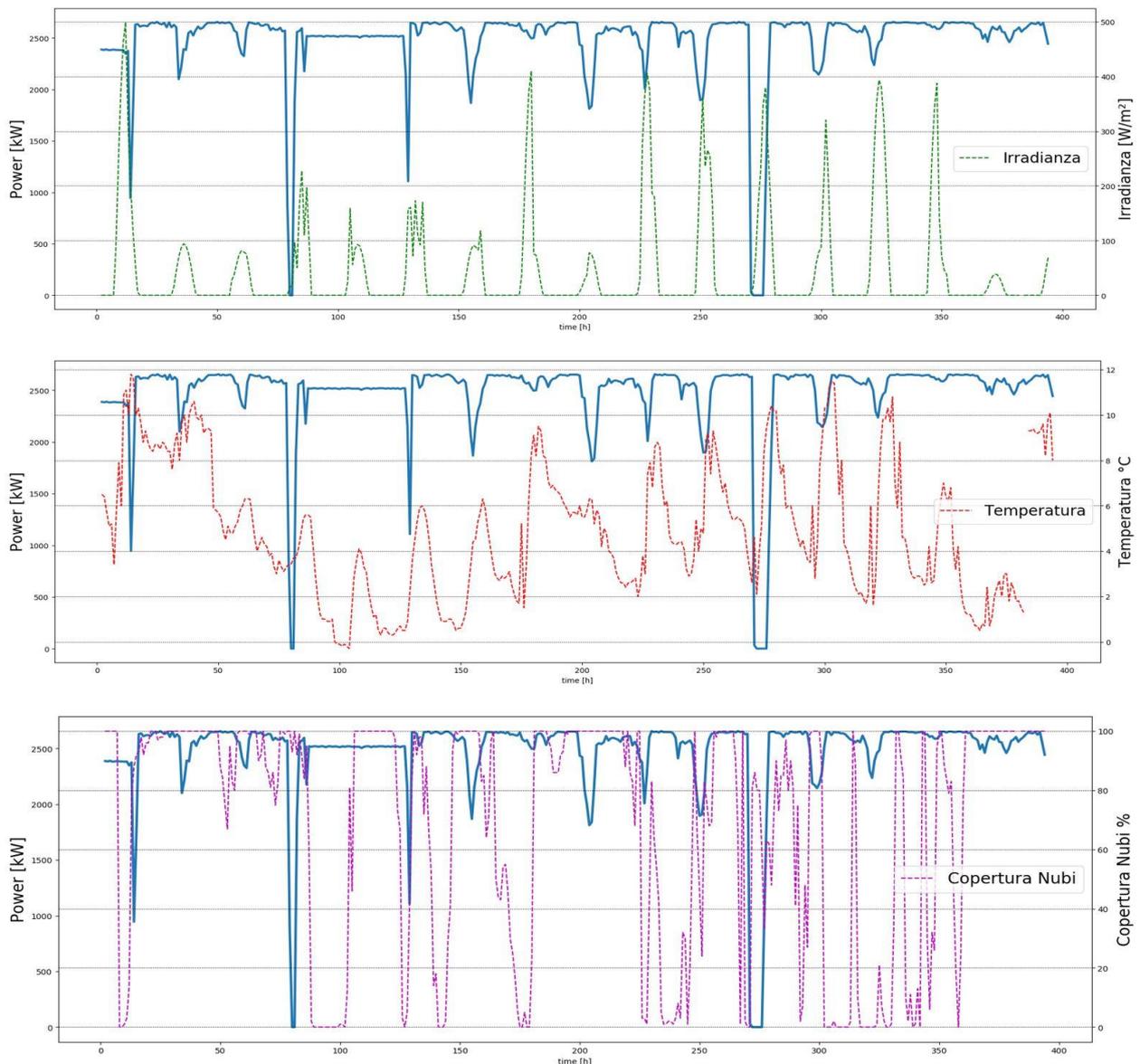


Figura 105: variabili atmosferiche in relazione alla produzione del cogeneratore

Poiché il cogeneratore è dimensionato per fornire una quantità di energia inferiore al fabbisogno totale dello stabilimento, al fine di poter lavorare il più possibile in condizioni nominali e garantire un rendimento massimo, la restante parte dell'energia richiesta è soddisfatta tramite l'impianto fotovoltaico e la rete elettrica nazionale. Per questo motivo le produzioni del cogeneratore e del fotovoltaico non sono strettamente complementari e quindi anche la relazione che lega la produzione da cogenerazione e l'irradianza solare risulta spesso debole, come si può notare in Figura 105.

Non potendo correlare in maniera stretta la produzione del cogeneratore con le condizioni climatiche esterne si è sviluppato un modello in grado di prevedere in un primo momento la produzione elettrica da parte del parco fotovoltaico e successivamente di correlare quest'ultima con la produzione del cogeneratore.

Questo modello opera secondo i seguenti passaggi:

- Raccolta delle previsioni metereologiche, in particolare dell'Irradianza solare. Esistono numerosi siti che consentono di scaricare queste informazioni grazie a specifiche API;
- Elaborazione delle irradianze previste tramite un'apposita rete neurale per prevedere la produzione dell'impianto fotovoltaico;
- Elaborazione delle produzioni fotovoltaiche previste e dei dati di produzione del cogeneratore (reali) tramite una rete neurale che fornisce come output la previsione della produzione del cogeneratore.

Di seguito il processo è schematizzato tramite il diagramma di flusso sottostante.

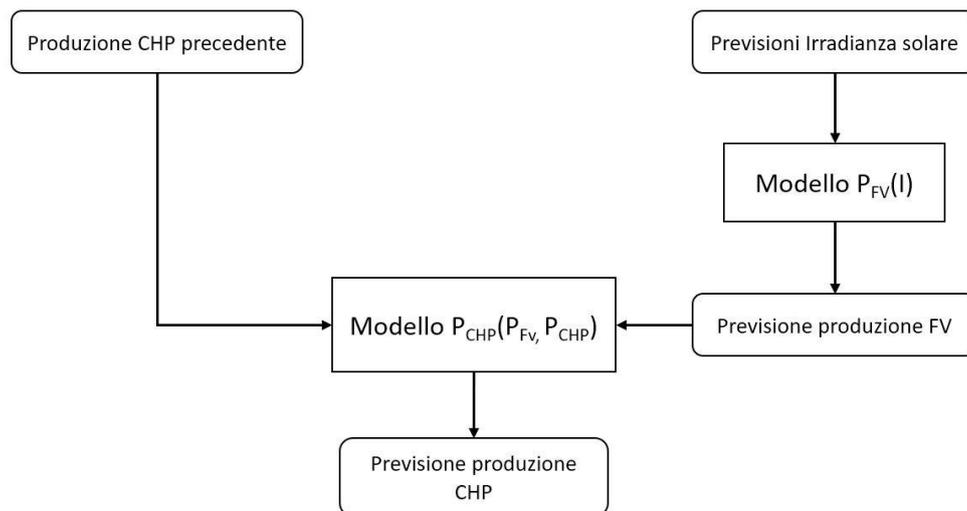


Figura 106: diagramma del modello con input esogeni

3.4.11.1 Costruzione del modello con input esogeni utilizzando dati storici

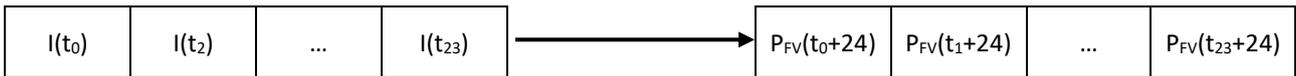
Rete neurale per la previsione della produzione fotovoltaica

La costruzione di un modello in grado di effettuare la previsione della sola produzione da fotovoltaico relativa alla giornata successiva può essere effettuata agevolmente, tuttavia occorre considerare che i dati che verrebbero utilizzati per effettuare la previsione (temperatura, pressione atmosferica, copertura delle nubi) sono anch'essi delle previsioni frutto a loro volta di un modello e quindi l'incertezza risulterebbe significativa. Si è comunque provato a sviluppare un modello di previsione analogo a quello utilizzato per i cogeneratori, basandosi sui dati climatici storici.

Il modello è stato addestrato a prevedere la produzione elettrica da parte del campo fotovoltaico basandosi su questi parametri. È stata realizzata una rete Feed Forward simile a quelle studiate in precedenza, le cui caratteristiche sono riassunte nella Tabella 35.

Tabella 35: capisaldi rete FF per la previsione del fotovoltaico

Rete	Layer	Tipo di layer	N° di nodi
Feed Forward	Nascosto	Denso	64
	Nascosto	Denso	32
	Output	Denso	24



I risultati, mostrati in Figura 107, risultano essere soddisfacenti, tuttavia occorre tenere conto che sono basati sui dati *storici reali*. Si osserva come la curva di carico della produzione elettrica da parte di un impianto fotovoltaico abbia un andamento molto diverso rispetto a quella ottenibile con un impianto tradizionale: si ha un picco di produzione nelle ore centrali della giornata e una produzione nulla la sera.

