



Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Assistenza Medica In Contextual awareness

Deliverable di Progetto (AMICO)

Antonio Coronato, Rosamaria Donnici, Giovanni Paragliola

Istituto di Calcolo e Reti ad Alte Prestazioni del Consiglio
Nazionale delle Ricerche (ICAR-CNR) Via Pietro
Castellino, 111 – 80131 Napoli

antonio.coronato@icar.cnr.it, rosamaria.donnici@icar.cnr.it,
giovanni.paragliola@icar.cnr.it

RT-ICAR-NA-2020 -04

Novembre 2020

I rapporti tecnici dell'ICAR-CNR sono pubblicati dall'Istituto di Calcolo e Reti ad Alte Prestazioni del Consiglio Nazionale delle Ricerche. Tali rapporti, approntati sotto l'esclusiva responsabilità degli autori, descrivono l'attività del personale e dei collaboratori dell'ICAR, in alcuni casi in un formato preliminare prima della pubblicazione definitiva in altra sede.

Assistenza Medica In Contextual awareness

Deliverable di Progetto (AMICO)

Antonio Coronato, Rosamaria Donnici, Giovanni Paragliola

Istituto di Calcolo e Reti ad Alte Prestazioni del Consiglio Nazionale delle Ricerche (ICAR-CNR) Via
Pietro Castellino, 111 – 80131 Napoli

antonio.coronato@icar.cnr.it, rosamaria.donnici@icar.cnr.it, giovanni.paragliola@icar.cnr.it

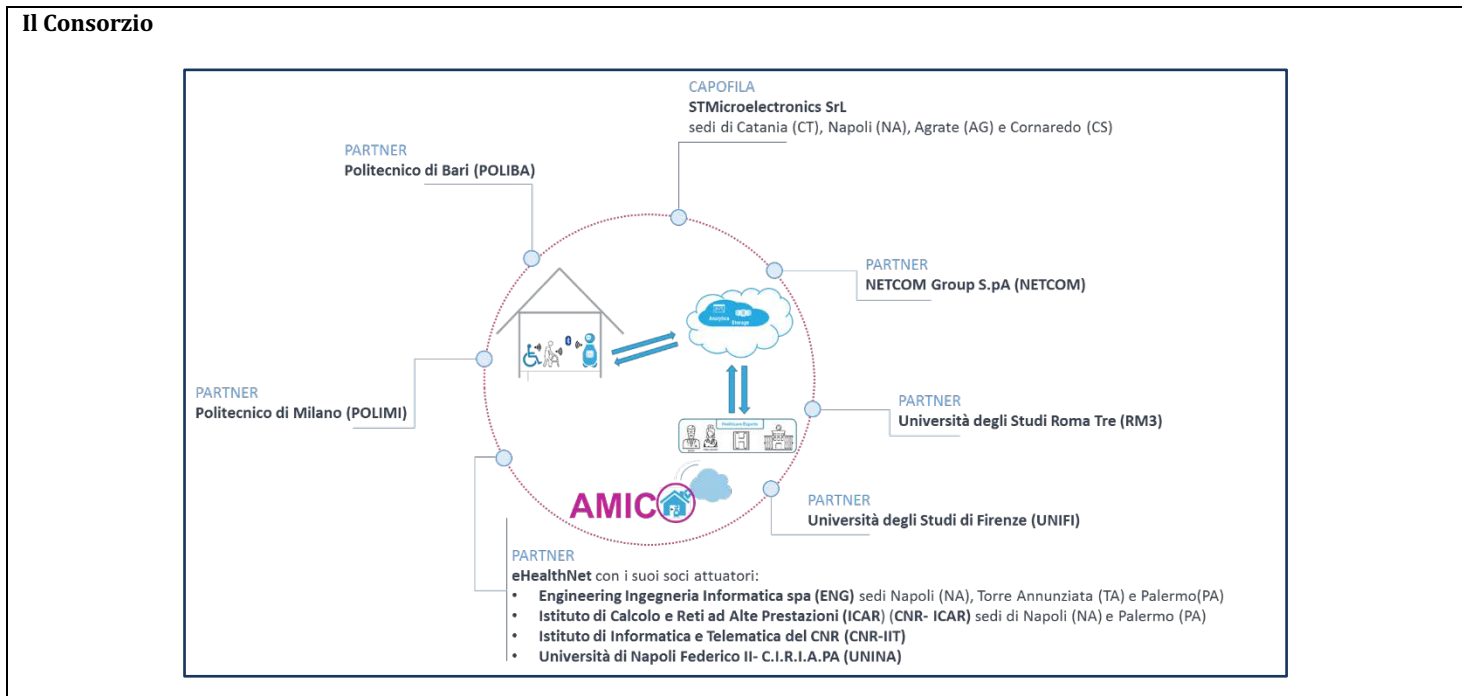
Abstract

Il presente deliverable presenta i) una panoramica sulle tecniche, metodologie e tecnologie per la progettazione, realizzazione e valutazione di modelli atti all'analisi e all'estrazione di conoscenza da dati eterogenei; ii) il modello predittivo sviluppato per l'analisi e la classificazione di segnali biomedicali tempovarianti che consente la predizione del livello di rischio per i pazienti affetti da ipertensione; ed infine, iii) i servizi intelligenti sviluppati per il supporto alla terapia domestica per i pazienti ipertesi. Il modello predittivo è definito a mezzo di una rete ibrida che comprende reti ricorrenti, reti convoluzionate e reti multilivello. La valutazione del modello è stata eseguita a partire da dataset clinici disponibili online. I risultati sperimentali dimostrano la netta superiorità del modello sviluppato per il progetto AMICO rispetto allo stato dell'arte. I servizi di supporto al paziente per l'aderenza alla terapia sono basati su un approccio ibrido che integra tecnologie di Reinforcement Learning e Deep Learning al fine, rispettivamente, di personalizzare il servizio in base alle capacità cognitive, uditive e visive del paziente e di identificare, attraverso un classificatore neurale, potenziali medication errors.

Keywords: Modelli Predittivi, Deep Learning, Elaborazioni di Segnali, Estrazione di conoscenza, Reti Ricorrenti, Reti Convoluzionate

Deliverable di Progetto

Codice identificativo	Titolo del progetto
ARS01_00900	Assistenza Medica In COntextual awareness (AMICO)



Titolo
D4.3.2 - Report di progettazione dei servizi di analisi dei dati ed estrazione di conoscenza

OR	Data rilascio
OR4	M32

Autori (Organizzazione)
CNR-ICAR

Versione	Data	Autore (Organizzazione)	Descrizione
V0.1	15/10/2020	CNR-ICAR	Draft
V0.2	9/11/2020	CNR-ICAR	Draft
V1.0	23/11/2020	CNR-ICAR	Final

Sommario

1	INTRODUZIONE	4
2	ALGORITMI DI INTELLIGENZA ARTIFICIALE PER LA DEFINIZIONE DI MODELLI PREDITTIVI..	13
2.1	TECNICHE DI MACHINE LEARNING.....	21
2.1.1	<i>Alberi Decisionali</i>	23
2.1.2	<i>Reti Neurali</i>	27
2.1.3	<i>Support Vector Machine</i>	33
2.1.4	<i>Classificatori a Regole</i>	36
2.2	TECNICHE DI DEEP LEARNING.....	43
2.2.1	<i>Convolutional Neural Network</i>	44
2.2.2	<i>Recurrent Neural Network</i>	47
2.2.3	<i>Reti Profonde</i>	49
2.2.4	<i>Encoder</i>	51
3	APPROCCI PER LA VALUTAZIONE QUANTITATIVA DEI MODELLI.....	53
3.1	INTRODUZIONE DELLE METRICHE.....	58
4	SOLUZIONI SVILUPPATE NEL PROGETTO AMICO	64
4.1	UNA HYBRID DEEP NETWORK PER L'IDENTIFICAZIONE PRECOCE DI SOGGETTI AD ALTO-RISCHIO DI IPERTENSIONE ...	65
4.1.1	<i>Stato dell'arte</i>	68
4.1.2	<i>Contributo del nostro lavoro</i>	73
4.1.3	<i>Metodo</i>	74
4.1.4	<i>Risultati</i>	79
4.1.5	<i>Discussione dei Risultati Ottenuti</i>	85
4.1.6	<i>Conclusioni</i>	90
4.2	SISTEMA INTELLIGENTE PER IL SUPPORTO DI PAZIENTI CON DISABILITÀ NEL TRATTAMENTO DOMICILIARE BASATO SU APPROCCI DI REINFORCEMENT LEARNING E DEEP LEARNING.....	91
4.2.1	<i>Motivazioni</i>	93
4.2.2	<i>Stato dell'arte</i>	94
4.2.3	<i>Contributo soluzione proposta</i>	99
4.2.4	<i>Background tecnico</i>	100
4.2.5	<i>Architetture del sistema e degli Agenti Software</i>	102
4.2.6	<i>Configurazione e risultati sperimentali</i>	111
4.2.7	<i>Conclusione e sviluppi futuri</i>	120
	BIBLIOGRAFIA	121

1 Introduzione

Negli anni '80 del 1900 molte attività ed imprese si ritrovarono con database di clienti talmente cresciuti che la quantità di dati in loro possesso risultava troppo grande perché impiegati umani potessero analizzarla da soli. Il termine "*dati mining*" fu, dunque, coniato per descrivere una serie di tecniche automatizzate che potevano essere applicate per interrogare tali database e fare inferenze sulle informazioni in essi contenute, assumendo la definizione sintetica di "*analisi di set di dati complessi e di grandi dimensioni*" come valido punto di partenza per una comprensione più profonda.

Molti degli strumenti utilizzati per eseguire il data mining sono metodi statistici standard che esistono da decenni, come la regressione lineare e il clustering; tuttavia, esso include anche una vasta gamma di altre tecniche per l'analisi dei dati, sviluppatesi dalla ricerca sull'intelligenza artificiale, sull'informatica evolutiva e sulla teoria dei giochi. Risulta, quindi, un argomento molto ampio, utilizzato in ogni genere di indagine, ma il cui uso più prolifico è quello di identificare le relazioni nei dati che danno un insight¹ delle preferenze individuali e, soprattutto, che indicano quale evento è più probabile in un determinato scenario.

Il termine "analisi predittiva" è stato introdotto dopo il 2010 come sinonimo di data mining nello sviluppo di strumenti specifici per prevedere il comportamento delle persone come individui o entità come società a responsabilità limitata. Analisi predittiva è, quindi, solo un termine usato per descrivere l'applicazione del data mining a specifici tipi di problema e che sottolinea l'interesse alla previsione di informazioni future sulla base della conoscenza delle informazioni passate a disposizione.

Una delle prime applicazioni di analisi predittiva è stata la valutazione del credito utilizzata per la prima volta dal settore degli ordini postali negli anni '50 del 1900 per identificare i soggetti di cui accettare la richiesta di credito. A metà degli anni '80 il punteggio del credito era diventato il principale strumento decisionale nel settore dei servizi finanziari. I soggetti richiedenti un prestito di denaro, una carta di credito, l'attivazione di un mutuo e così via, vengono valutati da un creditore che deve decidere se ritiene che quel soggetto rimborserà o meno ciò che prende in prestito. Ovviamente il prestito è assegnato solo se il creditore ritiene che il richiedente restituirà il credito. Un tempo questa decisione veniva presa da sottoscrittori umani, che esaminavano ogni richiesta di prestito e prendevano una decisione sulla base della loro opinione di esperti. Oggi, quasi tutte

¹ In psicologia rappresenta la capacità di vedere dentro una situazione, o dentro se stessi; quindi, in genere rappresenta una percezione chiara, un'intuizione netta e immediata di fatti esterni o interni. Tale significato è stato sublimato dal marketing come la capacità di focalizzare un bisogno non soddisfatto del consumatore, o una opportunità più favorevole di consumo che i prodotti esistenti ancora non colgono.

queste decisioni vengono prese automaticamente utilizzando i *modelli predittivi* che si trovano all'interno del sistema di elaborazione delle applicazioni di un'organizzazione.

Per costruire un modello di valutazione del credito, l'analisi predittiva viene utilizzata per vagliare i dati di migliaia di accordi di prestito storici ed identificare quali caratteristiche dei mutuatari siano indicative del loro essere clienti "buoni" che hanno rimborsato i loro prestiti o clienti "cattivi" che hanno fallito. Tali relazioni, una volta identificate sono incapsulate in un pattern che è possibile utilizzare per far previsioni sul comportamento di rimborso futuro di nuovi richiedenti credito. [\(1\)](#)

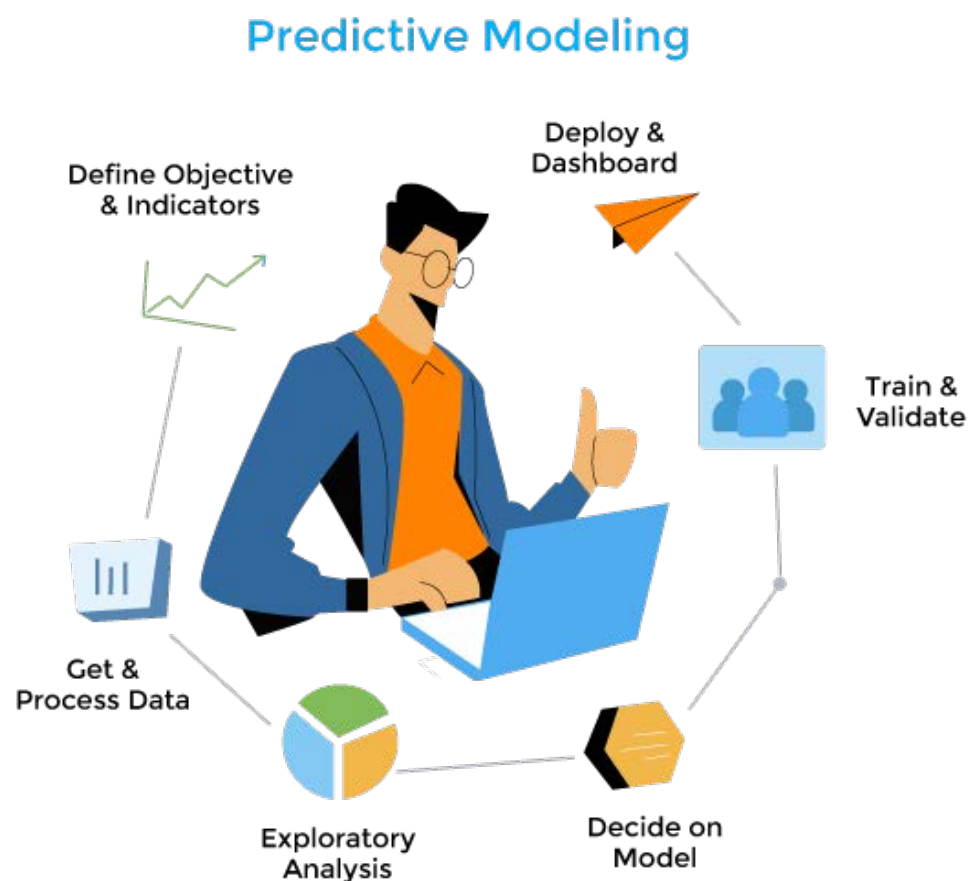


Figura 1: La definizione di un modello predittivo adeguato è un processo articolato di cui l'analisi predittiva e il data mining in genere sono solo un aspetto. Esso è strutturato non solo per predire nuovi dati, ma anche informazioni e comportamenti impliciti nei dati stessi [\(2\)](#)

L'analisi predittiva parte dalla previsione che lo studio si figura di raggiungere: solo dopo aver determinato chiaramente il *goal* a cui mirare è possibile individuare le informazioni necessarie a perseguirlo. Una volta che tali informazioni saranno convertite in dati² puliti ed ottimizzati sarà possibile definire un modello e valutarne la validità. Se le metriche del modello stesso risultano adeguate sarà possibile distribuirlo, monitorarlo e iterarlo coinvolgendo l'utente finale e creando una dashboard per tenere traccia delle prestazioni. [\(2\)](#)

Il *pre-processing* dei dati è, pertanto, il primo passo per la formalizzazione di un'analisi coerente, in quanto converte le informazioni del mondo reale - spesso incomplete, incongruenti e / o prive di determinati comportamenti o tendenze e, probabilmente, contenenti molti errori - in dati utili e consistenti. La qualità dei risultati è in relazione diretta con la qualità dei dati e lo sforzo atto al loro consolidamento influisce sui risultati del processo di predizione. In generale il pre-processing si articola nei vari steps in seguito elencati e descritti:

- **Data cleaning:** come anticipato i dati reperibili sono spesso "sporchi" ovvero affetti da mancanze, da incongruenze e/o rumori o da inconsistenze e/o ridondanze dovute ad errori umani, a malintesi interpretativi o semplicemente a eventi non considerati. Tale "sporcizia" minerebbe di sicuro la correttezza del modello predittivo e per tale ragione i dati vengono ripuliti attraverso processi specifici e spesso complessi, come la compilazione di casi o valori mancanti, il binning per il livellamento dei rumori o la risoluzione delle incongruenze introdotte con il data integration. La distinzione tra casi e valori mancanti, ad esempio comporta un diverso approccio risolutivo: per casi mancanti si intende che alcune unità statistiche, fra quelle originariamente programmate, non sono state rilevate; mentre per valori mancanti, si intende che per

² Tra i dati e le informazioni possiamo ipotizzare che esista lo stesso rapporto che intercorre tra materie prime e prodotti finiti: i dati sono rappresentazioni originarie, cioè non interpretate, di un fenomeno e la loro trasformazione dà vita ad interpretazioni che assumono un significato relativo al soggetto che ne coglie, appunto, l'informazione.

qualche unità statistica osservata risultano mancanti le modalità di uno o più caratteri. Nel primo caso le unità statistiche, effettivamente rilevate sono in numero di m , invece delle n originariamente previste, con $m < n$. E' possibile, quindi o ignorare i dati mancanti e considerare la matrice formata dalle sole m unità note disponibili o, viceversa sostituire le unità $n-m$ mancanti con altre opportunamente selezionate ed equivalenti. Nel caso di valori mancanti, invece, la matrice dei dati, sempre formata da n righe, presenta un vuoto, detto blank, in ciascuna delle posizioni corrispondenti ai dati mancanti, per cui risulta necessario affrontare anche un problema interpretativo oltre che strutturale. Per descrivere meglio, tramite un esempio, tale problema, si consideri un sondaggio in cui si valuta il gradimento di un determinato prodotto: il valore mancante potrebbe essere interpretato come risposta "Non so!", da parte dell'utente interrogato, alternativa a "Si, gradisco il prodotto!" e "No, non lo gradisco!". Anche strutturalmente i valori mancanti sono gestiti a seconda della dipendenza da altri valori - approccio Missing at Random (MAR) - o che la probabilità di riscontrare un valore mancante sia indipendente dagli altri valori e che i dati omessi siano completamente casuali - approccio Missing Completely at Random (MCAR).