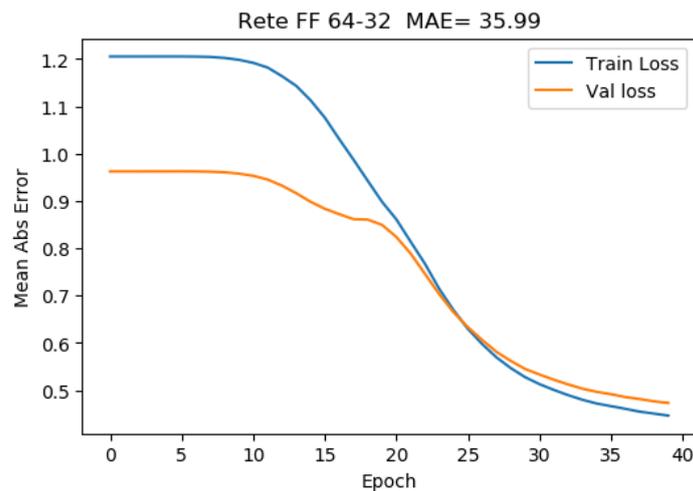


Figura 108: fase di training

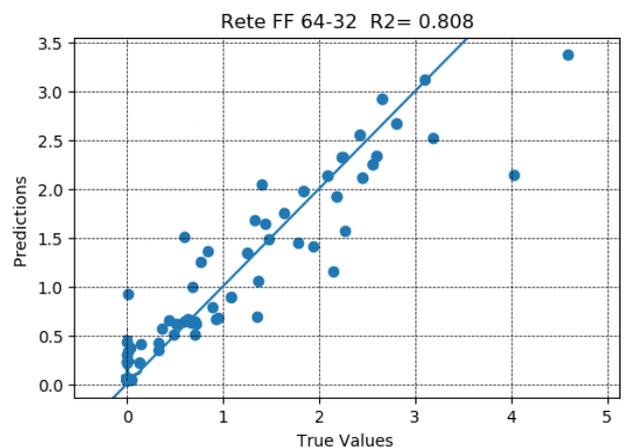
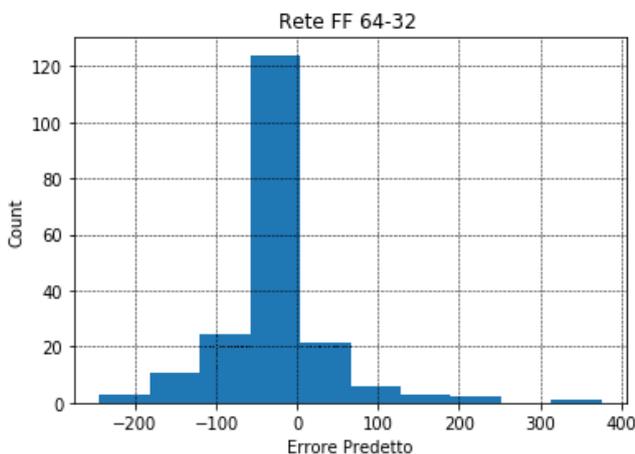


Figura 107: distribuzione dell'errore

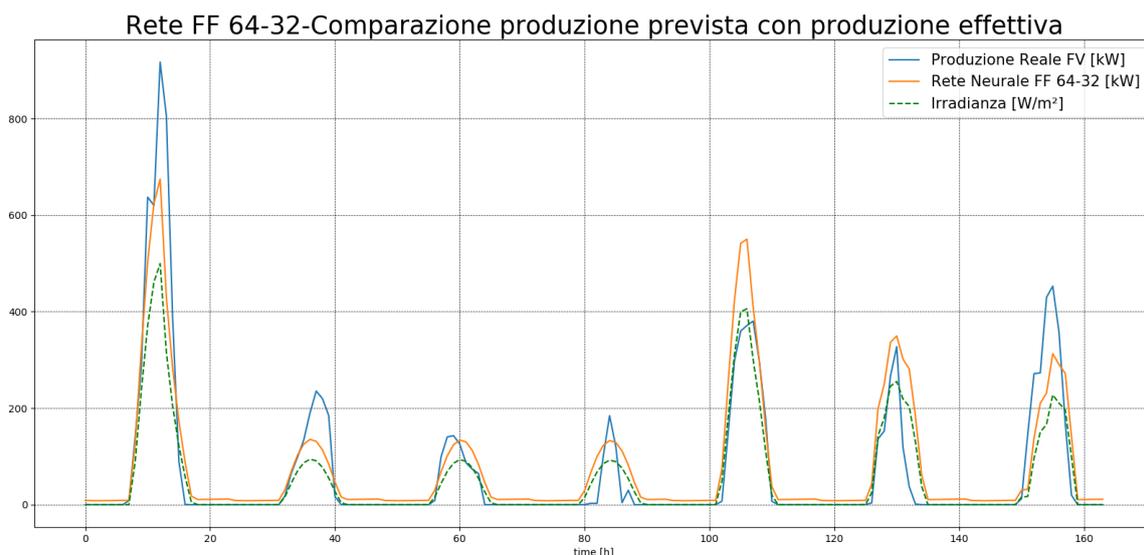


Figura 109: risultati previsione produzione FV

La difficoltà principale nello sviluppo ed utilizzo del modello deriva dal fatto che, anche in due giorni consecutivi, la produzione varia in maniera significativa (si guardi la seconda giornata presa in esame nella Figura 107).

Rete per la previsione della potenza del cogeneratore con input esogeni

Le previsioni relative alla produzione fotovoltaica vengono salvati ed utilizzati da un secondo modello, analogo ai modelli trattati in precedenza, addestrato a prevedere la produzione dell'impianto cogenerativo mettendo in relazione:

- Dato orario;
- Produzione dell'impianto fotovoltaico;
- Produzione del cogeneratore.

La struttura della rete è riassunta nella Tabella 36.

Tabella 36: capisaldi della rete LSTM con input esogeni per la previsione del cogeneratore

Rete	Layer	Tipo di layer	N° di nodi
Feed Forward	Nascosto	LSTM	50
	Nascosto	LSTM	30
	Output	Denso	24



Il modello utilizza un tensore di valori contenenti i dati della produzione del cogeneratore relativi ad una determinata giornata e i dati relativi alla produzione del fotovoltaico relativi al giorno successivo, in modo da poter poi utilizzare le previsioni atmosferiche e prevede simultaneamente 24 valori, corrispondenti alle 24 ore della giornata successiva, come schematizzato di seguito. Anche in questo caso sono state provate varie configurazioni della rete per testarne le prestazioni, riportate nella Tabella 37.

Tabella 37: prestazioni rete LSTM con input esogeni

Struttura	Training		Validation		R ²	MAE [kW]	epoche
	MSE	MAE	MSE	MAE			
35-30	0.356	0.254	0.729	0.747	0.376	150	56
	0.334	0.281	0.725	0.731	0.375	151	54
	0.324	0.290	0.754	0.750	0.365	148	54
	0.299	0.247	0.720	0.755	0.375	153	58
50-30	0.377	0.201	0.602	0.615	0.519	114	32
	0.348	0.196	0.598	0.612	0.519	115	30
	0.354	0.195	0.607	0.614	0.518	110	29
	0.302	0.198	0.610	0.613	0.519	115	33
70-50	0.376	0.168	0.613	0.635	0.519	114	29
	0.323	0.196	0.601	0.621	0.518	111	27
	0.341	0.201	0.599	0.604	0.518	110	29
	0.322	0.211	0.602	0.616	0.519	116	28
100-60	0.376	0.146	0.604	0.624	0.519	115	26
	0.302	0.190	0.616	0.617	0.519	113	26
	0.344	0.221	0.611	0.610	0.519	113	23
	0.317	0.210	0.594	0.602	0.519	112	25
Senza input esogeni	-	-	-	-	0.465	131	-

Come prevedibile i risultati in termini di epoche per l'addestramento e gli altri parametri caratteristici sono molto simili ai modelli precedenti, tuttavia, come illustrato nella Figura 112, si ottiene una previsione apprezzabilmente migliore per quanto concerne le riduzioni di carico dovute alla produzione fotovoltaica. In questo caso, infatti, il modello si adatta in maniera molto più efficace alle fluttuazioni dovute alla produzione dell'impianto fotovoltaico.