- Data Integration: per reperire dati utili è, spesso necessario attingere da più fonti che possono trattare le informazioni in modi diverso. Diventa, quindi necessario confrontare i valori raccolti e combinarli in modo da individuare gli stessi dati anche se denominati diversamente, eliminando, così le ridondanze. Eventuali incongruenze, vengono successivamente individuate analizzando i dati ed eliminando quei valori non attinenti al contesto o incoerenti. Un esempio può venire dalla valutazione del rapporto peso / altezza di una popolazione futura d'individui; il peso potrebbe essere descritto in kg in un database e in libbre in un altro, ma in ambo i casi sono attinenti

alla stessa informazione d'interesse. Viceversa il dato "colore dei capelli" non apporta alcuna informazione utile, ma è altrettanto inappropriato e, quindi, da eliminare anche il dato peso=500kg.

- **Data Transformation:** rende possibile l'uniformità dei dati riducendoli a valori coerenti tra loro mediante modificazioni matematiche e creazione di nuove variabili dalla combinazione di variabili esistenti mediante aggregazione e generalizzazione. In particolare le trasformazioni sulla distribuzione dei dati sono utili per standardizzare distribuzioni anormale e per la linearizzazioni delle variabile, mediante trasformazioni logaritmiche per le distribuzioni asimmetriche e trasformazioni polinomiali utili quando si vogliono linearizzare i dati che sono distribuiti in modo continuo. [\(3\)](#)
- **Data Reduction:** questo passaggio mira a proporre una rappresentazione ridotta dei dati in un data warehouse³, senza subire perdite di informazioni; perciò utilizza strategie opportune quali la discretizzazione del numero di valori di una variabile continua, mediante divisione in classi oppure la gerarchizzazione dei concetti sostituendo, appunto, concetti di livello "basso", come valori numerici ad esempio di una variabile 'età', con rappresentazioni di livello "alto" come, 'giovane, adulto, anziano' [\(4\)](#)

In generale è opportuno impostare l'attività di analisi su base campionaria, in quanto, soltanto in talune circostanze è consigliabile lavorare sull'intera popolazione di riferimento. Le motivazioni che portano all'estrazione di un campione rappresentativo si basano su varie considerazioni che vanno dalla similitudine dei dati all'ottimizzazione di tempi di elaborazione. In particolare lavorare

³ Il concetto data warehouse è più complesso del semplice significato di "magazzino" perchè incapsula anche il processo volto ad immagazzinare e organizzare le informazioni, ottimizzandone la struttura per un facile accesso e per supportare al meglio le decisioni aziendali. In questo processo i dati vengono estratti dalle varie fonti interne o esterne all'azienda, elaborati e ristrutturati all'interno di un'architettura logica da cui poter estrarre le informazioni utili per il business aziendale.

su un sotto-insieme rappresentativo dell'universo di riferimento permette di tenere sotto controllo il rischio di costruire modelli auto-esplicativi: la bontà di una regola consiste nella sua capacità discriminante, ma anche nella sua robustezza.

Il vantaggio di lavorare su base campionaria, costruendo un dataset per il training del modello, uno per la validazione, ed uno per il test, consiste proprio nell'evitare che la regola, adattandosi alle irregolarità ed alla variabilità propria dei dati sui quali è stimata, perda capacità di generalizzazione. [_\(3\)](#)

Lo split dei dati in un set di *training* ed in un set di *test* è pertanto una parte essenziale per l'elaborazione dei modelli predittivi. In genere la maggior parte dei dati viene utilizzata per l'allenamento e quindi per l'apprendimento da parte dell'algoritmo, mentre solo una parte più piccola viene utilizzata per i test. In pratica, dopo che un modello è stato istruito utilizzando il set di training, si testa la sua funzionalità facendo previsioni su di un set di dati che contengono valori già noti, per l'attributo che si desidera prevedere e, per i quali risulta, di conseguenza, facile determinare se le ipotesi siano corrette o meno. Utilizzando dati simili per addestramento e test, in generale è possibile ridurre al minimo gli effetti delle discrepanze e comprendere meglio le caratteristiche del modello. [_\(5\)](#)

L'idea di utilizzare dei dati di addestramento nei modelli di apprendimento automatico è un concetto semplice, ma molto influente per il funzionamento di queste tecnologie. Il training set rappresenta, infatti, l'incipit che descrive al programma come applicare la propria stessa tecnologia per apprendere e produrre risultati sofisticati; è, quindi, il materiale attraverso il quale il computer impara ad elaborare le informazioni. [_\(6\)](#)

Idealmente tra la fase di costruzione del modello attraverso l'apprendimento e la fase di test viene inserita una seconda fase di convalida che aiuta a gestire problemi come il sovradimensionamento, in cui l'algoritmo potrebbe non essere ben calibrato per gestire i dati futuri: se la macchina si

comporta in modo altrettanto accurato sui dati di test come sui dati di training o se esiste un ampio divario tra le prestazioni sui due stadi, si ha l'assenza o la presenza di un overfitting. La *data validation* è un processo che garantisce la consegna di dati puliti e chiari a programmi, applicazioni e servizi che li utilizzano; ne verifica l'integrità e la validità perché siano immessi in diversi software e relativi componenti e garantisce che siano conformi ai requisiti e ai parametri di riferimento. Ciò si ottiene attraverso controlli e regole di convalida - definite in appositi *dizionari di dati* o implementate attraverso software di convalida - specifiche dello studio iterato. [_\(6\)](#)

La *evaluation* dei modelli predittivi è uno dei punti chiave del data mining. I criteri più frequentemente considerati valutano l'accuratezza complessiva ottenuta con appropriati *model validation methods*, per visualizzare le prestazioni metriche. Uno dei più famosi di tali metodi è ottenuto costruendo una matrice di confusione con uno specifico layout, che consente la visualizzazione delle prestazioni di un algoritmo e mostra le classificazioni previste ed effettive. Una matrice di confusione ha dimensioni $L \times L$, dove L è il numero di classi ed è uno strumento prezioso non solo perché mostra la frequenza con cui il modello ha previsto correttamente un valore, ma perché evidenzia anche quali altri valori sono stati previsti in modo errato più frequentemente. [_\(1\)](#) In questa fase è possibile valutare il risultato nel contesto dell'obiettivo prefissatosi e possono emergere nuovi requisiti dovuti a nuovi modelli scoperti durante la *evaluation* stessa. Ottenere approfondimenti e miglioramenti è, quindi, un processo iterativo nel data mining e la decisione *go o no-go* deve essere presa in questo passaggio, prima che il progetto passi alla fase di implementazione.

Le informazioni, in questo modo ottenute devono essere rappresentate in modo tale che le parti interessate possano utilizzarle facilmente, ogni volta che lo desiderano. In base ai requisiti aziendali, la fase di *deployment* potrebbe essere semplice, come la creazione di un report o complessa al punto da richiedere la ripetizione dell'intero processo di previsione. Sicuramente il

rapporto finale deve sintetizzare le insights e i trends del progetto, per rileggerlo in termini di sviluppo e miglioramento futuri. [\(7\)](#)

Questo approccio tecnico alla costruzione di un modello analitico per la descrizione e la previsione di un evento è in realtà il risultato di più logiche, che si sono sviluppate nel corso degli anni per affrontare il medesimo problema di analizzare ingenti quantità di dati. In generale si descrivono tre differenti tipologie di analisi dei dati che ne rappresentano tre ottiche differenti:

- **Analisi descrittiva:** sintetizza e descrive dei dati grezzi con lo scopo di trovare delle correlazioni nascoste, al fine di riassumere e chiarire le dinamiche e le performance di metriche prestabilite e ricavarne indicazioni sul modo migliore per approcciare le attività future. Questo tipo di analisi è particolarmente introdotta negli studi di settore aziendali in quanto permette di comprendere cosa stia succedendo in un determinato dominio di interesse ed effettuare scelte mirate di conseguenza;
- **Analisi predittiva:** prevede cosa accadrà in futuro in un determinato campo applicativo. Per condurre queste analisi è necessario aver effettuato in precedenza un'analisi descrittiva sui dati di interesse, in modo da conoscerne l'evoluzione storica e le eventuali correlazioni tra i set da assegnare all'algorithm. Questo tipo di analisi viene utilizzata per migliorare, in termini di competitività, le strategie di marketing e far emergere nuove opportunità di business;
- **Analisi prescrittiva:** sintetizza i dati con l'utilizzo congiunto di scienze matematiche, regole di business e tecnologie di Machine Learning al fine di specificare, non solo l'evento di previsione, ma anche il percorso da intraprendere per raggiungere l'obiettivo stesso desiderato. Essendo complesse da amministrare, la maggior parte delle aziende non le utilizza ancora.

Da gennaio 2020, a Napoli, in particolare nella zona attinente all'ASL3 sono in sperimentazione 40 robot per l'assistenza ad anziani - anche affetti da Alzheimer - affidati a famiglie volontarie che, vivendo con loro, ne testano capacità, limiti e benefici per la loro salute. Lo scopo della sperimentazione è dotare i robot di un'intelligenza socievole e capace di adattare il proprio comportamento alla persona specifica che si trova ad assistere. Il robot, dunque, grazie al meccanismo di machine learning è in grado di individuare e apprendere, in input, segnali inusuali come quelli di sofferenza nella persona, grazie ai quali può attivare in automatico un'azione di risposta, che può essere quella di consigliare all'anziano di chiamare il medico o un parente. [\(8\)](#)

2 Algoritmi di Intelligenza Artificiale per la definizione di modelli predittivi

Un sistema intelligente viene realizzato cercando di ricreare una o più delle differenti forme di intelligenza riconosciute dalla teoria di Gardner - che vanno dall'intelligenza spaziale a quella sociale, da quella cinestetica a quella introspettiva - che, anche se spesso definite come semplicemente umane, in realtà possono essere ricondotte a particolari comportamenti riproducibili da alcune macchine.

Alla base delle problematiche legate allo sviluppo di sistemi e programmi di Intelligenza Artificiale vi sono tre parametri che rappresentano i cardini del comportamento umano, ossia

- una conoscenza non sterile,
- una coscienza che permetta di prendere decisioni non solo secondo la logica
- l'abilità di risolvere problemi in maniera differente anche a seconda dei contesti nei quali ci si trova.

Per potersi avvicinare a queste capacità, la ricerca si è concentrata sullo sviluppo di algoritmi sempre più numerosi che possano emulare diversi comportamenti a seconda degli stimoli ambientali. Tali algoritmi complessi, inseriti all'interno di sistemi intelligenti, sono quindi in grado

di 'prendere decisioni', ossia di effettuare scelte a seconda dei contesti in cui sono inseriti. Nel caso degli algoritmi connessi ai sistemi intelligenti dei veicoli, ad esempio, un'automobile senza conducente può decidere, in caso di pericolo, se sterzare o frenare a seconda della situazione, ossia a seconda che le informazioni inviate dai vari sensori permettano di calcolare una maggiore percentuale di sicurezza, per il conducente e i passeggeri, con una frenata piuttosto che con una sterzata.

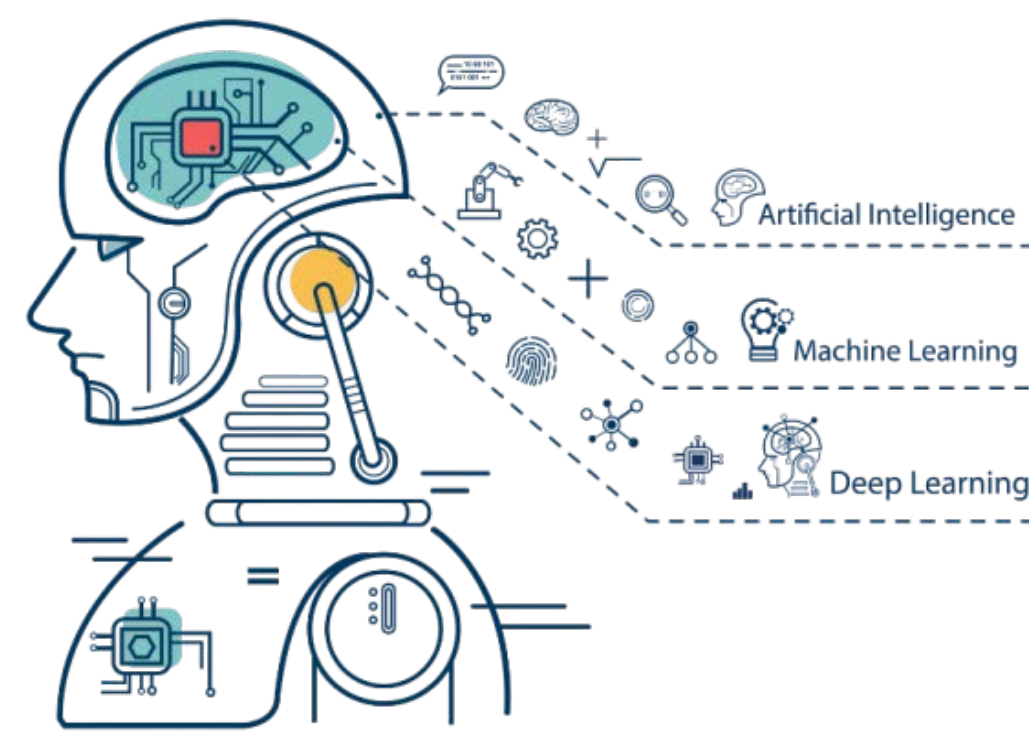


Figura 2 : «L'intelligenza artificiale è una disciplina appartenente all'informatica che studia i fondamenti teorici, le metodologie e le tecniche che consentono la progettazione di sistemi hardware e sistemi di programmi software capaci di fornire all'elaboratore elettronico prestazioni che, a un osservatore comune, sembrerebbero essere di pertinenza esclusiva dell'intelligenza umana.» Somalvico Ing. Marco

Per realizzare algoritmi in grado di prendere decisioni sempre più precisi e complessi è sorto un vero e proprio settore specifico, definito *rappresentazione della conoscenza*, che studia le possibilità di rendere la capacità di ragionamento dell'uomo comprensibile alle macchine, tramite un linguaggio e dei comandi dettagliati basati sulla Teoria dei Linguaggi Formali e sulla Teoria delle Decisioni. Nel primo caso, la Teoria dei Linguaggi Formali utilizza approcci diversi che si rifanno

alle Teorie delle Stringhe e ai loro utilizzi come veri e propri linguaggi e le cui proprietà variano appunto a seconda del metodo utilizzato. Si può quindi decidere di puntare su un approccio o sull'altro a seconda dei risultati che si intende ottenere, ossia a seconda del tipo di risposta che si vuole dalla macchina nelle differenti situazioni.

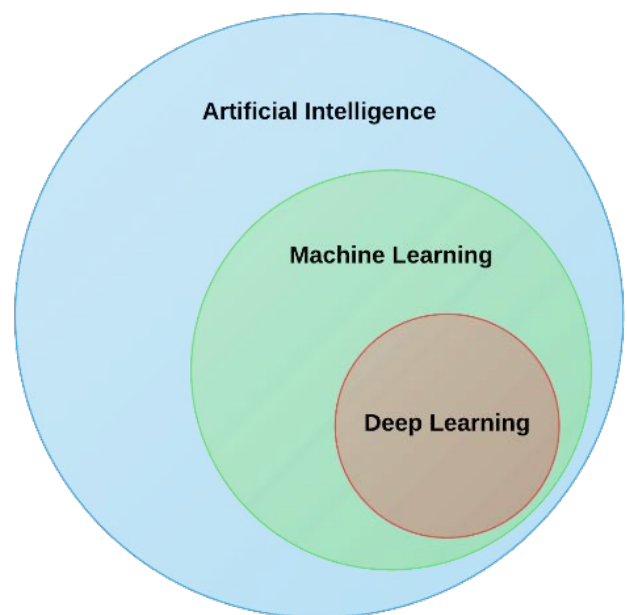
La Teoria delle Decisioni, invece, si basa su un albero decisionale che permette di valutare, per ogni azione/decisione, le possibili conseguenze e di scegliere, pertanto, la decisione più conveniente a seconda del tipo di piano di azioni definito dagli algoritmi della macchina.

Sviluppare algoritmi in grado di imparare dai propri errori risulta, pertanto, fondamentale per realizzare sistemi intelligenti che operino in contesti per i quali i programmatori non possono, a priori prevedere tutte le possibilità di sviluppo. [\(9\)](#)

Figura 3 Tipologie di Intelligenza Artificiale

L'Intelligenza Artificiale, attualmente, si approfondisce con due approcci all'interpretazione e allo studio dei dati, descrivibili con la discipline del Machine Learning e della sua ulteriore specializzazione

del Deep Learning. L'apprendimento automatico è una forma di intelligenza artificiale che migliora le proprie capacità e prestazioni nel tempo imparando automaticamente, con l'esperienza a svolgere determinati compiti. In sostanza, man mano che gli algoritmi acquisiscono i dati di training vengono prodotti modelli sempre più precisi che ne migliorano le performance.



A seconda della natura del problema affrontato è possibile individuare un differente approccio ai dati basato, principalmente, sul tipo, sulle caratteristiche e sul volume degli stessi.

In generale, le tecniche di machine learning si differenziano in supervisionato, non supervisionato e con tecniche di rinforzo che prevede un approccio del tutto differente introducendo il concetto di *ambiente* e di *sensation* che permettono, banalmente, la contestualizzazione del problema.

- **Supervised Learning:** ha lo scopo di trovare modelli nei dati che possano essere applicati ad un processo di analisi. Dato, quindi un set di variabili di input opportunamente etichettate, si individua un algoritmo per apprendere la funzione di mappatura dall'input all'output, in modo tale che, per ogni nuova variabile di input è possibile prevedere la variabile di output correttamente, anche per situazioni non presenti nel set di allenamento. Dato un insieme di immagini di mele, ad esempio, l'algoritmo è in grado di stabilire se il nuovo elemento in input sarà o meno una mela confrontandolo con le immagini, a sua disposizione, che ha imparato a riconoscere come mele grazie alle labels. La previsione è, quindi possibile solo grazie ad un training set opportunamente etichettato che prevede una supervisione esterna, molto spesso legata a studi e risultati ottenuti da umani, professionisti esperti di settore. L'algoritmo pecca quindi in esperienza e, pertanto non è adeguato per l'apprendimento in interazione.