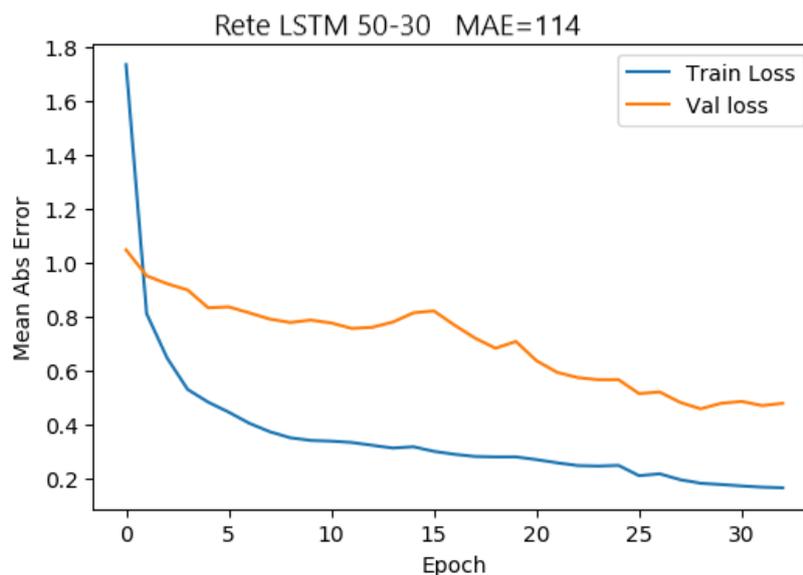


Figura 110: andamento della fase di training

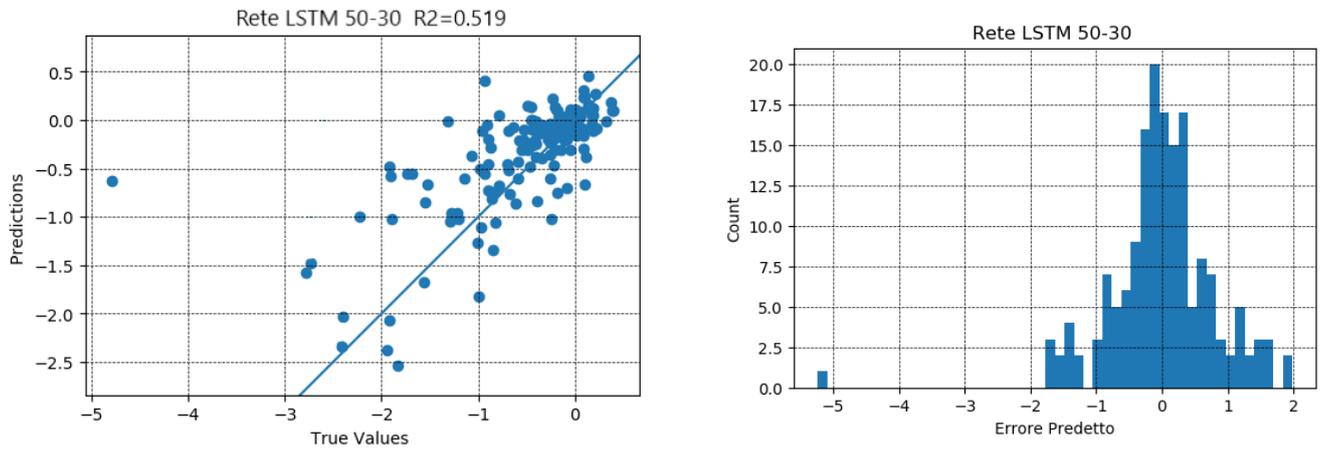


Figura 111: distribuzione dell'errore

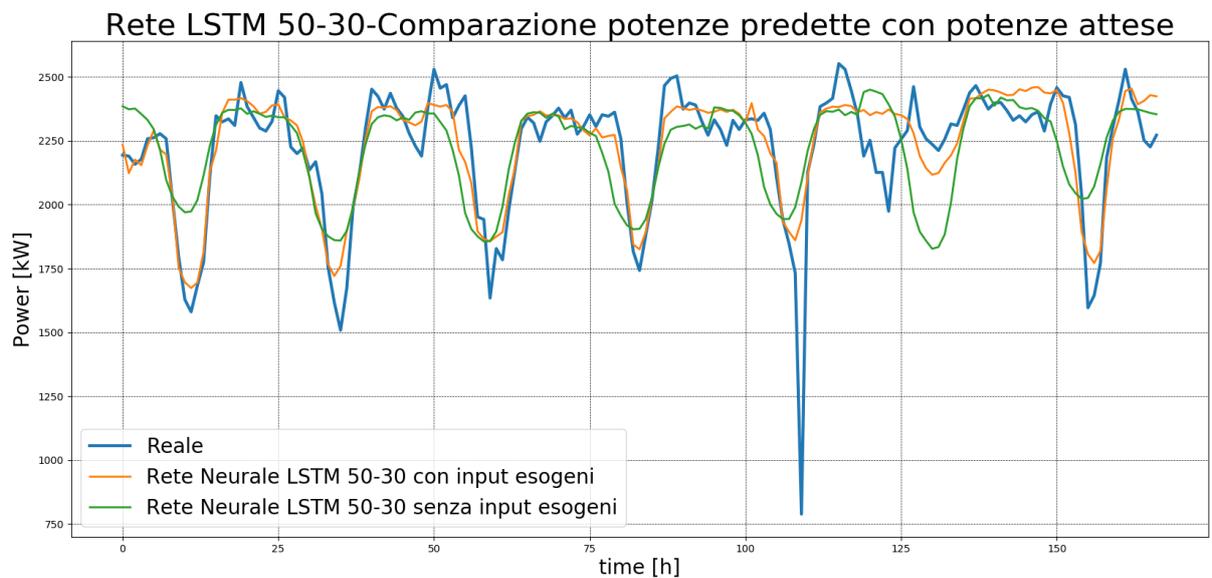


Figura 112: comparazione potenze previste con le potenze reali, rete LSTM giornaliera con input esogeni

La nuova struttura è stata applicata alle tipologie di rete già viste in precedenza, di seguito, sono riportati i confronti fra i modelli con e senza input esogeni.

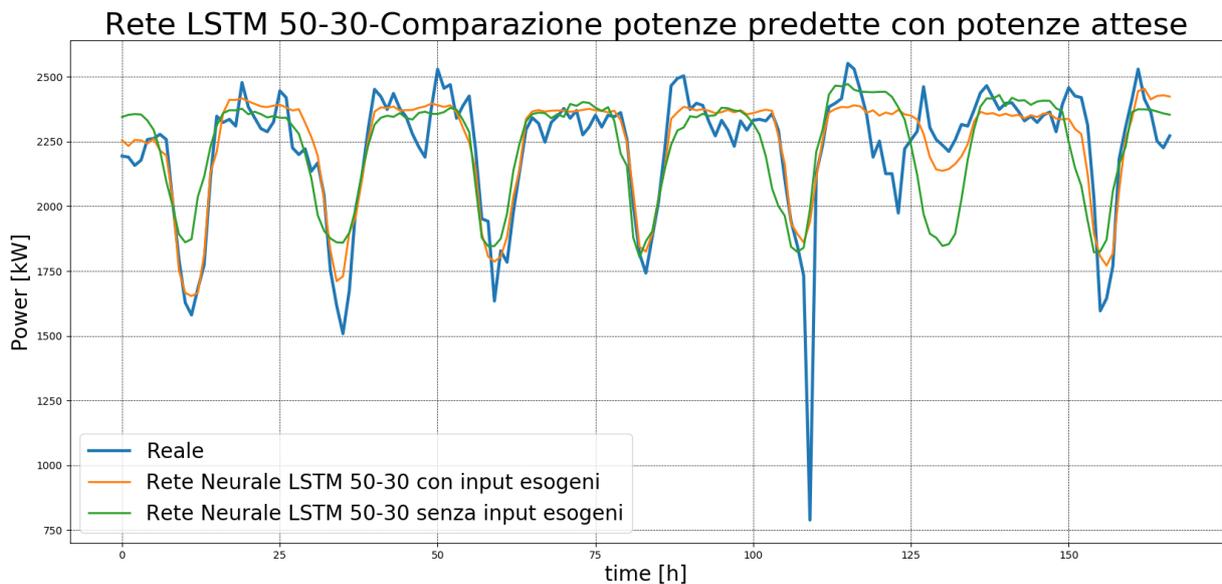


Figura 113: comparazione potenze previste con potenze reali, rete LSTM settimanale con input esogeni

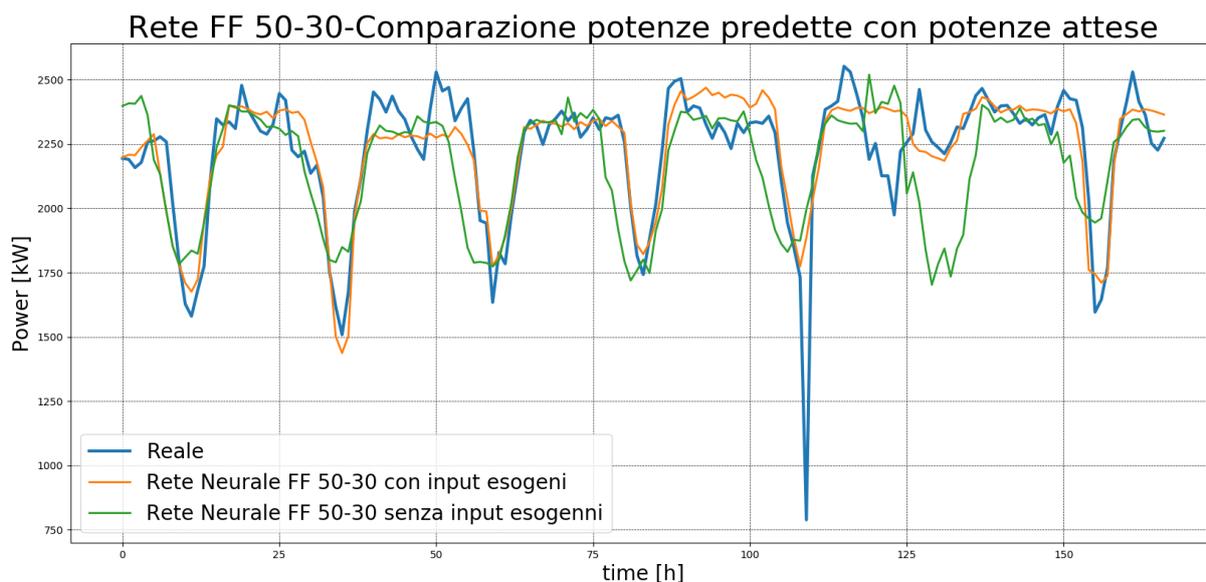


Figura 114: comparazione potenze previste con le potenze reali, rete FF giornaliera con input esogeni

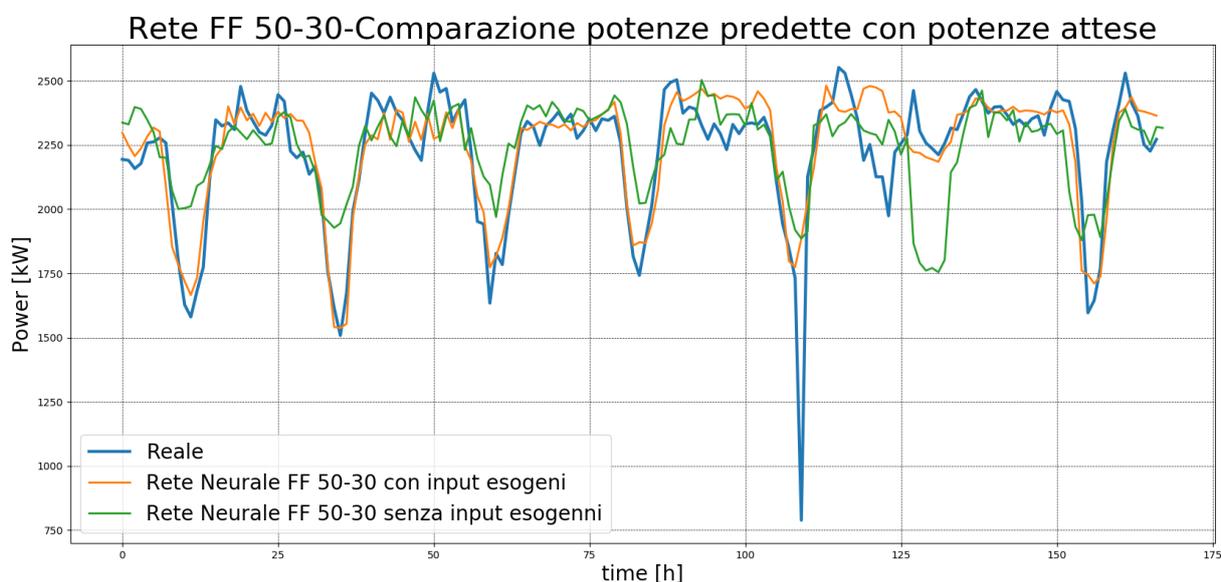


Figura 115: comparazione potenze previste con le potenze reali, rete FF settimanale con input esogeni

Le prestazioni dei modelli sono riassunte nella Tabella 38: prestazioni reti con input esogeni Tabella 38.

Tabella 38: prestazioni reti con input esogeni

Previsione giornaliera			Previsione settimanale		
Rete	R ²	MAE [kW]	Rete	R ²	MAE [kW]
FF	0,488	127	FF	0,501	123
LSTM	0,519	113	LSTM	0,534	101

Anche in questo caso si osserva come le prestazioni ottenute con reti FF e LSTM siano molto simili e, per questo stabilimento, il modello per la previsione di un'intera settimana e il modello per la previsione di una giornata forniscano risultati molto simili.

3.4.11.2 Rete con previsioni metereologiche

Una volta appurato che il modello modificato con input esogeni funziona, si è voluto testare il funzionamento dello stesso operando in una situazione equivalente a quella di un uso effettivo, utilizzando quindi delle vere previsioni climatiche e non lo storico dei dati come nella fase di addestramento. Le previsioni metereologiche relative all'irradianza sono state scaricate dal sito www.solcast.com.au.

Primo set di previsioni metereologiche (22/02/2019 - 27/02/2019)

Si sono selezionati i giorni dal 22/02/2019 a 27/02/2019 e ogni giorno è stata scaricata la previsione di irradianza del sito dove è locato l'impianto fotovoltaico e successivamente le previsioni sono state utilizzate per predire la produzione elettrica del parco fotovoltaico, di seguito i risultati della stima.

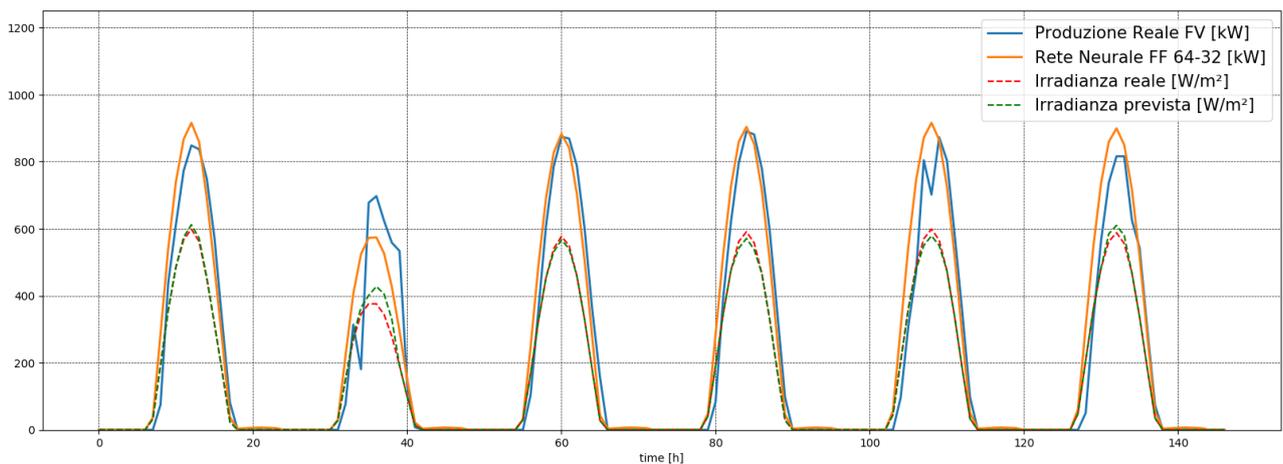


Figura 116: previsione produzione fotovoltaica basata su previsioni metereologiche

Come prevedibile il risultato è molto buono con uno scostamento medio di meno di 45 kW_e rispetto al totale. L'intervallo preso in esame è in grado di fornire un buon test poiché nella giornata del 23/03 si è verificata un'apprezzabile copertura nuvolosa (efficacemente prevista dal fornitore del servizio meteorologico) che ha ridotto significativamente la produzione fotovoltaica e di conseguenza utile a valutare se il modello per la previsione della produzione del cogeneratore sia effettivamente in grado di adeguarsi di conseguenza. I risultati della previsione fotovoltaica sono stati usati come uno degli input per la previsione del cogeneratore.

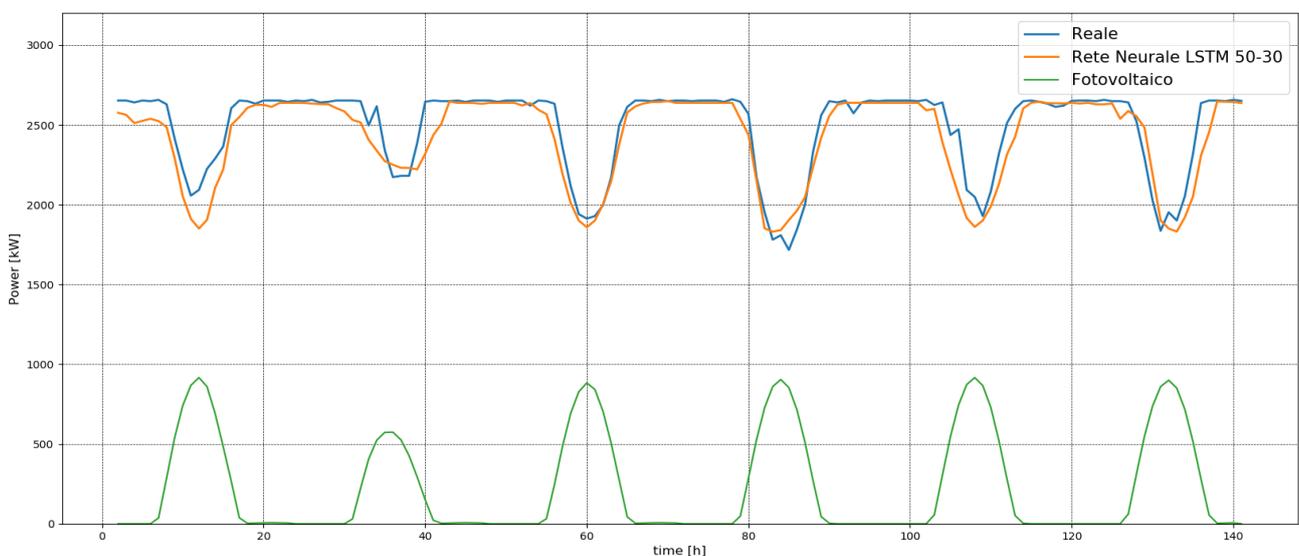


Figura 117: previsione della produzione del cogeneratore basata su previsioni metereologiche

Nella Figura 117 sono mostrati i risultati della previsione del cogeneratore rispetto alla produzione fotovoltaica (stimata). La curva di previsione segue in maniera fedele l'andamento reale, in particolare si nota come il modello sia in grado di prevedere efficacemente la diminuzione di produzione del fotovoltaico aumentando quella del cogeneratore. Il valore di R^2 si attesta 0.573 con un MAE di circa 100 kW_{el} . Nell'arco di tempo analizzato si nota anche come il profilo di carico risulti molto regolare nelle ore in cui non interviene il fotovoltaico; nelle giornate analizzate in precedenza l'andamento risultava maggiormente irregolare e di conseguenza la previsione risultava globalmente peggiore.

Secondo set di previsioni metereologiche (05/03/2019 - 10/03/2019)

Si sono selezionati i giorni dal 22/02/2019 a 27/02/2019 e ogni giorno è stata scaricata la previsione di irradianza del sito dove è locato l'impianto fotovoltaico e successivamente le previsioni sono state utilizzate per predire la produzione elettrica del parco fotovoltaico, di seguito i risultati della previsione.