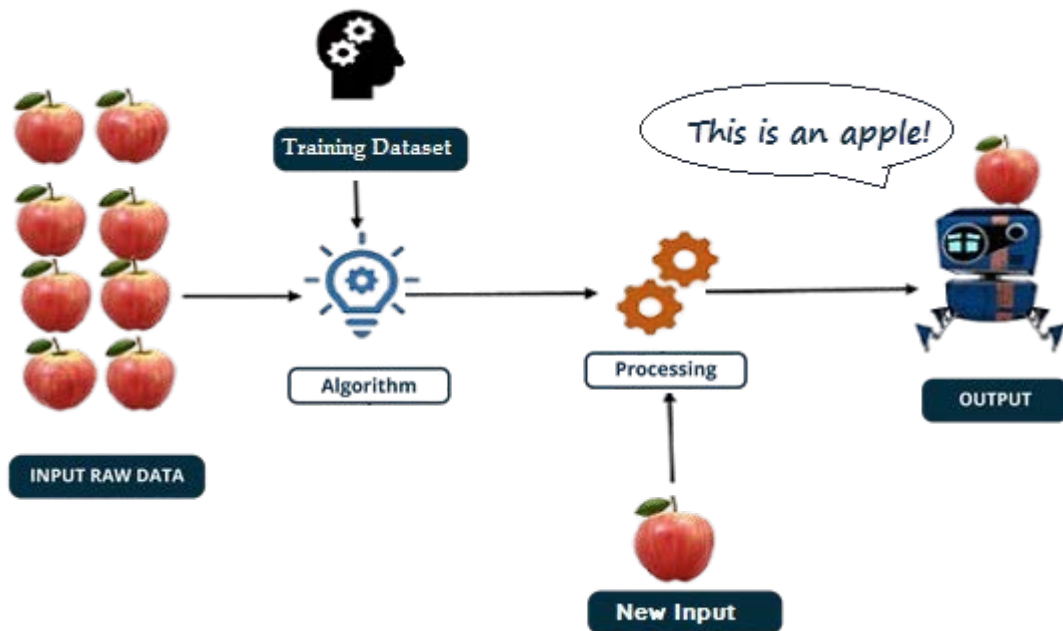


Figura 4: Apprendimento Supervisionato

Esso effettua, del resto, una generalizzazione in grado di applicare la conoscenza acquisita anche a dati non noti, ma nei casi in cui l'apprendimento è stato effettuato troppo a lungo o se c'è stato un scarso numero di esempi di allenamento, il modello potrebbe adattarsi a caratteristiche che sono specifiche solo del training set, ma che non hanno riscontro nel resto dei casi. Si parla quindi di overfitting che rappresenta un problema di calibratura risolvibile, in genere, con adeguati test su ben definiti set. Un esempio concreto di apprendimento supervisionato è rappresentato dalle previsioni meteorologiche in cui gli algoritmi vengono addestrati utilizzando esempi pre-elaborati costituiti da un insieme storico di modelli meteorologici noti e dalle condizioni attuali.

- Unsupervised Learning: conduce un processo iterativo di analisi senza alcun intervento umano, indagando sulle strutture nascoste in raccolte di dati senza etichetta. L'apprendimento avviene, quindi, man mano che gli elementi sono

introdotti, senza una conoscenza pregressa, ma sfruttando metriche pre-esistenti e mantenendo costante il peso dell'errore generato. Dato quindi un cesto di frutta, come in figura, l'algoritmo è in grado di stabilire che esiste una differenza tra banane, mele e arance senza, però conoscere nulla della frutta stessa, ma etichettandola in base ad associazioni, similitudini e accorpamenti.

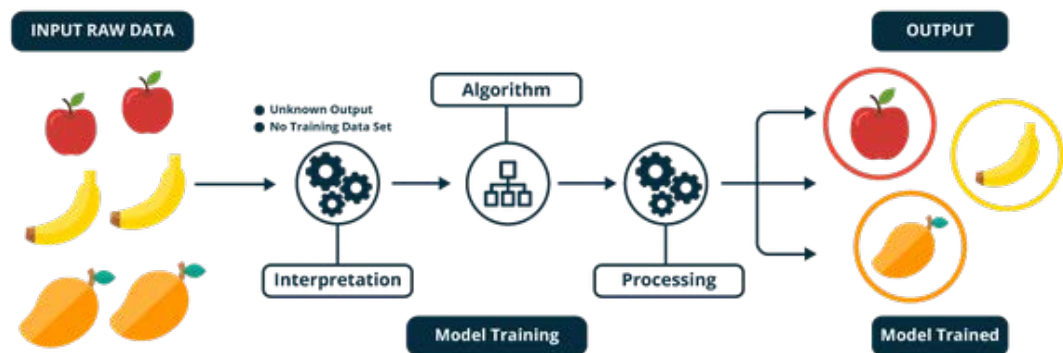


Figura 5 : Apprendimento Non Supervisionato

L'apprendimento non supervisionato è utilizzato, ad esempio per il rilevamento degli spam nelle e-mail.

- Reinforcement Learning: è un modello di apprendimento comportamentale in cui il sistema impara attraverso un meccanismo di trial-error, per cui una sequenza di decisioni di successo comporterà il "rafforzamento" del processo. Le decisioni sono supportate da un delayed reward che formalizza l'idea di obiettivo. Una delle applicazioni più comuni dell'apprendimento per rinforzo è la robotica. Nel tracciamento della traiettoria percorsa da un robot umanoide, ad esempio, lungo una rampa di scale, ogni volta che il robot cade, i dati acquisiti vengono ricalibrati così che, riprovando a salire la rampa, il robot esegue i passaggi in modo diverso. In altre

parole, il robot esplora l'ambiente e sfrutta ciò che ha acquisito come esperienza per migliorare le scelte di azioni future e per ottenere una ricompensa al raggiungimento del goal.

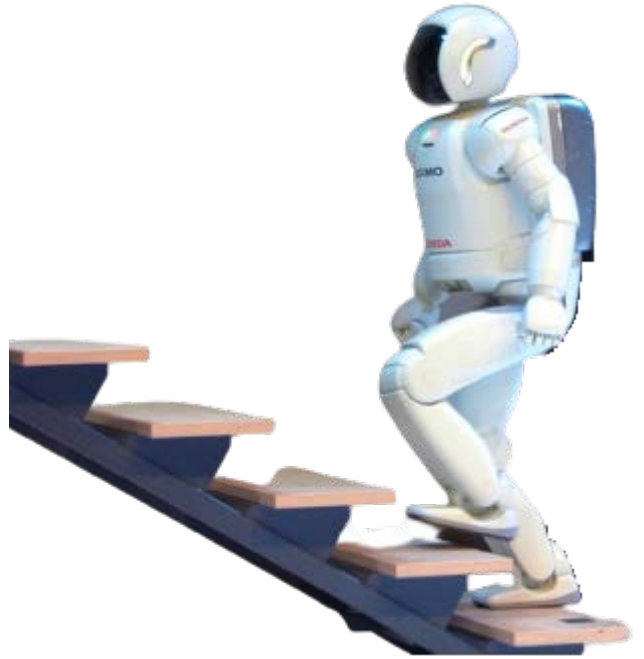


Figura 6 : Reinforcement Learning

comportamenti.

Durante l'apprendimento, quindi, l'algoritmo deve essere in grado di scoprire un'associazione tra l'obiettivo di salire le scale senza cadere e la sequenza di eventi che portano al risultato corretto.

- Deep Learning: è un metodo specifico di machine learning che incorpora reti neurali in strati successivi per apprendere dai dati in modo iterativo. L'apprendimento profondo è particolarmente utile quando si cerca di costruire modelli da dati non strutturati. Le reti neurali complesse, in generale sono progettate per emulare il funzionamento del cervello umano, in modo che i calcolatori possano essere addestrati ad affrontare astrazioni e problemi mal definiti. Si tratta di algoritmi che usano, in cascata, vari livelli di unità non lineari per svolgere compiti che possono essere classificati come *problemi di trasformazione di caratteristiche estratte dai dati*

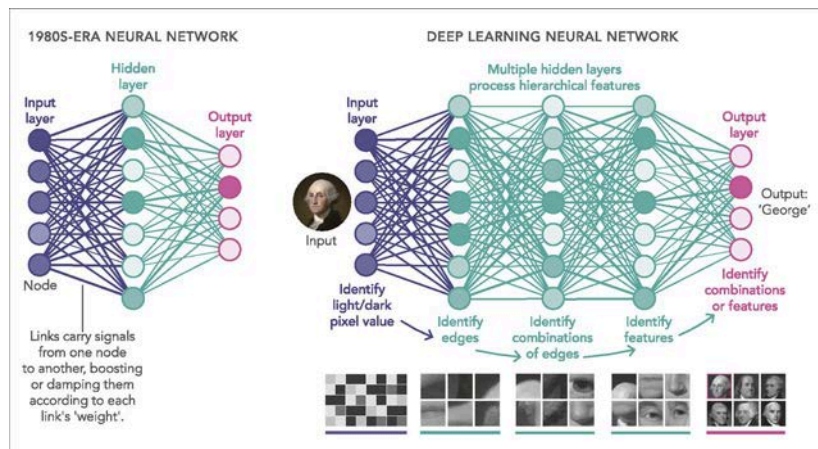
e ciascun livello

Figura 7 : Deep Learning

utilizza l'uscita del

livello precedente come input in modo da formare una gerarchia di concetti. [\(10\)](#)

Una rete neurale profonda è composta da tre o più livelli: un livello di input, uno o più livelli hidden e un livello di output. I dati vengono inseriti attraverso il livello di input



e modificati nel layer nascosto e nel layer di output in funzione dei pesi applicati ai suoi nodi. Una semplice rete neurale può essere costituita da

milioni di nodi di elaborazione densamente interconnessi. Usando un approccio iterativo, la rete si regola e crea continuamente inferenze, fino al raggiungimento di uno specifico punto di arresto. Più complesso è il problema, più livelli concettuali nascosti ci saranno nel modello. Seppure la richiesta di capacità computazionali enormi possa rappresentare un limite, la scalabilità del Deep Learning all'aumento dei dati disponibili e degli algoritmi è ciò che lo differenzia dagli algoritmi di Machine Learning tradizionali: i sistemi di Deep Learning, infatti, migliorano le proprie prestazioni all'aumentare dei dati, mentre le applicazioni di Machine Learning, una volta raggiunto un certo livello di performance non sono più scalabili, nemmeno aggiungendo un numero maggiore di esempi e dati di training. Affinché l'output finale dell'apprendimento profondo sia soddisfacente sono necessarie enormi quantità di dati; pensare, però, subito ai Big Data⁴ è un errore, in quanto almeno nella parte iniziale dell'addestramento, il sistema ha bisogno di dati etichettati e strutturati, perciò l'eterogeneità dei Big Data non può considerarsi una soluzione. I dati non strutturati possono essere analizzati da un modello di apprendimento profondo solo

⁴ Intesi letteralmente come vasta quantità, in termini di volume, di dati non strutturati, di grande varietà e di facile e veloce reperibilità.

una volta formato e raggiunto un livello accettabile di accuratezza. Questo tipo di tecnologia è usato nell'elaborazione delle immagini e nel riconoscimento di tracce audio o video. [\(11\)](#)

2.1 Tecniche di Machine Learning

Selezionare l'algoritmo giusto è in parte una scienza e in parte un'arte. Due *data scientists* incaricati di gestire la stessa sfida possono scegliere algoritmi diversi per risolvere lo stesso problema. Tuttavia, comprendere i diversi paradigmi di apprendimento automatico può essere utile ad individuare il miglior algoritmo di risoluzione. [\(11\)](#)

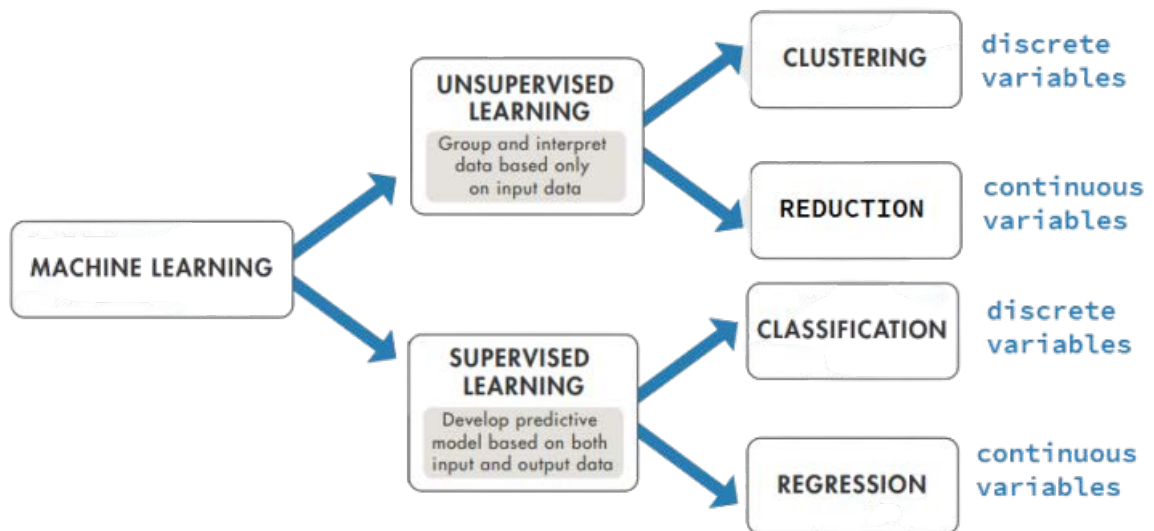


Figura 8: L'apprendimento supervisionato genera un modello da dati di input e output conosciuti mentre l'apprendimento senza supervisione individua schemi nascosti o strutture intrinseche nei dati di input

Nell'apprendimento supervisionato la macchina deve stimare una funzione $f(x)$ incognita che collega le *features* di input x a una variabile *target* di output y , dove $y=f(x)$. La macchina non conosce la funzione $f(x)$ ed il suo scopo è di stimare una *funzione d'ipotesi* $h(x)$ in grado di approssimare $f(x)$ in modo che $y=h(x)$. Per farlo analizza il training set fornito dal supervisore esperto esterno e costituito da N coppie di esempi input-output $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$ in cui y_i funge da *label* per x_i . A partire da questi dati, il calcolatore elabora la funzione $h(x)$ la cui accuratezza è valutata attraverso il test set, fornito anch'esso

dal supervisore esterno esperto. Ogni risposta R risultata dalla macchina viene confrontata con la risposta corretta Y indicata dagli esperti nel test set.

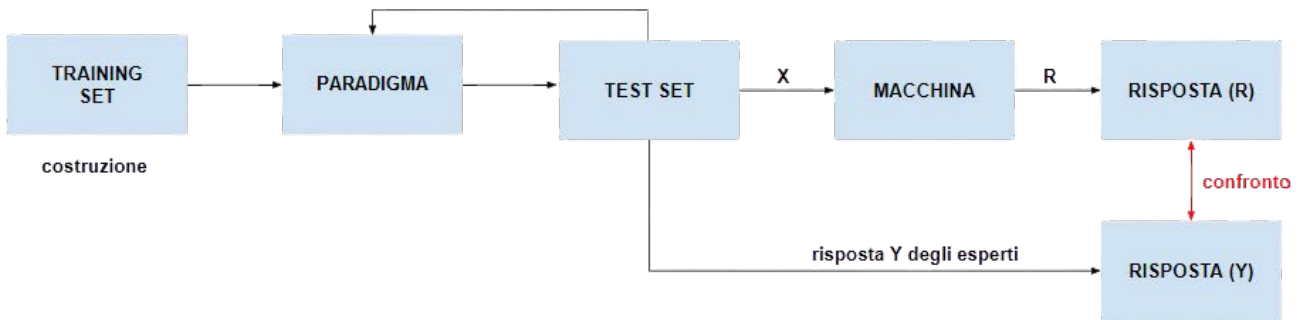
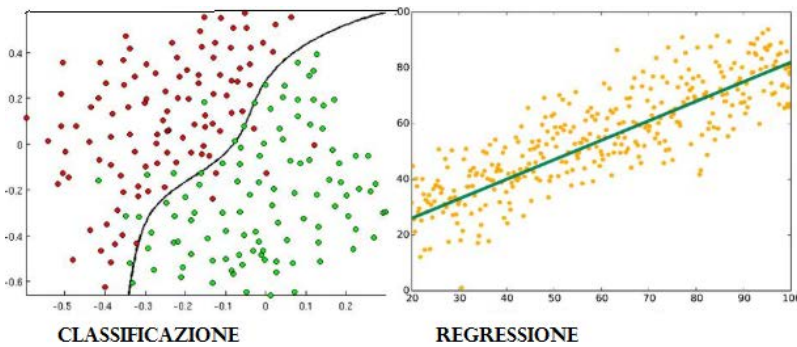


Figura 9: Algoritmo di apprendimento supervisionato

Le risposte coincidenti per cui $R=Y$ incrementano il numero delle risposte corrette R_c della macchina. Se la percentuale di risposte corrette

$$\frac{R_c}{N_t} = \frac{\text{Risposte corrette}}{\text{Numero di test}}$$

è soddisfacente, allora la funzione d'ipotesi $h(x)$ supera l'esame e viene accolta. Quando l'output y fa parte



di un insieme finito di valori e le caratteristiche d'ingresso sono un vettore di numeri reali x allora

$$y = f(\mathbf{w} \cdot \mathbf{x}) = f\left(\sum_j w_j x_j\right)$$

dove \mathbf{w} è un vettore reale di pesi che

mappa \mathbf{x} in \mathbb{R}^5 ed è appreso dai campioni d'allenamento etichettati. Si tratta, quindi, di un problema di *Classificazione lineare* in cui $f(\cdot)$ fornisce la probabilità che un elemento appartenga a una certa classe.

Nel caso di due sole classi la classificazione si definisce booleana o binaria e, semplicemente mappa tutti i valori sopra una certa soglia alla prima classe e tutti gli altri valori alla seconda classe. Nei problemi di classificazione supervisionata, quindi, ad ogni pattern x_i del training set è associata un'etichetta y_i che ne indica la classe di appartenenza. Quando i valori y_i sono numeri reali e l'obiettivo del training è

⁵ Insieme dei numeri reali

l'apprendimento di una funzione $f(\mathbf{x}) \rightarrow y$ si parla di *Regressione*, in cui \mathbf{x} è detta variabile indipendente e y variabile dipendente. Si assume, in genere, che la variabile indipendente sia esatta, mentre la variabile dipendente sia affetta da errore. Se la variabile indipendente è uno scalare x si ha una *simple linear regression* in cui

$$y_i = \alpha + \beta \cdot x_i + \varepsilon_i$$

dove α e β sono parametri da determinare e ε_i è l'errore di misura, non noto, del pattern x_i .

In questo caso f diventa l'equazione di una retta:

$$f(x) = y = \alpha + \beta \cdot x$$

Nel caso di *multiple linear regression* in cui \mathbf{x} è un vettore, f approssimerà un iperpiano di regressione.

(4)

2.1.1 Alberi Decisionali

I decision trees sono modelli di apprendimento automatico supervisionato che mappano le funzioni di classificazione e regressione in un albero binario di nodi interni e nodi foglia opportunamente etichettati.