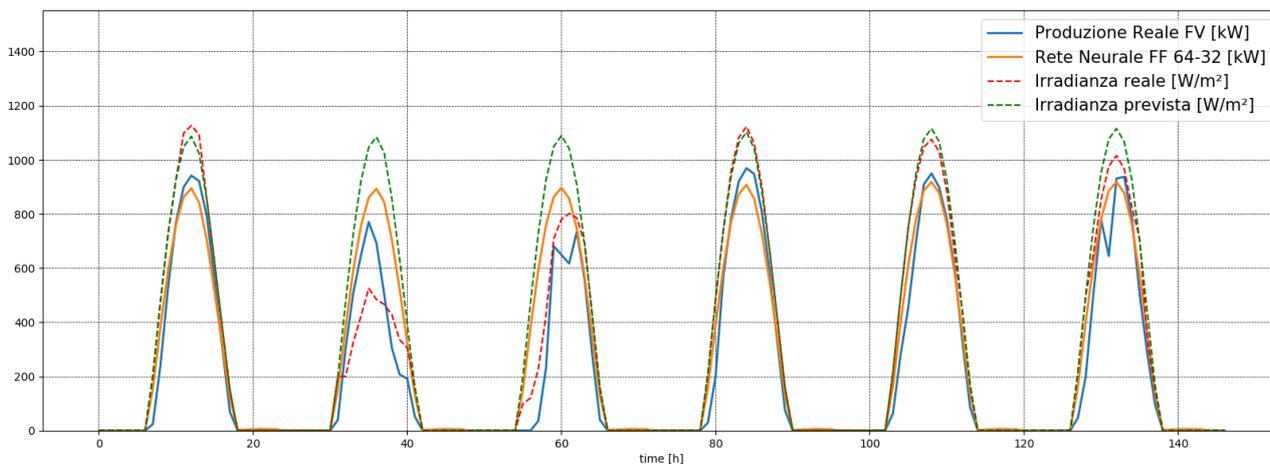


Figura 118: previsione della produzione fotovoltaica basata sulle previsioni metereologiche

In questo secondo caso la previsione metereologica risulta essere meno accurata e si osserva come nella seconda giornata la produzione fotovoltaica predetta dal modello risulti sensibilmente maggiore rispetto a quella reale. La previsione della produzione del cogeneratore risulta di conseguenza errata nel medesimo giorno.

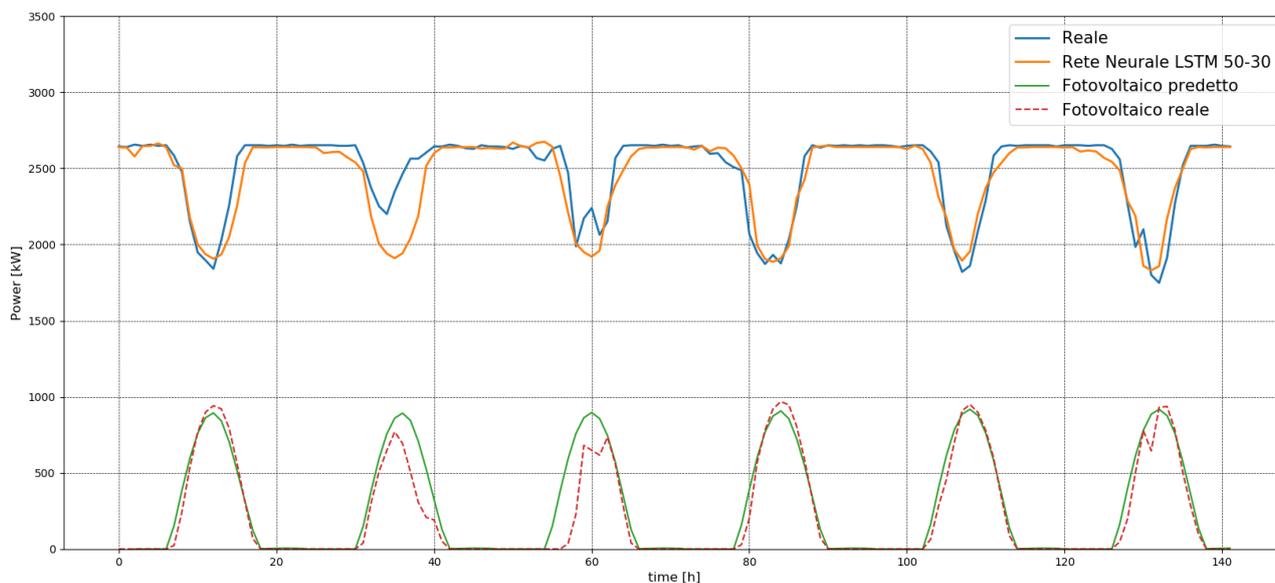


Figura 119: previsione della produzione del cogeneratore basata su previsioni metereologiche

4 Conclusioni

In questa tesi si è esposto l'utilizzo del Machine Learning applicato al monitoraggio energetico in ambito industriale, sia a livello teorico sia con un caso studio concreto riguardante impianti di cogenerazione di due siti industriali.

La grande quantità di sensori e apparecchiature che permettono la misurazione e la lettura di varie grandezze fisiche e la loro integrazione nel settore IoT consentono di raccogliere dati in grande quantità e qualità relativi ai consumi energetici. L'analisi di questi dati consente lo sviluppo nuovi servizi nel campo della gestione e nel risparmio dell'energia. L'utilizzo di algoritmi di Machine Learning ha consentito di sviluppare dei modelli che, basandosi sullo studio dei dati storici, permettono di prevedere efficacemente i consumi futuri degli impianti.

Il primo impianto analizzato è caratterizzato da un profilo di consumi che si ripete in maniera periodica fra una giornata lavorativa e le successive senza grandi scostamenti e differenze. I modelli sviluppati presentano tutti un valore di R^2 molto alto (attorno a 0.90) e un errore medio assoluto nelle previsioni al di sotto del 10%, corrispondente ad alcune decine di kW_{el} a fronte di un valore di picco di circa $1MW_{el}$. L'unico inconveniente riscontrato è stato la difficoltà del modello a distinguere efficacemente i cinque giorni lavorativi di una settimana dai giorni del fine settimana in cui lo stabilimento non è attivo, poiché il modello è stato inizialmente addestrato ad analizzare le singole giornate, tuttavia è stato sufficiente cambiare il metodo di addestramento, facendo analizzare una settimana nella sua interezza per ovviare al problema.

Un ulteriore aspetto emerso da questa analisi è la capacità dei modelli di fornire previsioni accurate anche nel caso in cui la strumentazione preposta alla misurazione delle grandezze abbia problemi di lettura o trasmissione dei dati. Il lavoro di analisi effettuato sul primo stabilimento ha riscontrato la presenza saltuaria di errori di misurazione dei consumi; questo inconveniente rende di fatto impossibile conoscere l'effettivo consumo dello stabilimento nell'arco di alcune ore, tuttavia, come mostrato in Figura 120, il modello di previsione non è influenzato da uno scostamento dei dati così piccolo in termini statistici ed è in grado di fornire una stima affidabile anche in mancanza di dati corretti.

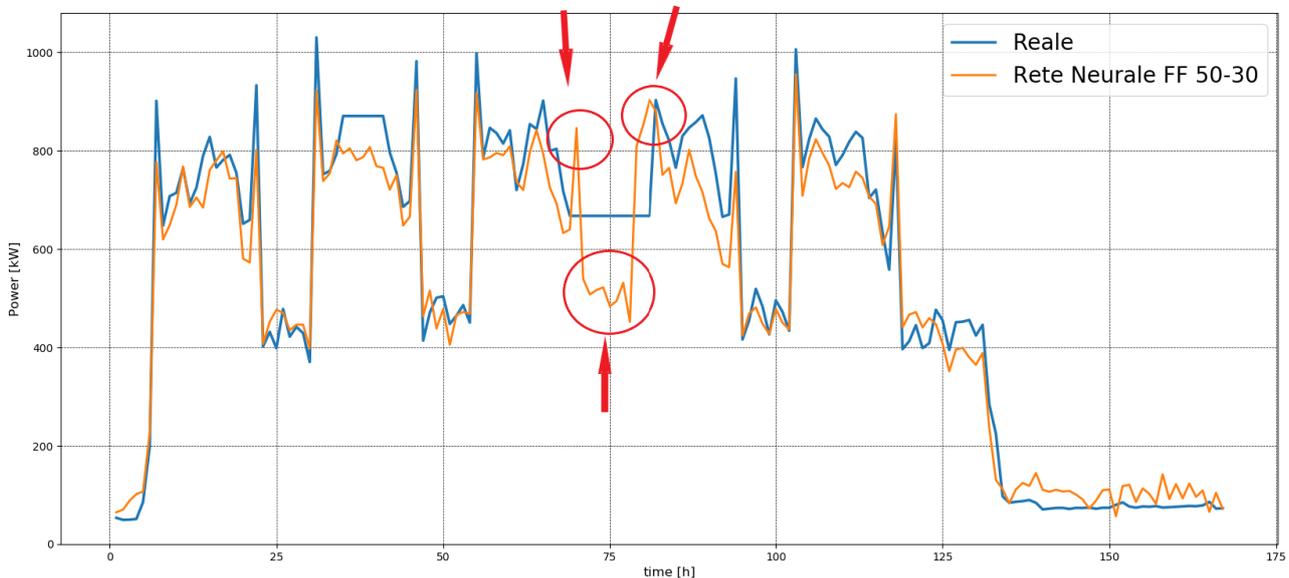


Figura 120: previsione di una settimana con raccolta dati parzialmente mancante

Dal punto di vista della tipologia di rete utilizzata, si sono sviluppate in parallelo due tipologie di rete, quelle Feed Forward (FF) e quelle Long Short Term Memory (LSTM), le quali, per questa tipologia di problema, hanno restituito risultati molto simili, risultando entrambi valide alternative nello sviluppo di sistemi di Machine Learning per la previsione dei carichi energetici. L'aumento della complessità delle reti utilizzate non ha consentito di ottenere un miglioramento significativo nella precisione dei risultati (si rimanda al paragrafo 3.3.10).