In particolare, gli archi provenienti da un nodo interno etichettato con una input feature sono etichettati, a loro volta, con ciascuno dei possibili valori della target feature e conducono a un nodo decisionale subordinato, su una diversa input feature.

La label dei nodi foglia è una classe o una distribuzione di probabilità sulle classi, il che significa che il set di dati può essere classificato dall'albero in una classe specifica, per la regressione, o in una particolare distribuzione di probabilità, per la classificazione.

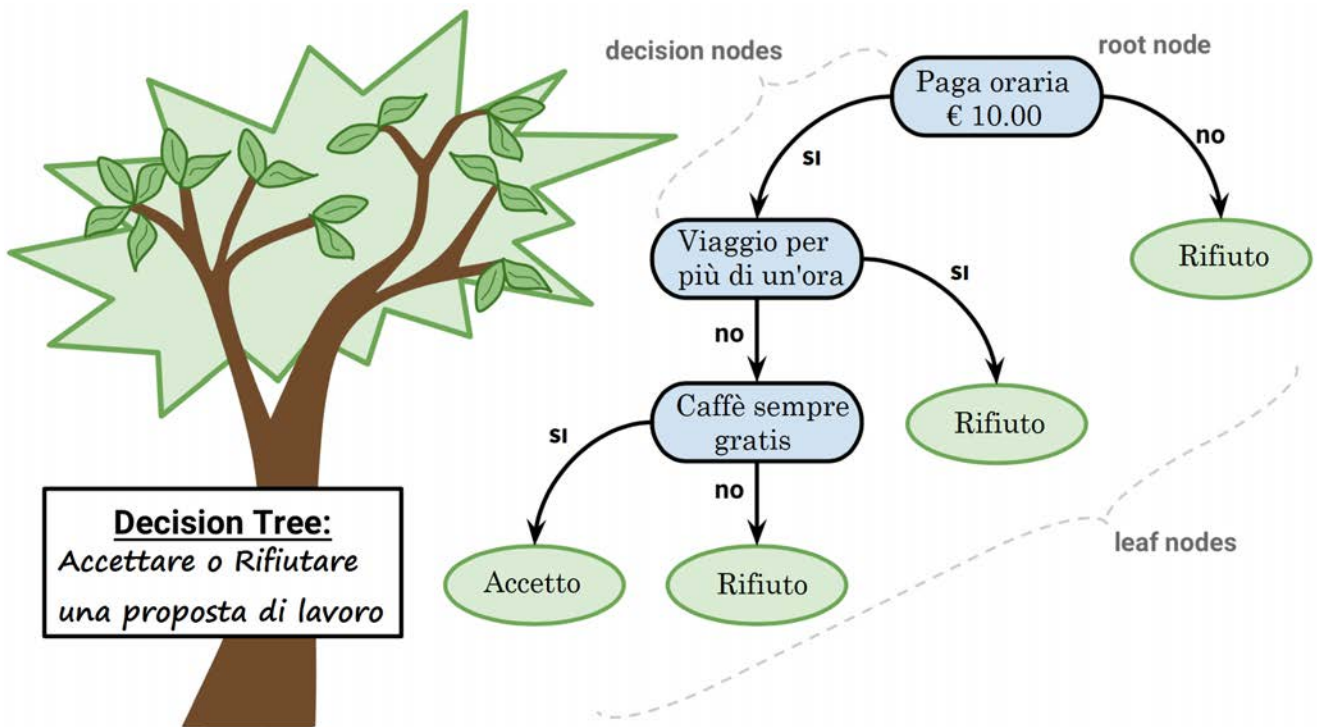


Figura 10: Esempio di utilizzo di un albero decisionale per la risoluzione di un problema di valutazione

Dalla root, il partizionamento avviene sulla base del valore dell'attributo. Ogni nodo interno, in pratica è associato ad una particolare "domanda" su una caratteristica e, da questo nodo dipartono tanti archi quanti sono i possibili valori che la feature può assumere, fino a raggiungere le foglie che rappresentano l'output. Una buona "domanda" divide i campioni di classi eterogenee in sottoinsiemi con etichette abbastanza omogenee, stratificando i dati in modo da mettere poca varianza in ogni strato, mediante un'adeguata metrica che misura le impurità. Questo processo top-down viene ripetuto su ciascun sottoinsieme con un *partizionamento ricorsivo* che termina quando il sottoinsieme in un nodo ha tutti gli stessi valori della variabile di destinazione o quando la divisione non aggiunge più valore alle previsioni.

(12) Lo split dei nodi si basa su una serie di regole di suddivisione che utilizzano metriche diverse per misurare "la migliore variabile di split" in modo da ottimizzare l'omogeneità della variabile target all'interno dei sottoinsiemi. Queste metriche vengono applicate a ciascun sottoinsieme candidato e i valori risultanti vengono combinati per fornire una misura della qualità della divisione.

Due metriche principali sono:

:

- Entropia:

$$I_H = \sum_{i=1}^m p_i \log_2 p_i$$

dove X è il dataset formato da m classi e $p_i = p(x_i)$ è la frequenza relativa della classe i all'interno dell'insieme X per ogni elemento x_i .

In sostanza implica la probabilità di distribuzione dei dati all'interno del dataset in modo che "Entropia elevata" indichi che X abbia una distribuzione uniforme, detta anche noiosa, mentre "Entropia bassa" che X abbia una distribuzione variegata con alti e bassi.

- Indice di Gini :

$$I_G(X) = 1 - \sum_{i=1}^m p_i^2$$

indica la disuguaglianza tra i valori ed è un numero compreso da 0 a 1 in cui più basso è il valore, più uguale è la distribuzione, quindi $I_G(X) = 0$ implica che i valori sono praticamente gli stessi, viceversa $I_G(X) = 1$

Una "domanda" $h_j(x)$, che ha k possibili risposte, divide l'insieme E nei sottoinsiemi E_1, \dots, E_k . Per testare quanto bene la condizione viene eseguita, bisogna confrontare il grado di impurità dei nodi figli con l'impurità del nodo padre: maggiore è la loro differenza, migliore è la condizione scelta. Data una metrica $I(\cdot)$ che misuri l'impurità, il guadagno Δ è un criterio che può essere usato per determinare la bontà della divisione:

$$\Delta = I(E) - \sum_{i=1}^k \frac{N(E_i)}{N(E)} I(E_i)$$

dove $N(E)$ è il numero di campioni nel nodo padre e $N(E_i)$ è il numero di campioni nel nodo figlio i -esimo.

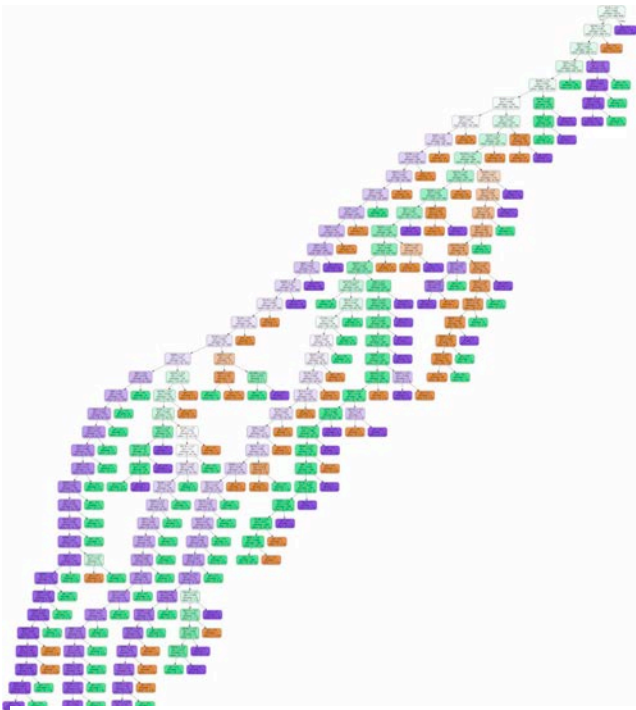


Figura11: esaminando e condensando i dati di addestramento in un albero binario di nodi interni e nodi foglia gli alberi decisionali risultano simili a diagrammi di flusso in cui un nodo interno rappresenta un attributo e il ramo rappresenta una regola decisionale

Se viene usata come metrica l'entropia, il guadagno Δ è conosciuto come *Information Gain* e indica il grado di informazione associato ad ogni attributo.

Gli alberi di decisione inducono algoritmi che scelgono una condizione di test che massimizza il guadagno Δ e, siccome $I(E)$ è uguale per tutti i possibili classificatori e $N(E)$ è costante, massimizzare il guadagno è equivalente a

minimizzare la somma pesata delle impurità dei nodi figli e, quindi una buona "domanda" $h_j(x)$ è quella che minimizza tale quantità.

Nel caso di classificatori binari, la metrica di Gini è ampiamente utilizzata, in quando il guadagno da minimizzare si riduce a

$$\frac{p_1 n_1}{p_1 + n_1} + \frac{p_2 n_2}{p_2 + n_2}$$

con p_1, n_1 numero di campioni positivi e negativi che il classificatore sposta nel ramo sinistro e p_2, n_2 numero di campioni positivi e negativi spostati nel ramo destro.

A seconda del modello di split gli alberi decisionali si distinguono in binari e non binari, mentre in base alla geometria del vincolo si differenziano in

- ortogonali : il vincolo è rappresentato graficamente da una linea parallela ad una delle dimensioni e lo spazio di ricerca si divide in sotto-parti delimitate da vincoli paralleli agli assi;
- obliqui: la condizione è definita a mezzo di una combinazione lineare di un sottoinsieme delle dimensioni e lo spazio di ricerca si divide in sotto-parti delimitate da segmenti non paralleli agli assi.

Questi tipi di classificatori sono facili da implementare e da capire in quanto offrono una rappresentazione grafica intuitiva; risultano, inoltre, veloci e applicabili a differenti tipi di problematiche,

ma sono soggetti a overfitting, comunque risolvibile, di solito settando i vincoli o facendo pruning dei rami.

2.1.2 Reti Neurali

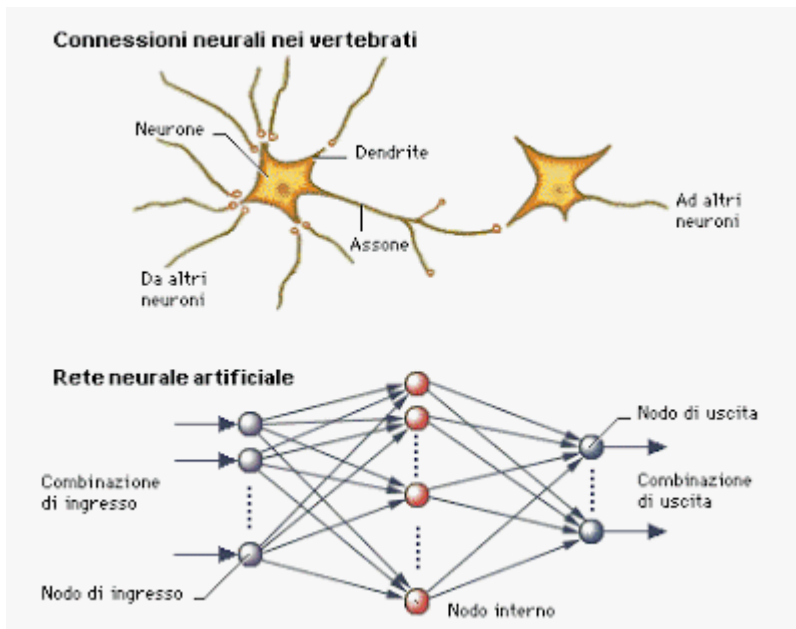


Figura 12: Le Reti Neurali nascono dall'idea di riprodurre alcune delle funzioni e capacità del cervello umano e tentano di imitarne la struttura e le funzionalità.

Un neurone biologico somma i suoi dati in ingresso e produce dei dati in uscita, se la somma di tale dati risulta maggiore di un valore di soglia, allora il neurone trasmette l'informazione ai successivi neuroni attraverso le giunzioni sinaptiche

collegate tra loro attraverso un link, le cui caratteristiche sono determinate dalla topologia e dalle proprietà dei "neuroni" stessi.

I link di collegamento tra l'unità i e l'unità j propagano l'eccitazione a_i da i a j con forza e segno ponderati da un peso numerico $w_{i,j}$ associato ad ogni connessione e su cui si agisce durante l'apprendimento. Nella maggior parte dei casi una rete neurale artificiale è un sistema adattivo che cambia la propria struttura in base ad informazioni, esterne o interne, che scorrono attraverso la rete stessa durante la fase di apprendimento. Questo la rende particolarmente indicata sia al

Le reti neurali artificiali si ispirano alla neuroscienza per la quale l'attività mentale principale consiste nell'elettrochimica dei neuroni e, così come in natura, il modello computazionale "spara" quando una combinazione lineare dei suoi input supera soglie prestabilite, alte o soft, implementando, in questo modo, un classificatore lineare.

Una neural network, quindi è una raccolta di unità di elaborazione

riconoscimento di regolarità, che allo studio dei sistemi complessi, per i quali diventa strumento di predizione tramite un processo di simulazione⁶.

Strutturalmente, ogni unità, j ha un input fittizio $a_0 = 1$ con un peso ad esso associato $w_{0,j}$ detto bias, utile a “tarare” il punto di lavoro ottimale del neurone e dal quale calcolare la somma ponderata dei suoi input:

$$in_j = \sum_{i=0}^n w_{i,j} a_i$$

Su tale somma si applica una funzione di attivazione g per ottenere l'output:

$$a_j = g(in_j) = g(\sum_{i=0}^n w_{i,j} a_i)$$

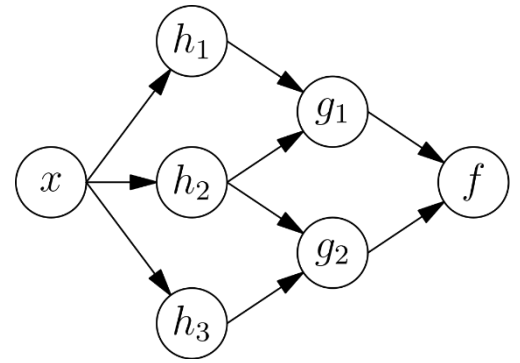
che determina il comportamento del neurone in funzione, appunto, del suo livello di eccitazione e che viene propagata nella rete, fino al termine del processo.

La funzione di attivazione g è in genere una soglia rigida chiamata *perceptrone* o a volte una funzione logistica, nel qual caso si può usare il termine *sigmoid perceptron*. Nei neuroni biologici g è una funzione *tutto-niente temporizzata* per cui il neurone “spara” uno spike per poi tornare a riposo. Le reti neurali più comunemente utilizzate operano con livelli continui e g è una funzione non-lineare, in grado di eseguire un mapping complesso dell'informazione di input, continua, per la retro-propagazione dell'errore e differenziabile quasi ovunque. [\(4\)](#)

Definito il modello matematico per i singoli "neuroni", il passo successivo è collegarli insieme per formare una rete. Ci sono due modi fondamentalmente distinti per farlo:

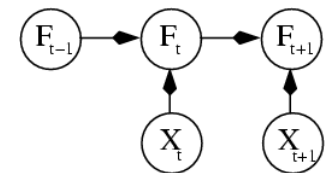
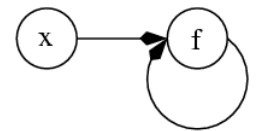
⁶ Il metodo della simulazione definisce un modello della realtà, in particolare di fenomeni su cui non si può sperimentare per ragioni di costi, etiche o di spazio/tempo, che consente di valutare e prevedere lo svolgersi dinamico di una serie di eventi, susseguenti all'imposizione di certe condizioni, da parte dell'analista. La scienza tradizionale, invece procede formulando, con il proprio simbolismo teorie e ipotesi, verificandone l'esattezza in esperimenti di laboratorio. [\(104\)](#)

- feed-forward network: le connessioni sono in una sola direzione con un grafico aciclico diretto. Ogni nodo riceve input dai nodi "upstream" e fornisce output ai nodi "downstream" non creando anelli. Una rete feed-forward rappresenta una funzione dei suoi stessi input correnti e, pertanto non ha uno stato interno diverso dai pesi;



- recurrent networks: reinserisce i propri output nei propri input con connessioni di feedback in genere, previste verso neuroni dello stesso livello, ma anche all'indietro. Ciò significa che i livelli di attivazione della rete

formano un sistema dinamico che può raggiungere uno stato stabile, mostrare oscillazioni o persino comportamenti caotici. Inoltre, la risposta della rete a un dato di input dipende dal suo stato iniziale, che, a sua volta può dipendere da input precedenti. Per



questo le reti ricorsive, diversamente dalle reti feed-forward possono supportare la memoria a breve termine che al tempo t rende disponibile l'informazione processata a $t - 1$, $t - 2$, etc. [\[13\]](#) L'addestramento, d'altro canto richiede di considerare il comportamento in più istanti temporali (*unfolding in time*). Queste caratteristiche, la memoria e l'unfolding rendono queste reti indicate per la gestione di sequenze, come tracce audio o video, Un particolare tipo di rete ricorrente è Long Short Term Memory.

Il percettore utilizza una funzione di attivazione lineare a soglia o scalino.

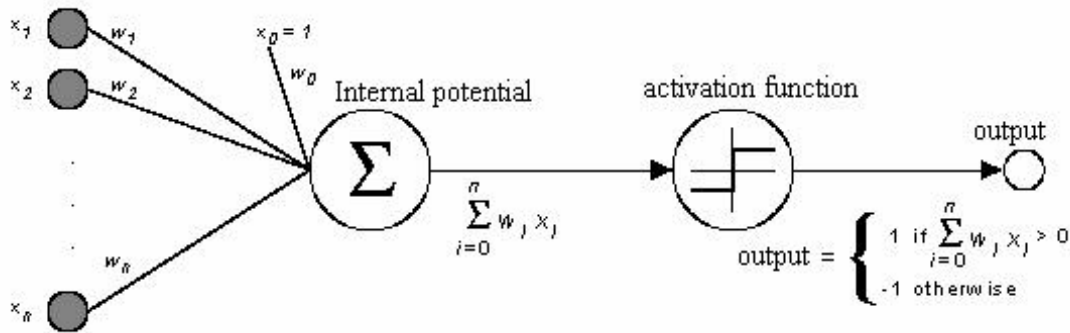


Figura 14: Percettrone semplice

Una rete a due livelli di percettroni lineari a soglia è in grado di apprendere solo mapping lineari e pertanto il numero di funzioni approssimabili è piuttosto limitato. Tuttavia, pur non essendo uno strumento molto potente, un semplice percettrone può rappresentare alcune funzioni booleane piuttosto "complesse" in modo molto compatto. Ne è un esempio, *la funzione maggioritaria*, che genera 1 solo se più della metà dei suoi n input sono 1 e che può essere rappresentati semplicemente imponendo ogni $w_i = 1$ e $w_0 = -n / 2$ nella funzione del percettrone, lì dove un albero decisionale avrebbe bisogno, esponenzialmente, di molti nodi per rappresentare questa funzione.

Un Multilayer Perceptron (MLP) è una rete feedforward con almeno 3 livelli, quindi 1 hidden e con funzioni di attivazione non lineari. Un teorema noto come *universal approximation theorem* asserisce che ogni funzione continua che mappa intervalli di numeri reali su un intervallo di numeri reali può essere approssimata da un MLP con un solo hidden layer. In effetti, con un singolo strato nascosto sufficientemente grande, è possibile rappresentare qualsiasi funzione continua degli ingressi con precisione arbitraria; mentre con due strati, possono essere rappresentate anche funzioni discontinue.

Fissata la topologia della rete, il training consiste nel determinare il valore dei pesi \mathbf{w} per ottenere la mappatura desiderata tra input e output. La differenza tra l'output prodotto della rete e quello desiderato è definito *errore della rete* $e = \mathbf{y} - h_{\mathbf{w}}(\mathbf{x})$. Obiettivo dell'algoritmo di apprendimento è di

modificare i pesi della rete in modo da minimizzare l'errore medio sui pattern del training set. In generale una rete perceptron si decompone in m separati problemi di apprendimento per ottenere un vettore target di dimensione m , mentre nel caso delle multilayer networks – in cui la rete è implementata con una funzione target vettoriale \mathbf{h}_w piuttosto che scalare h_w – questa decomposizione fallisce, perché ogni output dipende da tutti i pesi del livello di input, quindi gli aggiornamenti degli stessi dipenderanno da errori propagatesi tra tutti i nodi. Fortunatamente, questa dipendenza è molto semplice nel caso di Funzioni di Costo (*Loss function*) Additive tra i componenti di errore.

$$J(w, x) = \sum_k \frac{\partial}{\partial w} (y_k - a_k)^2$$

dove l'indice k si estende sui nodi nel livello di output. Ogni termine nella sintesi finale rappresenta il gradiente della perdita per il k -esimo output, calcolato come se gli altri output non esistessero. Quindi, possiamo scomporre un problema di apprendimento m -output in m problemi di apprendimento, purché ricordiamo di sommare i contributi dei gradiente di ciascuno di essi durante l'aggiornamento dei pesi.

Con l'introduzione della Loss Function è possibile risolvere anche l'errore propagatosi attraverso nuovi strati nascosti aggiunti alla rete. Si evince, infatti che è possibile effettuare una back-propagate dell'errore dall'output agli strati nascosti. L'idea è che il nodo hidden k sia "responsabile" di una frazione dell'errore Δ_k in ciascuno dei nodi di output a cui si connette. Pertanto, i valori di Δ_k sono divisi in base alla forza della connessione tra il nodo nascosto e il nodo di output e vengono propagati indietro per fornire i valori Δ_j per il livello nascosto. La regola di propagazione per Δ_j valori è la seguente:

$$\Delta_j = g'(in_j) \sum_k w_{j,k} \Delta_k$$

La regola di aggiornamento per i pesi tra gli input e il layer nascosto è essenzialmente identica alla regola di aggiornamento per il layer di output:

$$w_{i,j} \leftarrow w_{i,j} + \alpha \times a_i \times \Delta_j.$$

Il processo di propagazione retroattiva può essere riassunto come segue:

- Calcolare i valori Δ per le unità di uscita, utilizzando l'errore osservato.
- A partire dal livello di output, ripetere quanto segue per ciascun livello della rete, fino a quando viene raggiunto il primo livello nascosto:
 - Propagare i valori Δ al livello precedente.
 - Aggiorna i pesi tra i due strati.

Una volta addestrata, una rete neurale può semplicemente processare pattern attraverso la forward propagation che descrive il passaggio delle informazioni in avanti dal livello di input a quello di output.

L'ultimo step è individuare la migliore struttura di rete. Una rete troppo grande, infatti sarà in grado di memorizzare tutti gli esempi, formando una grande tabella di ricerca, ma non si generalizzerà necessariamente bene agli input che non ha mai visto prima, generando, quindi un problema di overfitting.

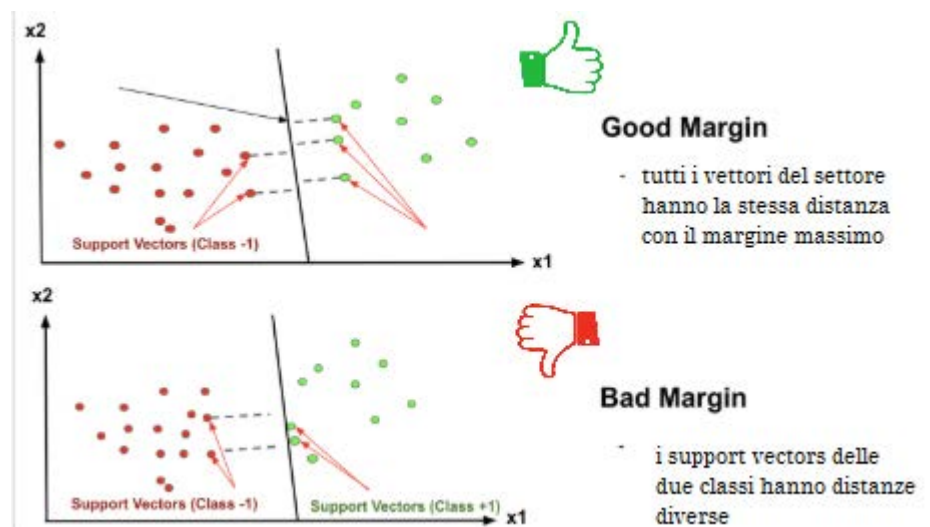
Se ci si attiene a reti completamente connesse, le uniche scelte da fare riguardano il numero di strati nascosti e le loro dimensioni. L'approccio abituale è provarne un numero diverso e mantenere il migliore scegliendo l'architettura di rete che offre la massima precisione di previsione. Per reti che non sono completamente connesse, è necessario, invece trovare un metodo di ricerca efficace attraverso l'ampio spazio di possibili topologie di connessione.

2.1.3 Support Vector Machine

Sono modelli di apprendimento automatico supervisionato che effettuano una classification boundaries determinando la migliore linea in due dimensioni o un iperpiano 'best fit' in più di due dimensioni, come superfici decisionali tra le classi.

Tali superfici di delimitazione si individuano attraverso i support vectors che sono i punti che

rappresentano il margine massimo, cioè la distanza massima tra i valori di entrambe le classi e che contribuiscono al risultato dell'algoritmo. Se un punto non è un vettore di supporto, la sua rimozione



non ha alcun effetto sul modello. D'altra parte, l'eliminazione dei vettori di supporto cambierà la posizione dell'iperpiano. I punti vengono mappati dallo spazio dati a uno spazio di features ad alta dimensione utilizzando una funzione kernel [\(14\)](#). La massimizzazione del margine è legata alla generalizzazione e se i pattern del training set sono classificati con ampio margine si può "sperare" che anche i pattern del test set, vicini al confine tra le classi siano gestiti correttamente. Più formalmente, una macchina vettoriale di supporto costruisce un iperpiano, o un insieme di iperpiani, in uno spazio di dimensione alta o infinita, che può essere utilizzato per la classificazione, la regressione o altre attività come il rilevamento di valori anomali. [\(15\)](#)

Nello spazio delle funzioni kernel, l'algoritmo cerca la sfera più piccola che racchiude l'immagine dei dati e questa sfera, quando viene mappata nello spazio dati, forma un insieme di contorni che racchiudono i punti. Questi contorni vengono quindi interpretati come limiti del cluster e i punti racchiusi da ciascuno di essi sono associati dall'algoritmo allo stesso cluster.

Date due classi di pattern, linearmente separabili e un training set contenente n campioni $x_1, y_1 \dots x_n, y_n$, pattern multidimensionali etichettati con $y_i \in +1, -1$ e etichette delle due classi, esistono diversi iperpiani in grado di eseguire la separazione voluta. Un generico iperpiano è definito dai parametri (\mathbf{w}, b) :

$$D \mathbf{x} = \mathbf{w} \cdot \mathbf{x} + b$$

Dove

\mathbf{w} : vettore normale all'iperpiano,

$b / \|\mathbf{w}\|$: distanza dall'origine,

$(\mathbf{x}) = 0$: luogo dei vettori sul piano.

La distanza di un vettore \mathbf{x} dall'iperpiano vale pertanto: $r = \frac{D(\mathbf{x})}{\|\mathbf{w}\|}$

Gli iperpiani (\mathbf{w}, b) che separano i pattern del training set, con distanza minima $1/\|\mathbf{w}\|$ su ogni lato, soddisfano, per $i = 1 \dots n$, le equazioni:

$$\mathbf{w} \cdot \mathbf{x}_i + b \geq +1 \text{ se } y_i = +1$$

$$\mathbf{w} \cdot \mathbf{x}_i + b \leq -1 \text{ se } y_i = -1$$

o in modo più compatto:

$$y_i \mathbf{w} \cdot \mathbf{x}_i + b \geq 1 \text{ per } i = 1 \dots n$$

Nella classificazione di tipo binario, in cui la classe y può assumere solo due valori, $y_i = \{+1, -1\}$ il classificatore migliore è quello che divide linearmente il training set mediante un iperpiano

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

in cui i campioni si separano perfettamente sui due lati

I Support Vectors, sono i punti per cui

$$y_i(x_i \cdot w_0 + b_0) - 1 = 0$$

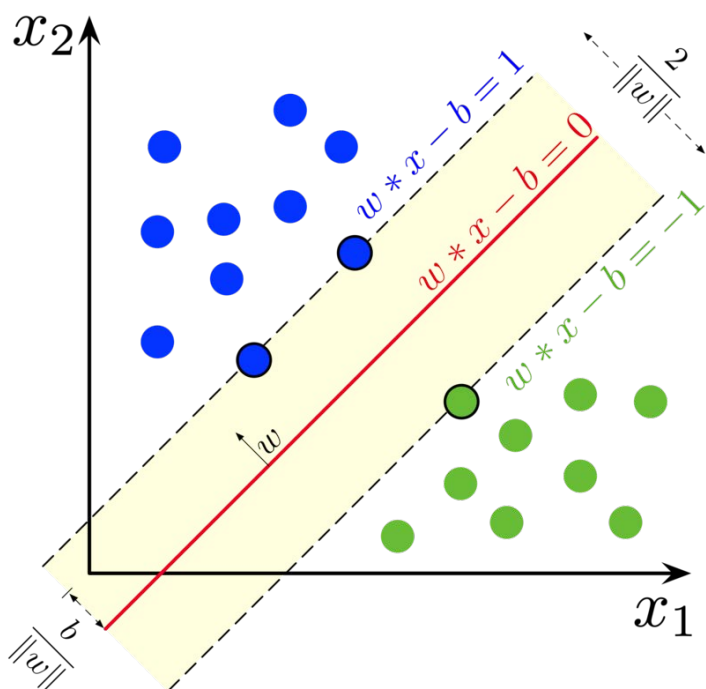
e quindi i più estremi della distribuzione e la loro distanza rappresenta la misura del margine di separazione tra le due categorie determinando il margine di separazione tra le stesse. La massimizzazione di tale distanza permette di trovare l'iperpiano ottimo.

Il vettore dei pesi viene modificato in modo da minimizzare l'errore

$$E(w) = \frac{1}{2} \sum_{t_k \in T} (t_k - y_k)$$

dove $T = \{x_k, t_k\}_{k=1}^n$ è il training set e $y_k = \text{sgn}(w \cdot x_k)$

tramite la discesa del gradiente che prevede di spostarsi sulla superficie di errore verso il minimo con la direzione determinata dalla derivata di E rispetto a ciascuna componente di w .



Se nel training set sono presenti dei campioni anomali, nel senso che pur appartenendo ad una classe si trovano nel semipiano sbagliato, la classificazione non produce un risultato corretto e ciò che si ottiene è una soluzione con modelli poco efficienti. In queste situazioni si parla di dati non linearmente separabili e per consentire la separazione viene sfruttata una funzione kernel che

Figura 15: Classificazione binaria

individua alcuni campioni dell'insieme di addestramento, come informazioni utili per capire quanto vicino a loro sia il campione di valutazione in esame.

I kernel più diffusi, in quanto semplici da valutare, sono:

- I kernel gaussiani (rbf) nella forma:

$$k(\mathbf{x}, \mathbf{x}') = e^{-\gamma \|\mathbf{x} - \mathbf{x}'\|^2}$$

con γ parametro da impostare,

- i kernel polinomiali di grado d nella forma:

$$k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}' + 1)^d$$

Per $d = 1$ la formulazione si riconduce al caso lineare.

2.1.4 Classificatori a Regole

Nel Unsupervised Learning i dati consistono soltanto di un insieme di features x_i , senza alcuna etichetta, y_i , pertanto la macchina deve estrapolare informazioni dal training set apprendendo dall'esperienza, senza il supporto né di esempi né di risposte di riferimento. Il processo di apprendimento avviene, quindi, man mano che gli elementi sono introdotti, in base ad analogie e metriche preesistenti di ripetizioni di itemset o per associazione e divisione in cluster. Se, quindi, i dati non prevedono un ordinamento cercare di ordinarli, imponendo una graduatoria arbitraria, non risolve il problema. Questo si può facilmente capire con un esempio. Supponiamo di disporre di un database con l'elenco dei colori utilizzati da uno stilista. Si potrebbe cercare di associare ad ogni colore uno specifico numero e su quello fare delle analisi

di tipo statistico. Ma dato che l'associazione tra colore e numero è arbitraria si possono pensare ad infinite associazioni che darebbero infiniti risultati diversi. Questi algoritmi, in sostanza lavorano correttamente in presenza di dati contenenti un ordinamento o un raggruppamento netto e chiaramente identificabile. [\(11\)](#) Nonostante l'ordinamento, spesso accertare la validità delle inferenze, tratte dall'output, della maggior parte degli algoritmi di apprendimento senza supervisione richiede ulteriori argomenti euristici, non solo per motivare la scelta dell'algoritmi stessi, - come può avvenire anche nel caso dell'apprendimento supervisionato -, ma anche per fornire giudizi di qualità dei risultati. [\(16\)](#)

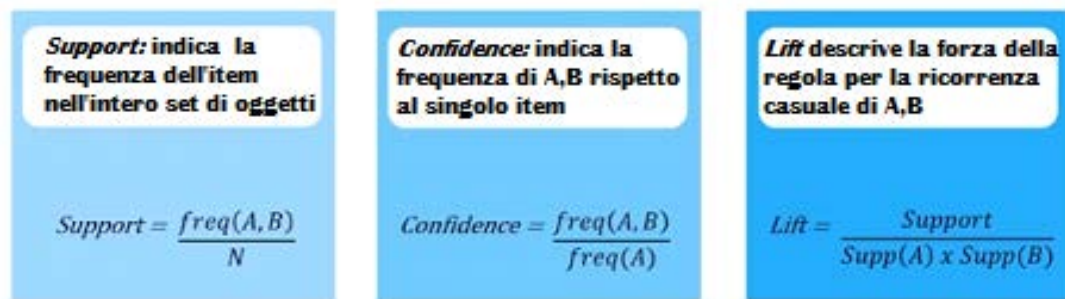


Figura 16. Regole associative

L'analisi delle regole di associazione è emersa, di conseguenza, come strumento popolare per il data mining. L'obiettivo è trovare valori congiunti delle variabili $X = (X_1, X_2 \dots X_n)$ che compaiono più frequentemente nel database. Nel caso specifico di $X_j \in \{0, 1\}$, si parla di "Analisi del paniere di mercato" e le osservazioni sono transazioni di vendita, come quelle che si verificano alla cassa di un negozio. Le variabili, in pratica rappresentano tutti gli articoli venduti e ad ognuna delle variabili di X viene assegnato uno di due valori $x_{i,j} = 1$ se il prodotto è stato acquistato $x_{i,j} = 0$ se non è stato acquistato. Le variabili che hanno spesso valore comune pari ad 1 rappresentano elementi che sono, frequentemente acquistati insieme. Più in generale, l'obiettivo di base dell'association rule è individuare una raccolta di prototipi $(v_1, v_2 \dots v_L)$ per il feature-vector, X , per cui la densità di probabilità $Pr(v_L)$ valutata per ciascuno di questi valori sia relativamente alta. In questo quadro

generale, il problema può essere visto come “mode finding” o “bump hunting”⁷ e si articola o in un’associazione per cui un elemento x_i implica un altro elemento x_j , - per usare l’esempio del paniere di mercato chi compra il pane, in genere compra anche il latte - o in una transazione tra elementi collegati - ad esempio, la lista della spesa.

Concetti base delle regole associative, siano esse per associazione o transazione sono

- Supporto di una regola: la regola $X \Rightarrow Y$ ha supporto s se una frazione pari a s delle transazioni contengono tutti gli item in $X \cup Y$, ad esempio il 40% delle transazioni per famiglie include pane e latte
- Confidenza di una regola: la regola $X \Rightarrow Y$ ha confidenza c se una frazione pari a c delle transazioni in cui compare X contiene Y ad esempio l'80% delle persone che compra il pane compra anche il latte
- Problema di lift: comprende tutte le regole associative che abbiano un supporto minimo rispetto ad una confidenza minima descrivendo la forza della regola. [\(4\)](#)

Nell’apprendimento non supervisionato le associations rules si esplicano nel clustering, per le variabili continue e nella riduzione per le variabili discrete.

Il clustering è una tecnica di intelligenza artificiale che ha come scopo quello di selezionare e raggruppare un insieme di elementi omogenei in “clusters”. Queste tecniche si basano sul concetto di distanza tra due elementi e possono essere di due diversi tipi:

⁷ Il mode finding è un iterativo basato sulla ricerca binaria finalizzato a trovare la modalità di un array ordinato e la sua frequenza [\(106\)](#) Bump hunting è un algoritmo computazionale per il rilevamento di massimi o minimi locali sulle funzioni target. [\(105\)](#)

- Bottom-Up: i dati sono inizialmente separati tra di loro e l'algoritmo provvede a unire gli elementi che reputa "meno distanti", ovvero più simili, in base a determinate caratteristiche.