Di seguito, nella Tabella 39, è riportato un confronto qualitativo i vari modelli sviluppati evidenziando pro e contro di ognuno.

Tabella 39: riassunto prestazioni reti per il primo stabilimento

Rete	Qualità Previsioni	Flessibilità modello	Semplicità modello	Facilità sviluppo	Utilità previsione	Generale	Note
Previsione Oraria	••••	•••	•••	••••	•	••	Previsioni precise ma limitate ad una sola ora.
Previsione Giornaliera	••	•	••	••	•	•	Buoni risultati nei giorni lavorativi, non affidabile nel fine settimana.
Previsione settimanale	••••	••	•••	••	••	•••	Ottime previsioni nell'arco della settimana, ma più complesso da sviluppare.

L'algoritmo utilizzato si è quindi dimostrato capace di predire efficacemente i consumi dello stabilimento analizzato. La previsione relativa all'ora successiva è molto affidabile, tuttavia l'arco temporale a cui è riferita e la conseguente utilità come strumento di supporto sono limitate. La previsione di un'intera settimana consente di avere una ottima previsione per un arco di tempo considerevole in grado di distinguere efficacemente i giorni lavorativi dai giorni di fermo nel fine settimana.

Il secondo impianto analizzato è caratterizzato dalla presenza simultanea di un motore cogenerativo e un impianto fotovoltaico per la generazione di energia elettrica. La produzione di energia da parte dell'impianto fotovoltaico dipende esclusivamente dalle condizioni metereologiche del sito, le quali non sono controllabili, questo comporta una produzione estremamente discontinua e non regolabile. Per questo motivo l'energia generata dal parco fotovoltaico ha la priorità rispetto a quella prodotta tramite l'impianto di cogenerazione, il quale viene regolato in base alla produzione da fonte rinnovabile. La conseguenza di questo accoppiamento è che il profilo di carico del cogeneratore risulta molto irregolare e dipendente sia dalla produzione dello stabilimento sia da condizioni esterne.

Il primo modello sviluppato per questo stabilimento non considerava le condizioni climatiche e di conseguenza, pur fornendo risultati in assoluto apprezzabili, è risultato significativamente peggiore rispetto al modello impiegato per lo stabilimento precedente, poiché, sebbene avesse individuato la tendenza del cogeneratore a diminuire la produzione durante le ore centrali della giornata, non era in grado di quantificare in maniera accurata le variazioni della produzione ed eventuali cali nella produzione a causa delle condizioni atmosferiche della singola giornata. Come si può osservare nella Figura 121 sottostante l'irraggiamento solare possa variare molto da una giornata alla successiva andando ad influenzare pesantemente la produzione di energia elettrica del parco fotovoltaico e, di conseguenza, la produzione del cogeneratore.

Si è quindi sviluppato un modello in grado di effettuare una stima della produzione fotovoltaica basandosi sulle previsioni metereologiche e, tenendo conto di queste ultime, predire con maggiore efficacia il profilo di carico dello stabilimento.

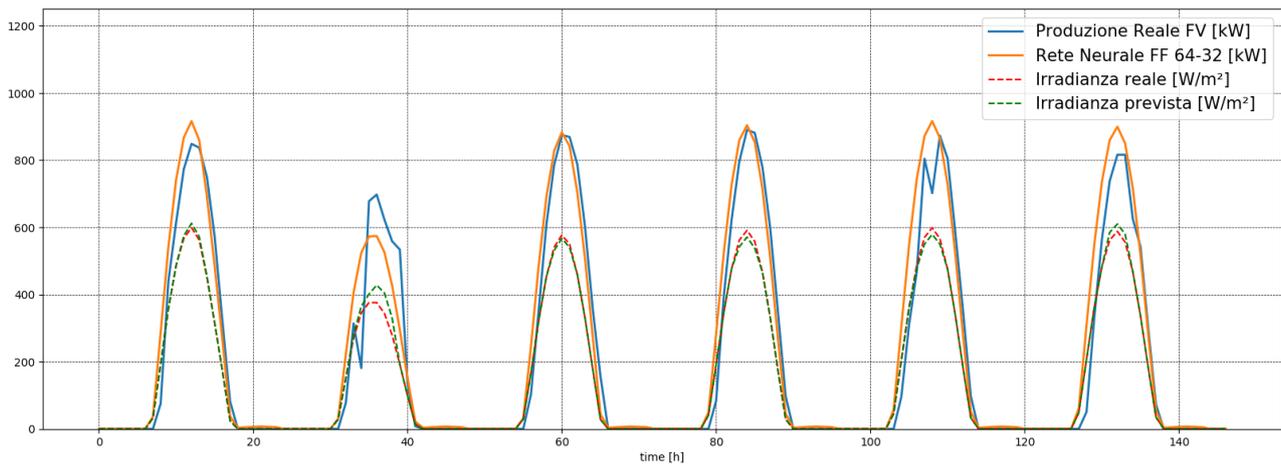


Figura 121: previsione della produzione del fotovoltaico in base all'irraggiamento solare

I risultati delle previsioni relative alla produzione del parco fotovoltaico sono stati salvati ed impiegati come input per la previsione della curva di carico del cogeneratore. Come si osserva nella Figura 122 sottostante, il modello è in grado di ridurre od aumentare efficacemente la produzione del cogeneratore basandosi sulle previsioni della produzione da parte dell'impianto fotovoltaico. Questa modifica ha consentito un incremento del valore di R^2 da circa 0.40 a oltre 0.50 e una diminuzione dell'errore medio assoluto da circa 150 kW_{el} a circa 100 kW_{el} a fronte di circa 2,6 MW_{el}.

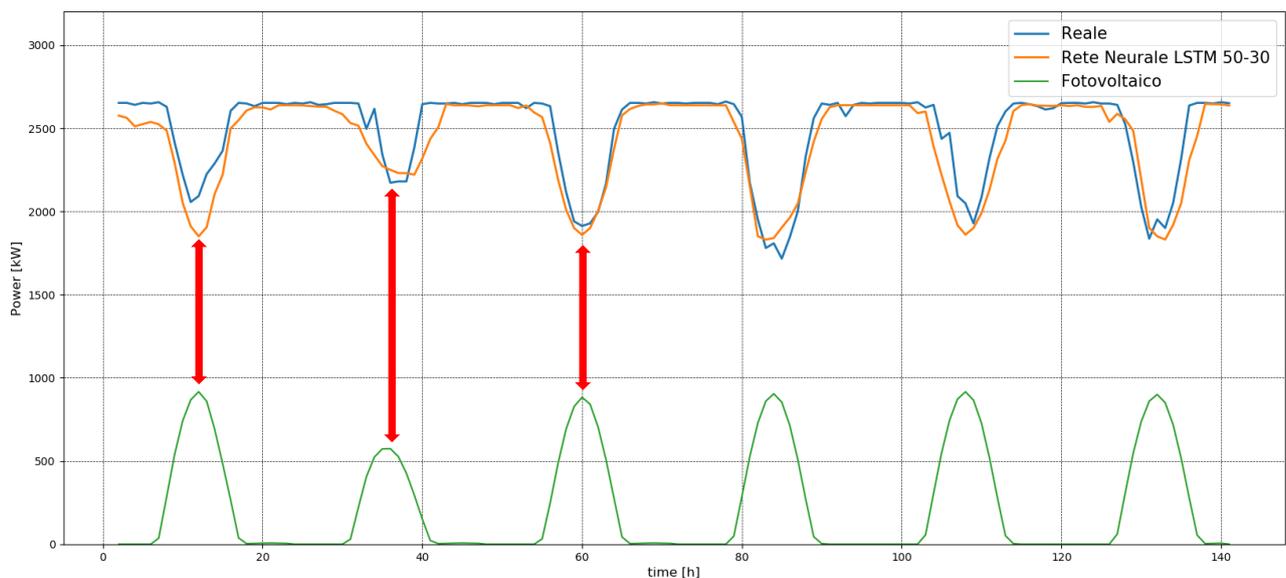


Figura 122: modello previsionale giornaliero con impianto fotovoltaico

L'algoritmo sviluppato per il secondo stabilimento, nonostante un grado di complessità maggiore rispetto al caso precedente, si è dimostrato efficace nella previsione dei carichi riuscendo a mettere in relazione le condizioni climatiche con la produzione richiesta al cogeneratore. Anche in questo caso si sono sviluppate parallelamente reti FF e LSTM ottenendo risultati simili, l'aumento della complessità delle reti utilizzate non ha consentito di ottenere un miglioramento significativo nella precisione dei risultati (si rimanda al paragrafo 3.4.10). Di seguito, nella Tabella 40, è riportato un confronto qualitativo i vari modelli sviluppati evidenziando pro e contro di ognuno.

Tabella 40: riassunto prestazioni reti per il secondo stabilimento

Rete	Qualità Previsioni	Flessibilità modello	Semplicità modello	Facilità sviluppo	Utilità previsione	Generale	Note
Previsione Oraria	●●●●	●●●	●●●	●●●	●	●●	Previsioni precise ma limitate ad una sola ora.
Previsione Giornaliera	●●	●●	●●	●●●	●●	●●	Buoni risultati, ma non tiene conto del FV.
Previsione settimanale	●●	●●	●●	●●	●●	●●	Buoni risultati, ma non tiene conto del FV. Risultati molto simili alla previsione giornaliera
Previsione giornaliera con accoppiamento al fotovoltaico	●●●●	●●●●	●	●	●●●●	●●●	Ottima qualità delle previsioni, ma molto complesso da sviluppare

Poiché il secondo stabilimento è attivo tutti i giorni della settimana non si presenta il problema della differenza sostanziale nei carichi durante il fine settimana, come nel caso precedente, tuttavia è stato necessario implementare la capacità di prevedere la produzione da parte dell'impianto fotovoltaico per ottenere una previsione accurata del profilo di carico del cogeneratore.