L'algoritmo apriori ne è un esempio. È utilizzato per la generazione di itemset frequenti per approssimazioni successive, a partire da itemset con un solo elemento. In sintesi, il presupposto teorico è il principio di anti-

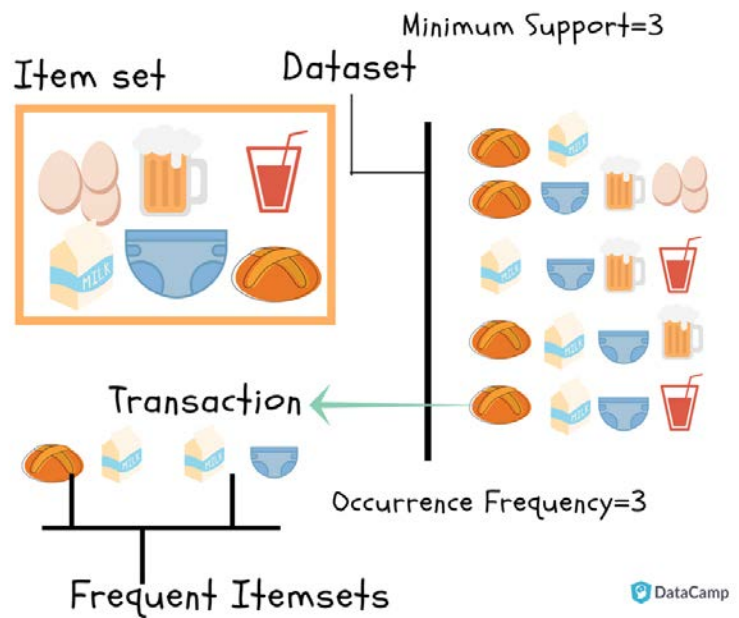


Figura 17: Algoritmo apriori

monotonicità per cui se un insieme di oggetti $\{A,B\}$ è frequente, allora anche i suoi sottoinsiemi $\{A\}$, $\{B\}$ sono frequenti, con un supporto maggiore, ma se un item $\{A\}$ non è frequente, allora neanche gli insiemi che lo contengono $\{A,B\}$ sono frequenti. I sottoinsiemi sono costruiti per *generazione di candidati* aggiungendo un elemento per volta. Ogni candidato deve soddisfare un vincolo di supporto minimo, su cui si generano le regole associative. Il primo passaggio sui dati calcola il supporto di tutti i set di elementi singoli. Quelli il cui supporto è inferiore alla soglia vengono scartati. Il secondo passo calcola il supporto di tutti gli insiemi di oggetti di dimensioni due che possono essere formati dalle coppie dei singoli oggetti sopravvissuti al primo passaggio. In altre parole, per generare tutti gli articoli frequenti con $|K| = m$, dobbiamo considerare solo i candidati tali per cui tutti gli insiemi di dimensioni $m-1$ sono frequenti. Ogni passaggio successivo sui dati associa solo quegli elementi

d'insiemi sopravvissuti al precedente passaggio. Se i dati sono sufficientemente scarsi o se la soglia t è abbastanza alta, il processo termina in un tempo ragionevole anche per set di dati molto grandi.

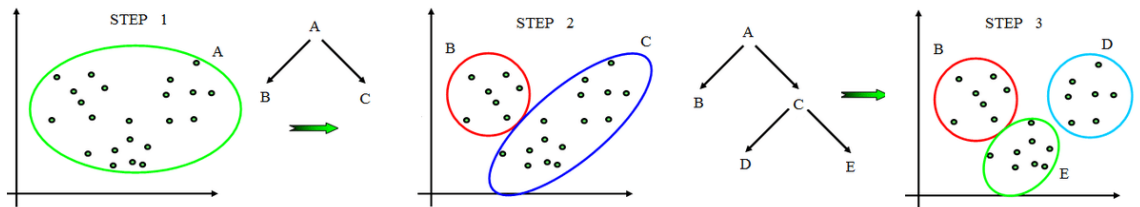


Figura 18: Clustering gerarchico

- Top-Down: inizialmente i dati sono tutti raggruppati in un unico cluster. L'algoritmo divide poi gli elementi in sotto-clusters. Ne sono esempi le strategie per il clustering gerarchico tipicamente di due tipi:
 - Agglomerativo: in cui si parte dall'inserimento di ciascun elemento in un cluster differente e si procede, quindi all'accorpamento graduale di cluster a due a due.
 - Divisivo: in cui tutti gli elementi si trovano inizialmente in un singolo cluster che viene via via suddiviso ricorsivamente in sotto-cluster.

La scelta di una metrica appropriata influenza la forma degli insiemi, poiché alcuni elementi possono risultare più "vicini" utilizzando una misurazione e più "lontani" utilizzandone un'altra. I concetti di *criterio* e *algoritmo* di clustering, tra l'altro sono ben distinti: i criteri descrivono cosa si vuol ottenere specificando il grado di ottimalità di ogni soluzione ammissibile; gli algoritmi, invece, dato un criterio di raggruppamento forniscono una procedura per determinare soluzioni che lo ottimizzano. [\[16\]](#)

La maggior parte dei criteri di clustering sono definiti sulla base delle due osservazioni seguenti:

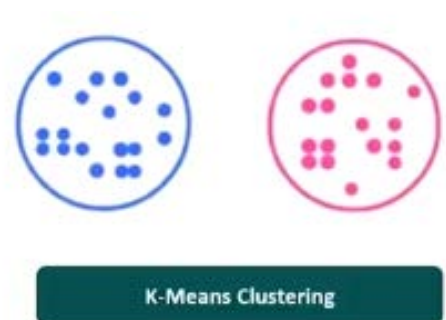
- 1) i patterns all'interno dello stesso cluster devono essere tra loro più simili rispetto ai patterns appartenenti a cluster diversi
- 2) i clusters sono costituiti da nuvole di punti a densità relativamente elevata separate da zone dove la densità è più bassa.

Tra i diversi criteri possibili vi sono:

- minimizzazione delle distanze dai centroidi: minimizza la somma dei quadrati delle distanze dei pattern x dai baricentri, o centroidi, delle classi, risultando valido per cluster circolari, ma penalizzando forme allungate o forme ad anelli concentrici;
- minimizzazione delle distanze intra-classe.

Le principali famiglie di algoritmi comprendono, oltre al clustering gerarchico che aggrega i dati in strutture ad albero dette dendogramma, anche

- Clustering basato su centroidi:
attraverso iterazioni si individuano i cluster cercando di minimizzare la distanza dei patterns dai loro baricentri.



Ne è esempio l'algoritmo il K-means che

risulta computazionalmente molto semplice da implementare in quanto minimizza implicitamente le distanze dai centroidi.

Per partire occorre inizializzare il k-means scegliendo l'ampiezza del set di dati e k centroidi iniziali disposti casualmente che rappresentano i cluster da cui il data set sarà composto. Viene, quindi calcolata la distanza euclidea tra ogni data points e ogni centroide individuato. Ogni data points sarà poi assegnato al baricentro la cui distanza risulti minima. Si procede in questo modo per tutti i cluster fino a che non è soddisfatta la condizione di stop per cui nessun data points cambia cluster, la somma delle distanze è ridotta al minimo ed è stato raggiunto un numero massimo di iterazioni. [\(17\)](#)

Una specializzazione del k-means è l'algoritmo c-means o anche *Fuzzy clustering* in cui ogni point può appartenere a più di un cluster.

L'analisi dell'insieme prevede che gli elementi

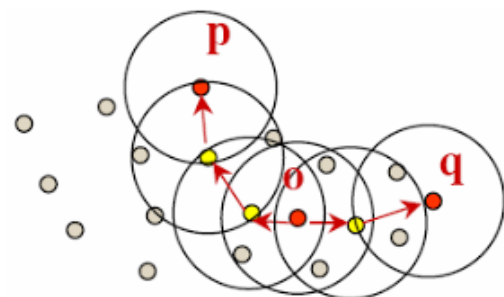
nello stesso cluster siano il più simili possibile, mentre gli elementi appartenenti a cluster diversi siano il più diversi possibile. I cluster sono, quindi identificati tramite misure di somiglianza che includono la connettività e l'intensità e stabiliti in base alle applicazioni. [\(18\)](#)



- Clustering basato sulla densità: i cluster individuati sono regioni connesse in aree ad elevata densità come il Density-Based

Spatial Clustering of Applications with Noise o DBSCAN. Un punto q è density-reachable da un punto p se esiste una sequenza di points p_1, p_2, \dots, p_n con $p_1=p$

e $p_n=q$ dove ogni p_{i+1} è density-reachable direttamente da p_i . Considerati il numero minimo di punti richiesti per formare il cluster si seleziona un punto casuale che non è stato ancora visitato e si determina il suo eps-vicinato, inteso come i numeri di



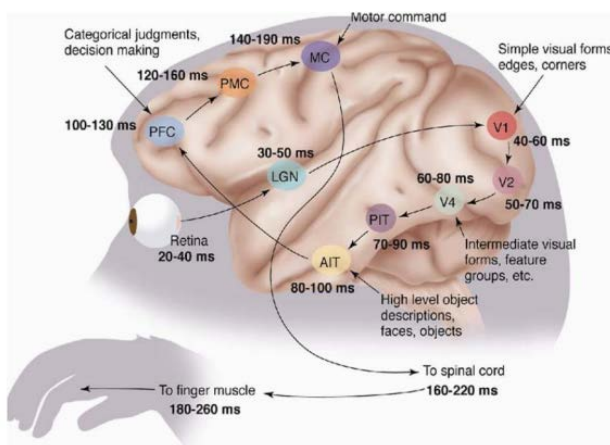
punti distanti al massimo un certo *eps* opportunamente definito; se tale insieme contiene un numero sufficiente di punti viene creato un nuovo cluster, altrimenti il punto viene etichettato come rumore, ma non eliminato perché potrebbe, comunque essere ritrovato in un *eps*-vicinato più grande riconducibile ad un differente point. Il processo continua fino a quando tutti i punti del cluster sono stati visitati. [_\(19\)](#)

2.2 Tecniche di Deep Learning

Il concetto di Deep Learning (DL) è apparso per la prima volta nel 2006 come nuovo campo di ricerca nell'ambito dell'apprendimento automatico, ma coinvolgendo molti altri settori relativi al riconoscimento di modelli.

L'apprendimento profondo considera principalmente due fattori chiave:

- elaborazione non lineare in più livelli: si riferisce a un algoritmo in cui il livello corrente prende l'output del precedente strato come input. La gerarchia consente di condividere e riusare informazioni e, stabilita in modo da definire l'importanza dei dati permette di selezionare features specifiche e scartare dettagli inutili;
- apprendimento supervisionato e non supervisionato in relazione ai dati a disposizione e all'etichetta del target di classe.



La profondità e, quindi il numero di livelli della rete è solo uno dei fattori di complessità caratterizzato altresì, dal numero di neuroni, dalle connessioni e dai loro pesi. In generale maggiore è il numero di pesi, ovvero di parametri da apprendere, maggiore è la complessità del training. Al tempo stesso un

Figura 19: Il sistema visivo umano opera su una gerarchia di dieci livelli tra la retina e i muscoli attuatori che gestiscono la reazione agli stimoli

elevato numero di neuroni e connessioni rende più costosa la propagazione, sia essa back o forward, dell'informazione, perché aumenta il numero di operazioni necessarie per ottenerla.

Le reti neurali artificiali possono essere organizzate in varie architetture quali

- Modelli feedforward: discriminativi per la classificazione o regressione con training prevalentemente supervisionato;
- Modelli generativi: addestrati a ricostruire l'input, utili per pre-training di altri modelli e per produrre features salienti che sfruttano training non supervisionato
- Modelli ricorrenti: utilizzati per natural language processing.

Caratteristica comune di tutte le strutture definite è che la complessità dell'informazione iterata tra uno strato e l'altro viene aumentata in modo che uno strato sottostante sia meglio addestrato rispetto al superiore.

2.2.1 Convolutional Neural Network

Le origini delle reti CNN risalgono agli anni '70, ma il documento fondamentale che ne stabilisce il tema moderno risale al 1998 definendo il significato di "apprendimento basato sul gradiente applicato al riconoscimento dei documenti". Per quanto l'ispirazione neurale in questi modelli è più tenue rispetto alle reti neurali, molte procedure come back-propagation, discesa del gradiente, funzioni di attivazione non lineari e così via sono in comune, per questo si parla di "rete neurale convoluzionale".

Principalmente sono utilizzate per l'immagine processing e, in generale il livello di input è direttamente collegato ai pixel dell'immagine, mentre gli ultimi livelli sono generalmente fully-connected⁸ e

⁸ Una rete neurale completamente connessa consiste in una serie di strati completamente collegati. Uno strato completamente collegato è una funzione da \mathbb{R}_m a \mathbb{R}_n . in cui ogni dimensione di output dipende da ciascuna dimensione di input. Principale

operano come un classificatore MLP. Nei livelli intermedi si utilizzano connessioni locali e pesi condivisi.

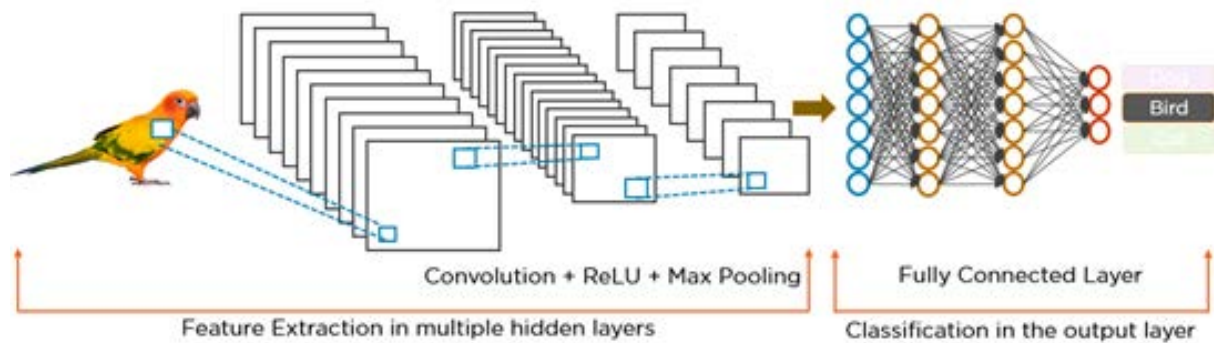


Figura 20: Processo di riconoscimento di un'immagine mediante Rete Neurale Convolutionale

I concetti fondamentali su cui si basano le CNN sono:

- campi ricettivi locali: rappresentano una piccola finestra sui pixel di input con cui ogni neurone nel primo strato nascosto è collegato ad una piccola regione dei

vantaggio delle reti completamente connesse è che sono *struttura agnostici* ossia, non necessitano la formulazione di ipotesi particolari sull'input. (107)

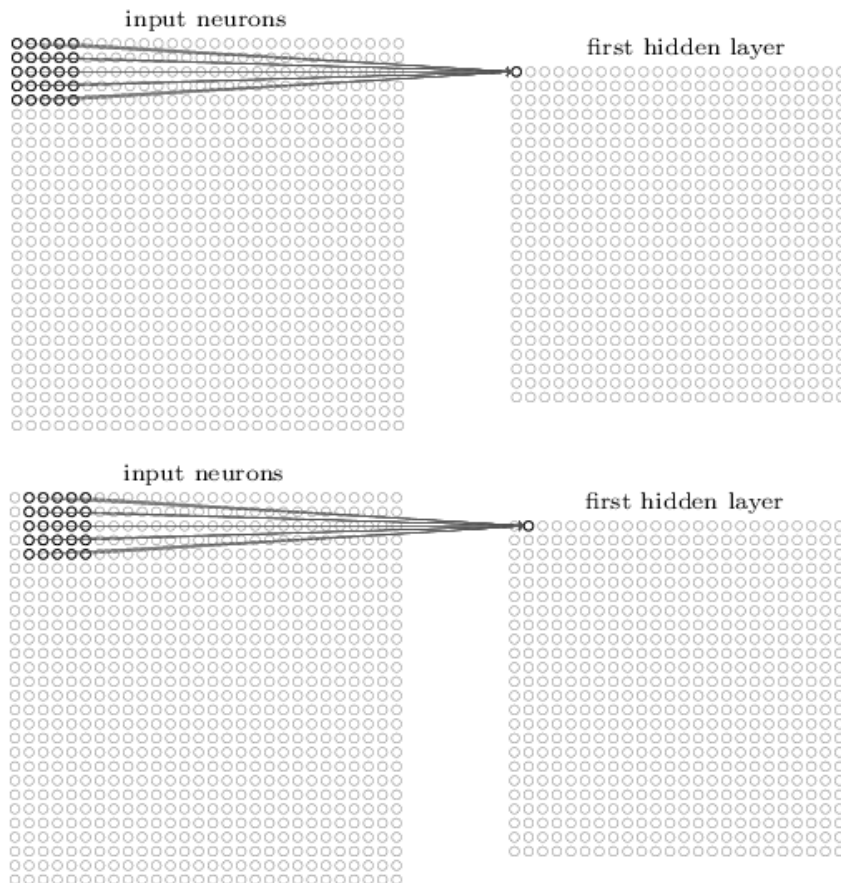


Figura 21: i pixel di input sono collegati ad uno strato di neuroni nascosti realizzando connessioni di piccole aree localizzate dell'immagine di input. Per cui, ogni neurone nel primo strato nascosto sarà collegato a una piccola regione dei neuroni di input, ad esempio una regione 5×5 , corrispondente a 25 pixel di input.

Per l'elaborazione dell'immagine si fa scorrere il campo ricettivo locale sull'intera immagine di input. Per ogni campo ricettivo locale esisterà un diverso neurone nascosto nel primo strato nascosto. Nella figura partendo da sinistra si fa scorrere il campo ricettivo locale di un pixel a destra, cioè di un neurone, per connetterlo ad un secondo neurone nascosto e così via, fino alla costruzione del primo livello nascosto. Si osservi che per un'immagine in ingresso 28×28 con campi ricettivi locali 5×5 , esisteranno 24×24 neuroni nel livello nascosto. Questo perché possiamo solo spostare il campo ricettivo locale di 23 neuroni, a destra o verso il basso prima di scontrarsi con il bordo dell'immagine.

neuroni di input con un peso e un bias associati per l'apprendimento. Ogni neurone nascosto impara ed analizza il suo particolare campo ricettivo locale;

- pesi condivisi: gli stessi pesi e gli stessi bias sono utilizzati per tutti i neuroni nascosti.

Ciò significa che tutti i neuroni nel primo livello hidden rilevano esattamente la stessa feature⁹ in posizioni diverse nell'immagine di input. Per capire perché ciò abbia senso,

⁹ Informalmente, si può pensare ad una feature rilevata da un neurone nascosto come al tipo di schema di input che causerà l'attivazione del neurone: potrebbe essere un bordo nell'immagine, ad esempio, o forse un altro tipo di forma.

si supponga che pesi e distorsioni siano tali che il neurone nascosto possa individuare, per esempio, un bordo verticale in un particolare campo ricettivo locale. È probabile che tale abilità sia utile anche in altri punti dell'immagine. Risulta, di conseguenza interessante applicare lo stesso rilevatore di funzionalità ovunque, Questa particolare specifica rende le reti convoluzionali adatte all'invarianza della traduzione delle immagini;

- **pooling layers:** oltre ai livelli convoluzionali, le CNN contengono anche livelli di raggruppamento, generalmente utilizzati immediatamente dopo, che semplificano le informazioni nell'output ricevuto dal livello convoluzionale superiore. Un livello di pool prende ciascuna feature map, output dallo strato convoluzionale e, attivandosi costruisce una propria feature map condensata. Un modo concreto per la rete di chiedere se una determinata funzionalità si trova ovunque in una regione dell'immagine è il *max pooling* che è un processo di discretizzazione che, riducendo la dimensionalità consente di formulare ipotesi sulle caratteristiche contenute nelle sotto-regioni, in questo caso, dell'immagine. In generale, il max pooling opera subordinando la posizione esatta di una feature a quella approssimativa rispetto ad altre caratteristiche riducendo i parametri necessari ai livelli successivi. [\(20\)](#)

2.2.2 Recurrent Neural Network

Una rete neurale ricorrente (RNN) è una classe di reti neurali artificiali in cui le connessioni tra i nodi formano un grafico diretto lungo una sequenza temporale che le consente di esibire un comportamento dinamico. Derivate da reti neurali feedforward, le RNN possono utilizzare il loro strato interno come una memoria ed elaborare sequenze di input di lunghezza variabile.

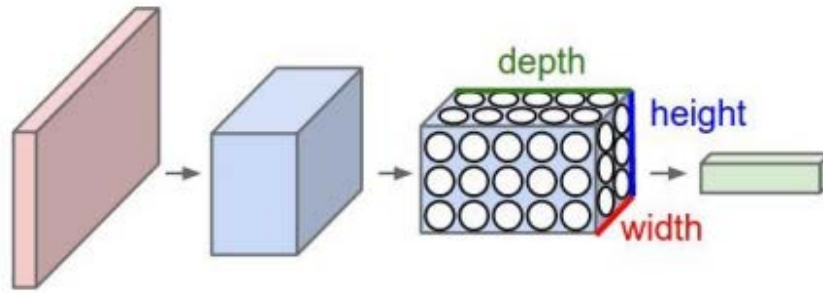


Figura 22: i livelli di ciascun neurone sono organizzati come griglie 3D

Il termine "rete neurale ricorrente" viene usato indiscriminatamente per riferirsi a due ampie classi di reti con una struttura generalmente simile, in cui una reagisce ad un impulso finito e l'altra ad un impulso infinito, ma che presentano strati di memorizzazione aggiuntivi che possono essere controllati direttamente. Nelle RNN, infatti, a differenza che nelle altre reti, tutti gli input sono correlati tra loro. Si supponga, ad esempio di voler prevedere la parola successiva in una determinata frase, in tal caso, la relazione tra tutte le parole precedenti aiuta a prevedere il risultato migliore. La RNN ricorda tutte queste relazioni apprese lungo tutto l'allenamento stesso.

Per ottenere questo risultato, le architetture delle RNN si compongono di loop di feedback che consentono di richiamare “alla memoria” l’informazione.

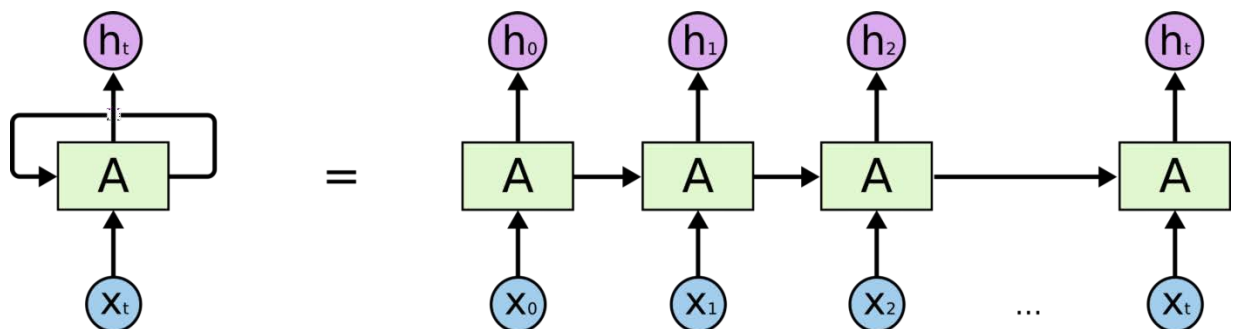


Figura 23: La Rete Neurale Ricorrente fornisce $x(0)$ dalla sequenza di input e quindi emette $h(0)$ che insieme a $x(1)$ è l'input per il passaggio successivo. Quindi, $h(0)$ e $x(1)$ sono l'input per il passaggio successivo. Allo stesso modo, $h(1)$ dal prossimo è l'input insieme a $x(2)$ per il passaggio successivo e così via. In questo modo, continua a ricordare il contesto durante l'allenamento. (21)

L'archiviazione può avvenire in celle o reti più complesse, che ne incorporino ritardi o cicli di feedback.

Tali strati controllati sono definiti *stato o memoria gated* e compongono reti di memoria a lungo termine e unità ricorrenti. Questo processo è anche chiamato Feedback Neural Network.

Nell'apprendimento supervisionato con impostazioni temporali discrete, ai nodi di input le sequenze di vettori di input con valori reali sono inviate un vettore alla volta. In qualsiasi momento, ciascuna unità sottostante calcola la sua attuale attivazione come una funzione non lineare della somma ponderata delle attivazioni di tutte le unità che si collegano ad essa. Le attivazioni target fornite dal supervisore possono essere generate per alcune unità di output in determinate fasi temporali. Ad esempio, se la sequenza di input è un segnale vocale corrispondente a una cifra parlata, l'uscita target alla fine della sequenza sarà un'etichetta che classifica la cifra. Ad ogni fase, l'input viene trasmesso in avanti e viene applicata una regola di apprendimento. Le connessioni posteriori fisse salvano una copia dei valori precedenti nelle unità di contesto. In questo modo la rete può mantenere una sorta di stato, permettendole di svolgere compiti, come ad esempio, la previsione di sequenze, che vanno oltre il potere di un perceptrone multistrato standard. (22)

2.2.3 Reti Profonde

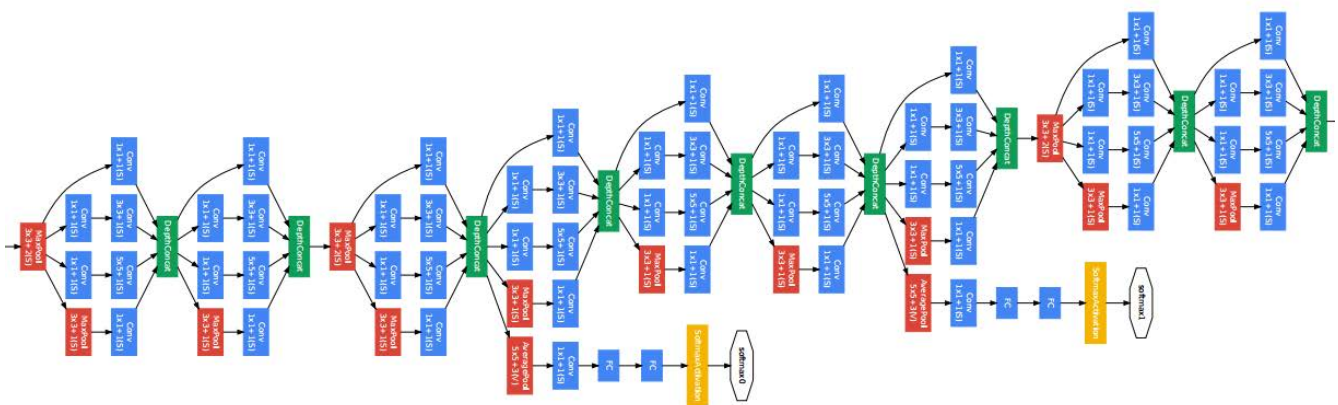


Figura 24. Reti Neurali Residuali

Una rete con n strati dovrebbe essere capace, quantomeno di raggiungere lo stesso grado di accuratezza di una rete a n-1 strati copiando la rete più piccola ed eseguendo un'identity mapping per l'ultimo strato. Perpetrando il modello a ordini superiori si dovrebbe, così ottenere un'accuratezza sempre maggiore al

crescere della profondità. Questo principio, però non è sempre vero, anche a causa della propagazione dell'errore.

L'ipotesi, per correggere questo gap è che, essendo le mappature dirette difficili da allenare, invece di apprendere direttamente da esse si apprenda tramite una funzione $f(x)+x$ generata da un *residuale* definito come

$$f(x)=h(x)-x$$

dove $h(x)$ è sempre una funzione d'ipotesi, sul quale è possibile agire, successivamente, per “aggiustare” le previsioni ottenute.

Una rete di questo genere si definisce, di conseguenza, *residuale* o ResNet e, ogni suo blocco, consiste in una serie di strati e di una identity mapping che associa l'input del blocco al suo output corrispondente. Questa operazione di addizione viene effettuata elemento per elemento in modo da semplificare notevolmente la costruzione di strati di identità. Risulta, infatti, più semplice da una trasformazione d'identità spingere $F(x)$ verso 0 lasciando x come output di riferimento da cui apprendere, piuttosto che partire da zero.

Prima di questa tecnica, nelle reti neurali profonde si riscontrava un problema di annullamento del gradiente, la cui discesa, data dalla minimizzazione della funzione di errore, si riduce esponenzialmente attraverso la retro-propagazione degli strati precedenti. In sostanza, la lunga strada attraverso gli strati precedenti rendeva i segni degli errori talmente piccoli da non permettere alla rete di apprendere. Grazie alla nuova tecnologia delle ResNet risulta possibile costruire reti di innumerevoli strati che presentano anche un elevato grado di accuratezza. [\(23\)](#)

2.2.4 Encoder

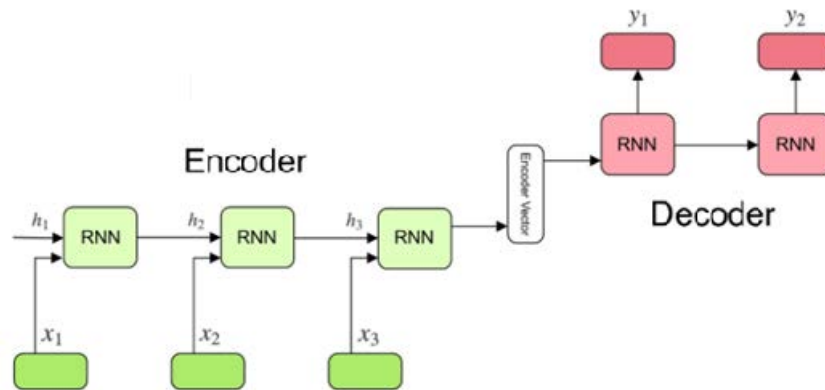


Figura 25: Modello di sequenza Encoder-Decoder

Un encoder è una pila di più unità ricorrenti in cui ciascuna accetta un singolo elemento della sequenza di input, raccoglie informazioni per quell'elemento e le propaga in avanti.

In un problema di risposta ad una domanda, ad esempio la sequenza di input è una raccolta di tutte le parole della domanda. Ogni parola è rappresentata come x_i dove i è l'ordine della parola stessa.

Gli stati nascosti h_i sono rappresentati dal risultato di una normale rete neurale ricorrente in cui

$$h_t = f(w_h h_{t-1} + w_x x_t)$$

è definito *Encoder Vector* e rappresenta lo stato nascosto finale generato dalla parte encoder del modello. Questo vettore mira a incapsulare le informazioni per tutti gli elementi di input al fine di aiutare il decodificatore a fare previsioni accurate fungendone da input.

Il decoder invece è una pila di più unità ricorrenti in cui ciascuna prevede un output y_t nella fase temporale t .

Ogni unità ricorrente accetta uno stato nascosto dall'unità precedente e produce ed emette così il proprio stato nascosto definito [\(24\)](#)

$$h_i = f(w_h h_{i-1})$$

In sostanza gli autoencoder sono reti neurali con lo scopo di generare nuovi dati dapprima comprimendo l'input in uno spazio di variabili latenti e, successivamente, ricostruendo l'output sulla base delle informazioni acquisite. Questa tipologia di network è composta dalle due parti prima definite dell'Encoder che comprime l'input in uno spazio di variabili latenti, rappresentato dalla funzione di codifica $h=f(x)$ e del Decoder che si occupa di ricostruire l'input sulla base delle informazioni precedentemente raccolte ed è rappresentato dalla funzione di decodifica $r=g(h)$.

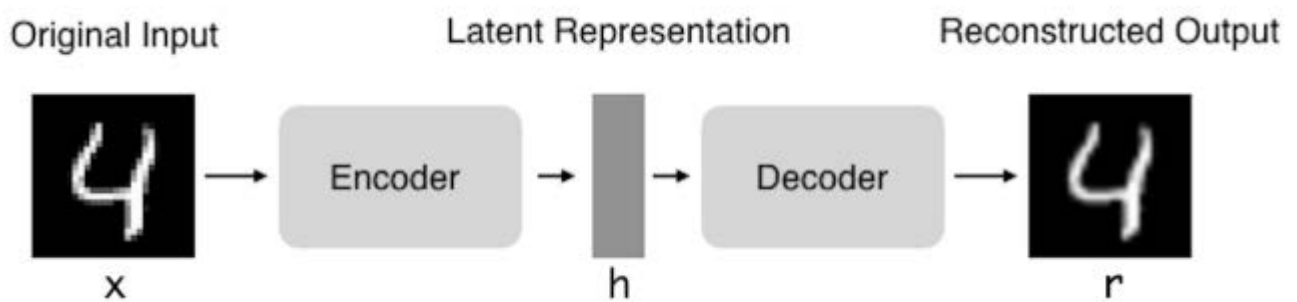


Figura 26: Architettura di un Autoencoder

L'autoencoder nel suo complesso può, quindi essere descritto dalla funzione

$$d(f(x)) = r$$

dove r è quanto più simile all'input originale x .

Allenando l'autoencoder a copiare l'input, lo spazio di variabili latenti h può assumere delle caratteristiche utili imponendo dei limiti all'azione di codifica e costringendo lo spazio h a dimensioni minori di quelle di x . In questo caso l'autoencoder viene chiamato undercomplete. Senza vincoli troppo stringenti, la rete si limita al compito di copiare l'input in output, senza estrapolare alcuna informazione utile sulla distribuzione dei dati. Ciò può accadere anche quando la dimensione del sottospazio latente ha la stessa grandezza dello spazio di partenza o, nel caso di autoencoder overcomplete, quando le dimensioni dello spazio di variabili latenti è maggiore degli input.

La riduzione del rumore e la riduzione della dimensionalità per la visualizzazione dei dati sono considerate le applicazioni più interessanti degli autoencoder.

Queste reti neurali sono allenate a preservare quante più informazioni possibili e l'allenamento avviene automaticamente attraverso dati di esempio, in modo che risulti, particolarmente performante su tipologie di input simili, anche senza necessità di generalizzare, ma riuscendo, comunque a far sì che le nuove rappresentazioni acquisiscano differenti tipi di proprietà.

La riduzione del rumore e la riduzione della dimensionalità per la visualizzazione dei dati sono, attualmente le applicazioni più interessanti degli autoencoder. [\(25\)](#)

3 Approcci per la valutazione quantitativa dei modelli

La evaluation di un modello predittivo, come già accennato rappresenta un passaggio molto importante dello sviluppo del modello stesso. Alcuni metodi come le reti ANN eseguono la valutazione in fase di back-propagation, ma in generale essa viene effettuata con tecniche dirette utilizzando una varietà di metriche differenti.

Ogni modello di apprendimento automatico risolve un medesimo problema ponendosi un obiettivo diverso e utilizzando set di dati diversi, di conseguenza è importante comprendere il contesto prima di scegliere una metrica.

Se da un lato gli algoritmi di apprendimento automatico hanno le potenzialità per migliorare i processi, affinare le previsioni dei risultati e contenere i costi, dall'altro presentano potenziali criticità da tenere in considerazione, quali

- **Overfitting:** si verifica quando un modello basa le sue previsioni su correlazioni spurie all'interno di un campione di dati, anziché su relazioni autentiche esistenti all'interno della popolazione nel suo complesso. Gli alberi di classificazione e di regressione sono particolarmente sensibili a questa tipologia di criticità. Se un modello caratterizzato da overfitting viene applicato a nuovi dati appartenenti alla stessa popolazione, può, potenzialmente, produrre previsioni poco accurate.
- **Underfitting e Riduzione della trasparenza del modello:** l'underfitting si verifica quando il modello ha prestazioni scarse sui dati di addestramento perché non è in grado di ridurre

l'errore tra gli esempi di input e i valori di uscita; in sostanza il dimensionamento è stato strutturato in modo che sia nella fase di test, che nella fase di training, l'algoritmo non è in grado di identificare errori e quindi li propaga. Alcuni algoritmi presentano grafici abbastanza comprensibili, ma altri, quali le Random Forest e le Reti Neurali, funzionano come delle "scatole nere" diminuendo la trasparenza e l'interpretabilità della metodologia sottostante. Di conseguenza, per quanto i dati e le specificità definite siano corrette e pur essendo possibile esaminare i vari step intermedi - prodotti dall'algoritmo per capire come genera le previsioni, - spesso è necessario ed opportuno possedere una conoscenza molto approfondita e consolidata della tecniche e del problema per poter valutare con trasparenza e giudizio critico l'intero processo. Se, in pratica non si è in grado di capire in che modo il modello genera i risultati, si potrebbe ignorare la presenza di distorsioni latenti ed errori, che sbagliano la predizione. Nelle tecniche di regressione tradizionali, ad esempio, quando la maggioranza dei coefficienti delle variabili risulta non significativa, la bontà di adattamento del modello è inaccurata; o, anche, quando le variabili non hanno il potere esplicativo previsto, il modello potrebbe contenere un errore o un settaggio non ottimale. Analogamente se i dati sono utilizzati in un contesto sbagliato il risultato potrebbe essere completamente diverso da quello atteso.