In definitiva i modelli sviluppati nello studio forniscono la capacità di prevedere con buona accuratezza i consumi di uno stabilimento permettendo di adottare strategie volte ad ottimizzarne le prestazioni sia da un punto di vista energetico sia da un punto di vista economico:

- È possibile effettuare una stima preventivo del funzionamento del cogeneratore per un determinato intervallo di tempo in modo da gestire e bilanciare in maniera migliore i carichi sia elettrici sia termici, individuando le ore nelle quali risulti economicamente vantaggiosa la produzione tramite cogenerazione e le ore nelle quali convenga acquistare elettricità dalla rete. Risulta particolarmente utile nei casi in cui il fabbisogno è soddisfatto da un mix di fonti (cogenerazione, rete elettrica nazionale e impianti da fonti rinnovabili);
- Conoscere in anticipo la presenza di picchi e buchi nel fabbisogno consente una migliore integrazione con sistemi di accumulo, le cui fasi di carica e scarica possono essere pianificate con cura;
- La presenza di una stima affidabile dell'evoluzione dei carichi consente di individuare con maggiore facilità la presenza di consumi anomali ed eventuali malfunzionamenti nel ciclo produttivo o nella strumentazione di controllo.

Dalle analisi effettuate nella tesi sono emersi alcuni elementi che necessitano di particolare attenzione nell'utilizzo di algoritmi di Machine Learning, i quali debbono essere sempre tenuti in considerazione nello sviluppo di questi modelli:

- L'importanza di selezionare con cura le diverse grandezze in input, utilizzando solo quelle che sono strettamente correlate fra loro da un rapporto di causalità rispetto all'output desiderato;
- La necessità di aver uno storico dei dati quanto più grande possibile, in modo da poter addestrare efficacemente il modello per sviluppare la capacità di generalizzare e quindi di fornire una previsione

affidabile e, contemporaneamente, la possibilità di garantire una continua raccolta dei dati al fine di effettuare le previsioni future;

- Nell'eventualità di modifiche sostanziali al ciclo produttivo o cambiamenti delle apparecchiature di generazione occorre sviluppare un nuovo modello o effettuare nuovamente il processo di addestramento, poiché le previsioni fornite dal modello rispecchiano lo storico dei dati analizzati.

Nonostante i limiti sopra indicati, la presente tesi ha dimostrato la solidità delle previsioni ottenibili grazie all'utilizzo del Machine Learning, che si è rivelato in grado di fornire un efficace strumento di supporto nel campo del monitoraggio energetico. Per studi futuri risulta particolarmente interessante lo sviluppo di modelli in grado di gestire in maniera ancora più efficace impianti combinati formati da una componente tradizionale ed una rinnovabile, la cui diffusione diventerà sempre maggiore nel prossimo futuro.

Appendice

Rete FF per la previsione oraria

```
# carichiamo i dati
df=pd.read_csv('lanificio.csv')
df['Timestamp']=pd.to_datetime(df['Timestamp'])
df.describe() #conviene sempre fare un check-out dei dati
df.dtypes
#convertito il Timestamp
df['tempo']=df['Timestamp'].dt.time
df['time_str'] = df['tempo'].astype(str)
df['time_str'] = df['time_str'].str.split(':')
df['minuti'] = df['time_str'].apply(lambda x: int(x[0]) * 60 + int(x[1]))
del df['time_str']
del df['tempo']
df.set_index('Timestamp')

# stabilisco il numero di step
SEQ_LEN=24
FUTURE_PERIOD_PREDICT=1

#creo una colonna dove metto i consumi sfalsati di un intervallo temporale
df['fabb_futuro']=df['Pel'].shift(-FUTURE_PERIOD_PREDICT)
df=df[:-1] # elimino l'ultima riga
df.dropna(how='any', inplace=True) #elimino tutti i dati NaN

# funzione per normalizzare i dati
mean_pot=df['Pel'].mean()
std_pot=df['Pel'].std()

def normalization(seq): #funzione per normalizzare i dati
    seq=(seq-mean_pot)/std_pot
    return seq

# funzione per denormalizzare i dati
def denormalization(seq):
    real_val=seq*std_pot+mean_pot
    return real_val

# funzione per preprocessare i dati
def preprocess_df(sequence, steps):
    X, Y = list(), list()
    for i in range(len(sequence)):
        # trovo la fine della sequenza
        end_ix = i + steps
        # controllo se se è oltre
        if end_ix > len(sequence)-1:
            break

        seq_x, seq_y = sequence[i:end_ix], sequence[end_ix]
        X.append(seq_x)
        Y.append(seq_y)
    return np.array(X), np.array(Y)

#organizzo i dati in training e validatio
times=sorted(df.index.values)
last_5pct = sorted(df.index.values)[-int(0.066*len(times))]
```

```

test_df=df[df.index >= last_5pct]      # creiamo il validation set
main_df=df[df.index < last_5pct]      # riaggiorniamo anche il training set

pot_train=pd.Series.tolist(main_df['Pel'])
pot_test=pd.Series.tolist(test_df['Pel'])

pot_train=normalization(pot_train)
pot_test=normalization(pot_test)

train_x, train_y = preprocess_df(pot_train, SEQ_LEN)
test_x, test_y = preprocess_df(pot_test, SEQ_LEN)

# Creazione del modello
model =Sequential()
model.add(Dense(25,                                activation='relu',
input_shape=(train_x.shape[1:]),kernel_regularizer=keras.regularizers.l2(0.001)))
model.add(Dropout(0.2))      #applico un dropout per prevenire l'overfitting
model.add(BatchNormalization())
model.add(Dense(12,activation='relu', input_shape=(train_x.shape[1:])))
model.add(Dropout(0.2))
model.add(BatchNormalization())
model.add(Dense(1))

model.compile(loss='mse', optimizer='Adam', metrics=['mae'])

# 3)APPRENDIMENTO DA PARTE DEL MODELLO
early_stop = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=5)

EPOCHS=200
BATCH_SIZE=10
history = model.fit(train_x, train_y,
                    batch_size=BATCH_SIZE,epochs=EPOCHS, validation_split=0.1,
                    callbacks=[early_stop], verbose=1)

# 4)VALUTAZIONEDELL'APPRENDIMENTO
results = model.evaluate(test_x, test_y)

# 5)PREDIZIONE
test_predictions = model.predict(test_x).flatten()

```

Rete FF per la previsione giornaliera

```

SEQ_LEN=24
FUTURE_PERIOD_PREDICT=1

#1) carichiamo i dati

df=pd.read_csv('lanificio.csv')

df['Timestamp']=pd.to_datetime(df['Timestamp'])
df.describe() #conviene sempre fare un check-out dei dati
df.dtypes

df.set_index('Timestamp')
df.dropna(how='any', inplace=True) #elimino tutti i dati NaN

mean_pot=pd.to_numeric(df['Pel']).mean()
std_pot=pd.to_numeric(df['Pel']).std()

def normalization_pot(seq): #funzione per normalizzare i dati

```

```

    seq=(seq-mean_pot)/std_pot
    return seq

def denormalization_pot(seq):
    real_val=seq*std_pot+mean_pot
    return real_val

#funzione per preprocessare i dati
def preprocess_df(df) :
    for col in df.columns:
        if col != 'Timestamp' :
            df[col]=normalization_pot(pd.to_numeric(df[col]))

    X=pd.DataFrame()
    X1=pd.DataFrame()
    Y=pd.DataFrame()
    for count in range (0,24):
        X1[f"ore {count}:00"]=df['Pel'].shift(-count)

    kk=(len(df))/24
    kk=int(round(kk))
    X1=X1[:-24]

    for count in range (1, kk):
        X[count]=X1.iloc[0]
        X1=X1.tail(-24)

    X=X.T
    Y=X.tail(-1)
    X=X[:-1]
    return X, Y

#organizzo i dati in training e validation
times=sorted(df.index.values)
last_5pct = sorted(df.index.values)[-int(0.05*len(times))]

test_df=df[df.index >= last_5pct]      # creiamo il validation set
main_df=df[df.index < last_5pct]      # riaggiorniamo anche il training set

train_x, train_y = preprocess_df(main_df)
test_x, test_y = preprocess_df(test_df)

# 2)COSTRUZIONE DEL MODELLO
#=====

model = keras.Sequential([
    keras.layers.Dense(50, activation=tf.nn.relu,
                       input_shape=(train_x.shape[1],)),
    keras.layers.Dropout(0.2),
    keras.layers.Dense(30, activation=tf.nn.relu),
    keras.layers.Dropout(0.2),
    keras.layers.Dense(24)
])

model.compile(loss='mse',
              optimizer='Adam',
              metrics=['mae'])

model.summary()

```

```

# 3)APPRENDIMENTO DA PARTE DEL MODELLO
early_stop = keras.callbacks.EarlyStopping(monitor='val_loss', patience=5)

EPOCHS=200
BATCH_SIZE=1
history = model.fit(train_x, train_y,
                    batch_size=BATCH_SIZE, epochs=EPOCHS, validation_split=0.1,
                    callbacks=[early_stop], verbose=1)

# 4)VALUTAZIONEDELL'APPRENDIMENTO
results = model.evaluate(test_x, test_y)

# 5)PREDIZIONE
test_predictions = model.predict(test_x).flatten()

```

Calcolo degli indici

Le principali leggi che regolamentano la cogenerazione sono la direttiva comunitaria 2004/8/CE, la delibera 2002/42, la legge 2006/26, il provvedimento del consiglio dei ministri del 8/2/2007 e il decreto ministeriale dello Sviluppo Economico del 4/08/2011.

In questi provvedimenti viene introdotto il concetto di cogenerazione ad alto rendimento ed i criteri necessari al suo riconoscimento.

Direttiva europea 2004/8/CE

La direttiva comunitaria promuove l'utilizzo della cogenerazione, è stata recepita a livello nazionale attraverso il decreto legislativo del 20/02/2007.

Vengono indicati i parametri caratteristici della cogenerazione ad alto rendimento:

- Per le unità di piccola cogenerazione (potenza installata inferiore a 1 MW_e) e microcogenerazione (potenza installata inferiore a 50 kW_e) è sufficiente che garantiscano un risparmio di energia primaria rispetto alla produzione separata;
- In tutti gli altri casi l'impianto di cogenerazione deve garantire un risparmio di energia primaria non inferiore al 10% rispetto alla produzione separata.

L'energia prodotta mediante cogenerazione è valutata pari alla produzione totale annua di elettricità misurata in uscita ai principali tipi di generatori.

Per tutte le altre unità aventi un rendimento inferiore ai valori sopraindicati la cogenerazione va calcolata mediante la formula:

$$E_{CHP} = C \cdot H_{CHP} \quad ()$$

Dove:

- E_{CHP} è l'energia elettrica prodotta dalla cogenerazione;
- C è il rapporto energia/calore;
- H_{CHP} è la quantità di energia termica utile prodotta dalla cogenerazione.

La quantità di energia elettrica prodotta da cogenerazione deve basarsi sul rapporto effettivo energia/calore, nel caso in cui non sia noto si possono utilizzare i valori fissati dalla normativa in base alla tecnologia adoperata.

Il risparmio di energia primaria deve essere calcolato attraverso la specifica formula nella quale vanno inseriti gli indici di prestazione dell'impianto. Tali valori vanno modificati in relazione alle condizioni climatiche

relative alla posizione del sistema. I valori aggiornati sono contenuti nel DM del 4/08/2011, in riferimento al Regolamento delegato (UE) 2015/2402 della Commissione del 12 ottobre 2015.

LA formula per il calcolo del PES (Primary Energy Saving) è al seguente:

$$PES = 1 - \frac{1}{\frac{\eta_{t,CHP}}{\eta_{t,ref}} + \frac{\eta_{el,CHP}}{\eta_{el,ref}}} \quad ()$$

Dove:

- $\eta_{th,CHP}$ è il rendimento termico della produzione mediante cogenerazione, definito come la quantità annua di calore utile divisa per l'energia contenuta nell'intero combustibile di alimentazione, impiegato per produrre sia il calore utile che l'energia elettrica da cogenerazione;
- $\eta_{th,s}$ è il valore di rendimento di riferimento della produzione separata di calore;
- $\eta_{el,CHP}$ è il rendimento elettrico della produzione mediante cogenerazione, definito come energia elettrica annua da cogenerazione divisa l'energia contenuta nell'intero combustibile di alimentazione, impiegato per produrre sia il calore utile che l'energia elettrica da cogenerazione;
- $\eta_{el,s}$ è il valore di rendimento di riferimento per la produzione separata di energia elettrica.

Decreto ministeriale dello Sviluppo economico del 4/08/2011

Il decreto ministeriale del 4 agosto 2011 stabilisce i nuovi criteri per il riconoscimento della condizione di CAR. Il decreto completa il recepimento della direttiva 2004/8/CE iniziato con il decreto legislativo 8 febbraio 2007, n. 20 e in particolare ne integra e sostituisce gli allegati. Il decreto esplicita le metodologie e i criteri da utilizzare per la valutazione del funzionamento di una unità come CAR.

Il DM 4 agosto 2011 recepisce tutte le precisazioni della direttiva 2004/8/CE così come di seguito riportato:

- la produzione di energia elettrica da unità di produzione combinata di energia elettrica e calore con turbina a vapore a contropressione, turbina a gas con recupero termico, motore a combustione interna, microturbine, motori Stirling e celle a combustibile è da ritenere interamente energia elettrica qualificabile come cogenerativa se dette unità presentano un rendimento di primo principio annuo almeno pari al 75%;
- la produzione di energia elettrica da unità di produzione combinata di energia elettrica e calore a ciclo combinato con recupero di calore e con turbina a condensazione con estrazione di vapore è da ritenere interamente energia elettrica qualificabile come cogenerativa se dette unità presentano un rendimento di primo principio annuo almeno pari all'80%.

Se tali unità di produzione combinata di energia elettrica e calore presentano rendimenti di primo principio inferiori alle soglie rispettivamente del 75% e dell'80%, l'energia elettrica prodotta è divisa in due frazioni: una qualificabile come cogenerativa, l'altra qualificabile come non cogenerativa. La quantità di energia elettrica qualificabile come cogenerativa è la quota parte di energia elettrica che sarebbe prodotta se, a parità di calore utile, l'unità funzionasse con un rendimento di primo principio pari al valore di soglia caratteristico per ciascuna tecnologia. La quantità di energia elettrica qualificabile come cogenerativa e così calcolata è da utilizzarsi anche per armonizzare i dati statistici a livello europeo. Discorso analogo è da riproporre anche per quanto concerne l'energia associata al combustibile in ingresso all'unità. Infatti, anche tale energia deve essere suddivisa in due aliquote qualora il rendimento globale dell'impianto non raggiunga le soglie stabilite e sopra riportate.

Il riconoscimento del titolo CAR garantisce una serie di benefici, in particolare:

- la precedenza nel dispacciamento dell'energia elettrica prodotta tramite cogenerazione rispetto a quella prodotta da fonti convenzionali;
- agevolazioni fiscali relative alle accise sul gas metano adoperato come combustibile per la cogenerazione;
- accesso al servizio di scambio sul posto dell'energia elettrica prodotta da impianti CAR sino a 200kW;

- applicazione di condizioni tecniche ed economiche semplificate per quanto concerne l'allacciamento alla rete elettrica;
- agevolazioni tariffarie per la realizzazione di ulteriori impianti a fonti rinnovabili;
- incentivazione dell'energia elettrica prodotta tramite CAR netta immessa in rete da impianti alimentati a biometano. [14]

Gli impianti CAR possono accedere al meccanismo dei certificati bianchi. Il rilascio dei certificati avviene in base all'energia risparmiata, calcolata come segue:

$$RISP = \frac{E_{chp}}{\eta_{ERIF}} + \frac{H_{chp}}{\eta_{TRIF}} - F_{chp} \quad ()$$

Dove:

- RISP è il risparmio di energia primaria realizzato in un anno solare dalla unità CAR;
- E_{chp} è l'energia elettrica prodotta dalla parte CHP nell'anno solare;
- H_{chp} è l'energia termica prodotta dalla parte CHP nell'anno solare;
- F_{chp} è l'energia di alimentazione consumata dalla parte CHP nell'anno solare;
- η_{ERIF} è il rendimento medio convenzionale del parco di produzione elettrica italiano, assunto pari a 0,46 e corretto in funzione della tensione di allacciamento;
- η_{TRIF} è il rendimento medio convenzionale del parco di produzione termico italiano, assunto pari a 0,82 se si utilizzano direttamente i gas di scarico e 0,90 nel caso di produzione di acqua calda e vapore.

Il numero di Certificati Bianchi a cui l'impianto ha diritto viene calcolato con la seguente formula:

$$CB = RISP \cdot 0,086 \cdot K \quad ()$$

Dove K è un coefficiente di armonizzazione che dipende dalla potenza dell'unità.

Bibliografia

- [1] D. Chiaroni, F. Frattini e M. Chiesa, «Energy Efficiency Report,» Milano, 2014.
- [2] D. Evans, «The Internet of Things-How the next evolution of the Internet is changing everything,» 2016.
- [3] Business Insider, «www.businessinsider.com,» 2018. [Online]. Available: <https://www.businessinsider.com/iot-forecast-book-2018-7?IR=T>.
- [4] McKinsey Global Institute, «The Internet of Things: mapping value beyond the hype,» 2015.
- [5] B. Marco, L. Degiovanni, M. C. Masoero e A. Portoraro, «Monitoraggio energetico 4.0 Il nuovo ruolo della digitalizzazione,» *SISTEMI&IMPRESA*, 2018.
- [6] K. Minhaj Ahmad e S. Khaled, «IoT security: Review, blockchain solutions, and open challenges,» 2018.
- [7] L. Settineri, «Industry 4.0: la chiave per la competitività,» 2017.
- [8] S. Rota Bulò, «Appunti di reti neurali,» UniVe, 2015. [Online].
- [9] Edison EDF Group, «Building Energy Management Systems,» Melfi, 2018.
- [10] S. Seijo, d. C. Inés e J. Echanobe, «Electric Efficiency Modelling of a Complex Cogeneration,» *International Journal of Machine Learning and Computing*, p. 5, 5 October 2015.
- [11] E. Isaksson e M. Karpe Conde, *International Journal of Machine Learning and Computing*, Stoccolma: KTH Royal Institute of Technology, 2018.
- [12] N. Shukla, *Machine learnin with Tensorflow*, New York: Manning Publication, 2017.
- [13] A. Gulli, *Deep Learning with Keras*, Birmingham: Packt Publishing, 2017.
- [14] GSE, «Guida alla cogenerazione ad alto rendimento,» 2018.