- Risultati che si basano su dati campionari non adeguati: la qualità dei risultati ottenuti da un modello è direttamente proporzionale a quella dei dati su cui viene costruito ed è facile che possano sfuggire potenziali criticità presenti nel processo di modelling, ad esempio dovute a dati campionari non adeguati. Tecniche di campionamento¹⁰ poco idonee al tipo di proiezione che si intende produrre, possono fornire dati non rappresentativi della popolazione nella sua interezza. [\(26\)](#)

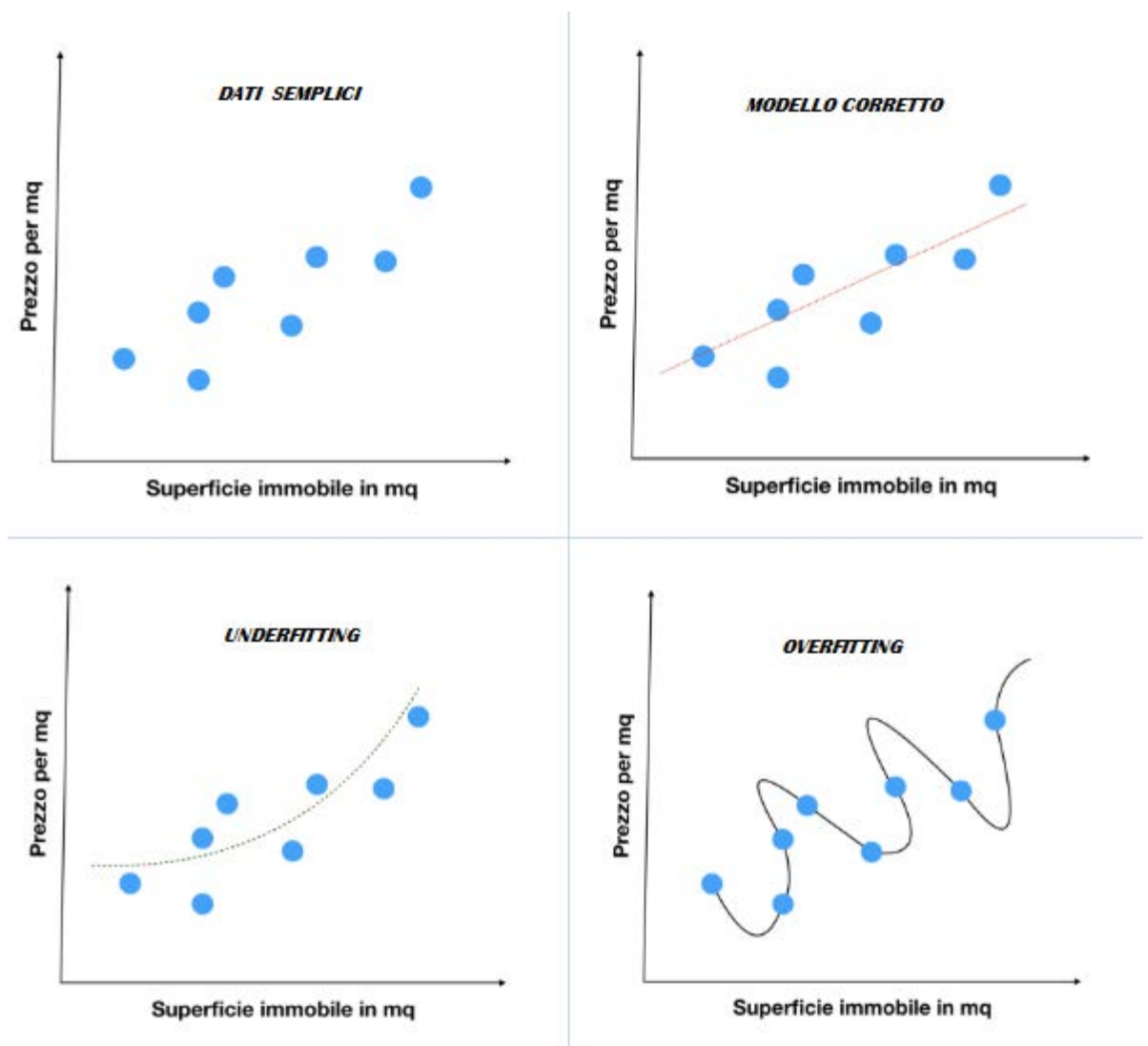


Figura 27 Modello regressivo applicato per la previsione dell'andamento sul mercato del prezzo delle abitazioni in funzione della metratura. La retta (tracciata in rosso) è probabilmente il modello in grado di predire meglio i dati non noti conoscendo che normalmente il prezzo al mq di un immobile, tende a non crescere proporzionalmente all'aumentare della superficie anziché tendere a ridursi come per la curva del secondo ordine (in verde nel terzo quadrante) che potrebbe introdurre all'aumentare della superficie un errore sul prezzo in rialzo. Diventa difficile, invece dare un significato ai risultati oscillanti previsti dalla curva più complessa nel quarto quadrante. In definitiva il modello più semplice potrebbe di fatto essere quello più adeguato. [\(27\)](#)

¹⁰ Intese come “analisi su base campionaria”

La maggior parte delle funzionalità applicate deve, di conseguenza essere identificata da un esperto e quindi codificata in base al dominio e al tipo di dati a disposizione.

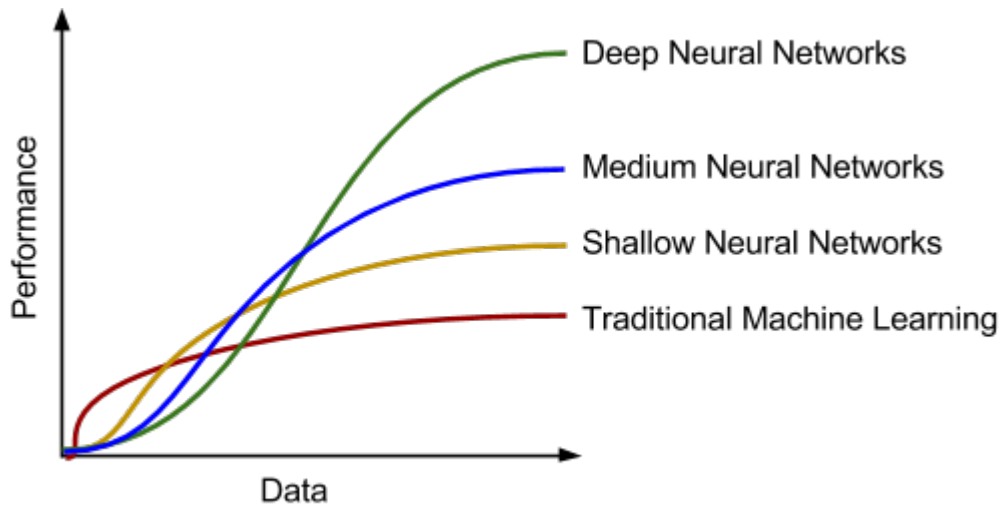


Figura 28: La superiorità delle tecniche di deep learning rispetto ad altri approcci si manifesta quando sono disponibili grandi quantità di dati di training. Mentre le altre tecniche infatti si assestano su una certa performance all'aumentare dei dati di training, il deep learning migliora continuamente.

Nel caso del deep learning, l'algoritmo rileva autonomamente le funzionalità di alto livello dai dati stessi, non è, pertanto necessario selezionarle manualmente perché, appunto apprese direttamente durante l'allenamento. Per questo motivo il deep learning richiede una quantità di dati di gran lunga maggiore rispetto ad algoritmi non profondi che mantengono accuratezza elevata anche con quantità inferiori di dati, ma che non sono in grado di fare estrapolazioni autonomamente. Per lo stesso motivo all'aumentare della quantità dei dati le prestazioni dei modelli di deep learning migliorano, mentre quelle degli algoritmi classici tendono ad assestarsi su di un valore di accuratezza pressoché costante anche in seguito all'aggiunta di nuovi modelli.

La Cross-validation permette di verificare quale modello funziona meglio, tra i vari algoritmi testati, utilizzando un set di convalida incrociata e di ottimizzare gli iper-parametri del classificatore per scegliere, ad esempio, il numero di unità nascoste nella rete neurale. [\(28\)](#)

In Amazon ML, appunto si utilizza il metodo di convalida incrociata k-fold per eseguire la cross-validation. I dati di input vengono suddivisi in k sottoinsiemi di dati, noti anche come folds e si allena il

modello ML su tutti tranne uno, il (k-1)-esimo, dei sottoinsiemi; quindi si effettua la cross-validation sul sottoinsieme che non è stato utilizzato per l'allenamento. Questo processo viene ripetuto k volte, con un sottoinsieme diverso escluso dalla formazione precedente. [\(29\)](#)



Figura 29; Esempio dei sottoinsiemi di addestramento e dei sottoinsiemi di valutazione complementari generati per ciascuno dei quattro modelli creati e formati durante una convalida incrociata quadrupla. Il modello uno utilizza il primo 25 per cento dei dati per la valutazione e il restante 75 per cento per la formazione. Il modello due utilizza il secondo sottoinsieme del 25 per cento (dal 25 per cento al 50 per cento) per la valutazione, e i restanti tre sottogruppi di dati per la formazione e così via. [\(29\)](#)

In caso di apprendimento profondo il modello di suddivisione del set di dati dipende totalmente dalla quantità disponibile. La dimensione del set di training sarà molto più elevata rispetto alla validazione incrociata e al set di test. Per spiegarne il funzionamento con un esempio si consideri un set di dati con 1.000.000 points, il modello di suddivisione può essere 98/1/1, ovvero il set di addestramento conterrà 980.000 punti dati, il set di convalida incrociata conterrà 10.000 punti dati e il set di test conterrà 10.000 punti dati. In generale la convalida incrociata e il set di test non richiedono una grande quantità di dati e le dimensioni dei set non devono necessariamente coincidere. [\(28\)](#)

3.1 Introduzione delle metriche



Figura 30 L'intuizione medica può essere comparata a come un dottore usa la propria rete neurale integrata: dopo aver analizzato un numero sufficiente di casi, un medico è in grado, consapevolmente o meno, di identificare alcuni fattori aggiuntivi che lo aiutano a restringere lo spazio della soluzione le raggiungere un'ipotesi che considera una possibile diagnosi. Rappresenta, quindi un processo automatizzato che può essere emulato non meno bene da una macchina. (30)

Nella pratica medica una capacità essenziale è l'intuizione che permette al medico di capire velocemente la possibile malattia di un paziente, con poche domande e la semplice osservazione. La definizione fornita dal dizionario italiano Treccani di *intuizione* è di una “ *conoscenza diretta e immediata di una verità, che si manifesta allo spirito senza bisogno di ricorrere al ragionamento, considerata talora come forma privilegiata di conoscenza che consente, superando gli schemi dell'intelletto, una più vera e profonda comprensione (e, a volte, creazione) dell'oggetto*”; in realtà, però, il processo di conoscenza e acquisizione che porta all'affermazione che una *percezione passiva* sia una reale possibile diagnosi da parte del medico è rappresentabile come un processo automatizzato che può essere eseguito non meno bene da una macchina. Si basa, infatti su informazioni ed esperienze immagazzinate negli anni di studio e pratica che permettono al medico esperto di asserire che un'anamnesi faccia capo ad una malattia piuttosto che ad un'altra. (30)

Lì dove un medico non ha esperienza e conoscenza bastevole a convalidare il proprio intuito una macchina può acquisirle molto velocemente sfruttando anche poche informazioni a disposizione. La modalità con cui una macchina afferma la propria sicurezza è rappresentata dalle metriche di valutazione, specifiche per ogni paradigma in base al set di dati utilizzato per il training e che descrivono la precisione con cui la predizione è avvenuta permettendo al modello scelto di essere utilizzato per studi analoghi.

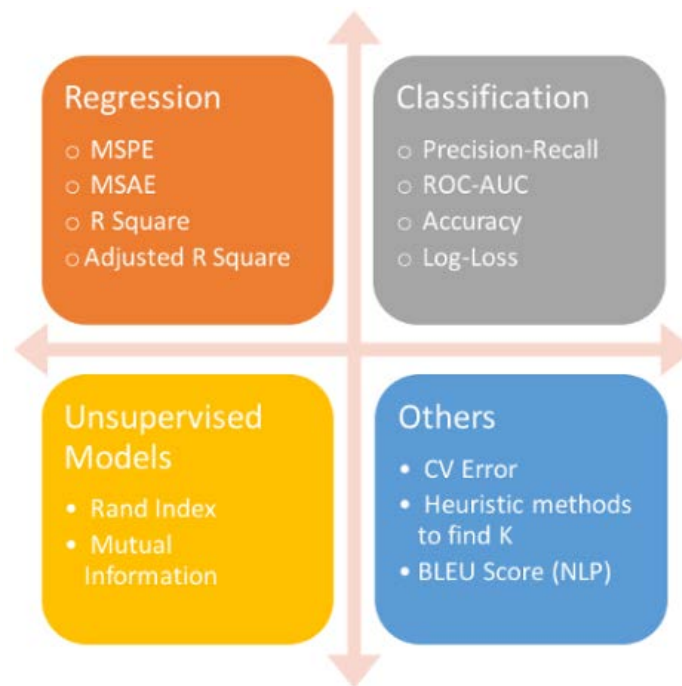


Figura 31; Metriche di valutazione

Si consideri, ad esempio la valutazione di un ecocardiogramma mediante una rete neurale istruita con un database composto da oltre 4000 istanze di segnale ECG estratte da pazienti ambulatoriali e distinto in tre classi:

- a. normale;
- b. battito prematuro atriale;
- c. contrazione ventricolare prematura

Una prima metrica di valutazione per la predizione di malattia o salute per l'ECG di un paziente esterno al database è l'*accuratezza* che indica la percentuale di istanze, sia positive che negative classificate correttamente. Più il valore di accuratezza si avvicina a 1 più il classificatore predice con precisione.

Nell'esempio utilizzato il valore di accuratezza indica che il modello predittivo scelto ha eseguito una corretta classificazione tra le due classi correlate alle malattie cardiache - battito prematuro atriale e contrazione ventricolare prematura - e quella relativa allo stato di buona salute. Sulla base dei risultati ottenuti dalla *matrice di confusione* in cui ogni colonna rappresenta i valori predetti, mentre ogni riga rappresenta i valori reali è possibile valutare il metodo proposto, applicando le funzioni di classificazione statistica. In pratica attraverso la matrice è osservabile la "confusione" nella classificazione delle tre diverse classi, in modo che un segnale normale sia effettivamente riconosciuto come tale restituendo, quindi, una rappresentazione dell'accuratezza. I parametri di classificazione statistica rappresentano i rapporti tra i valori corretti e non corretti individuati nella matrice e ne indicano la correttezza di funzionamento in base alla *sensibilità*, nota anche come *vero rapporto positivo -TPR*, la *specificità* o anche *rapporto negativo reale -TNR*, il *Fall - Out*, noto anche come *rapporto di falsi positivi - FPR* e misura dell'accuratezza del test.

$$\text{Precision} = \frac{\text{True Positive}}{\text{Actual Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{Total}}$$

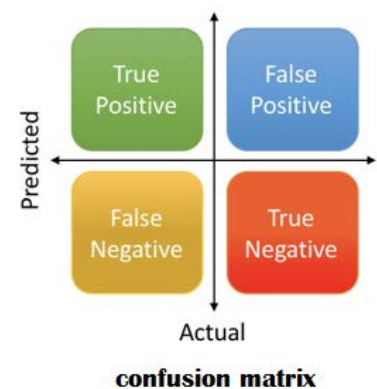


Figura 32: Principali metriche di valutazione

Nell'esempio della valutazione degli ecocardiogrammi la sensibilità rappresenta la percentuale di registrazioni ECG appartenenti a una categoria specifica correttamente classificata, la specificità è la frequenza con cui il classificatore potrebbe classificare un ECG non appartenente alla categoria corretta e il Fall-Out indica che sono state considerate le registrazioni ECG come appartenenti a una categoria specifica, ma, in realtà, di cui non facevano parte. Ulteriore parametro di valutazione è definito dallo F1-

score che tiene in considerazione precisione e recupero di un test, dove la precisione è il numero di veri positivi (TP) diviso il numero di tutti i risultati positivi, mentre il recupero è il numero di veri positivi diviso il numero di tutti i test che sarebbero dovuti risultare positivi, ovvero veri positivi (FP) + falsi negativi (FN). La F1 può assumere valori compresi fra 0 e 1 dove il valore 0 è possibile solo se almeno uno dei due vale 0, mentre assume valore 1 se sia precisione che recupero valgono 1. (31)

Ulteriori parametri di valutazione di un modello predittivo riguardano la velocità intesa non solo come learning curve cioè come rapporto tra tempo necessario per l'apprendimento e quantità di informazioni correttamente apprese, ma anche come tempo di valutazione per costruire il modello e tempo per usarlo in base ai dati in possesso.

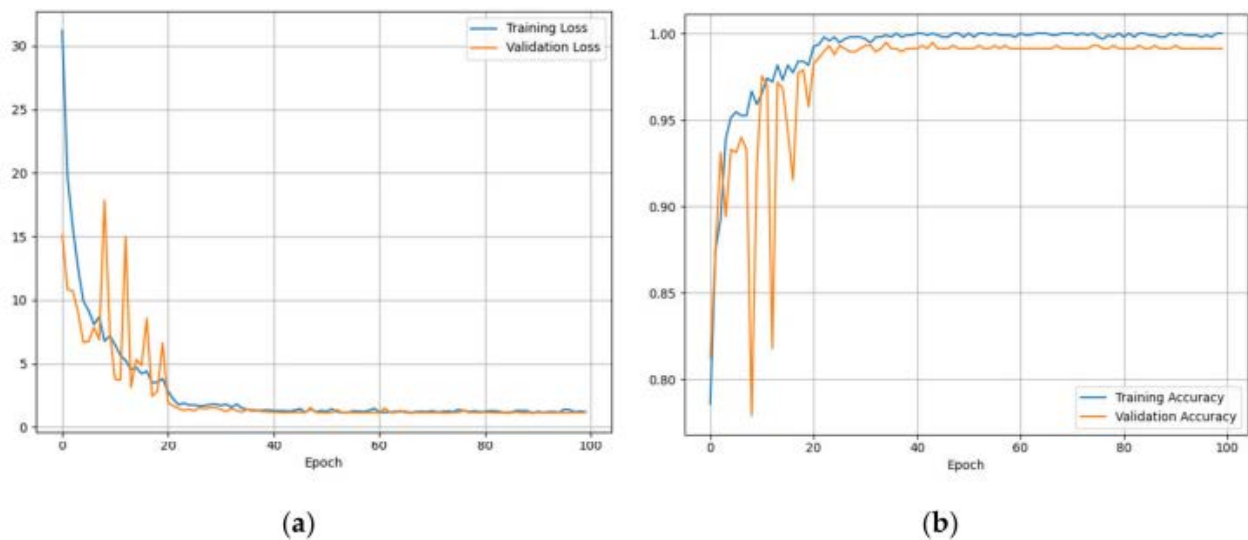


Figura 33 Valutazione di un dataset di 4000 istanze di segnale ECG ottenuti da 47 soggetti ambulatoriali: 25 maschi e 22 femmine valutato mediante reti neurali convoluzionali (CNN) per lo sviluppo di sistemi automatici di diagnosi delle malattie cardiache mediante elettrocardiogrammi. La Figura a indica la perdita di formazione e validazione e quindi la perdita di addestramento mentre la Figura b mostra la precisione di addestramento e validazione. Come mostrano i grafici, dopo 100 epoche, allenamento e le perdite di validazione si sono stabilizzate ad un valore vicino allo zero mentre l'addestramento e l'accuratezza della validazione stabilizzato al 100% (31)

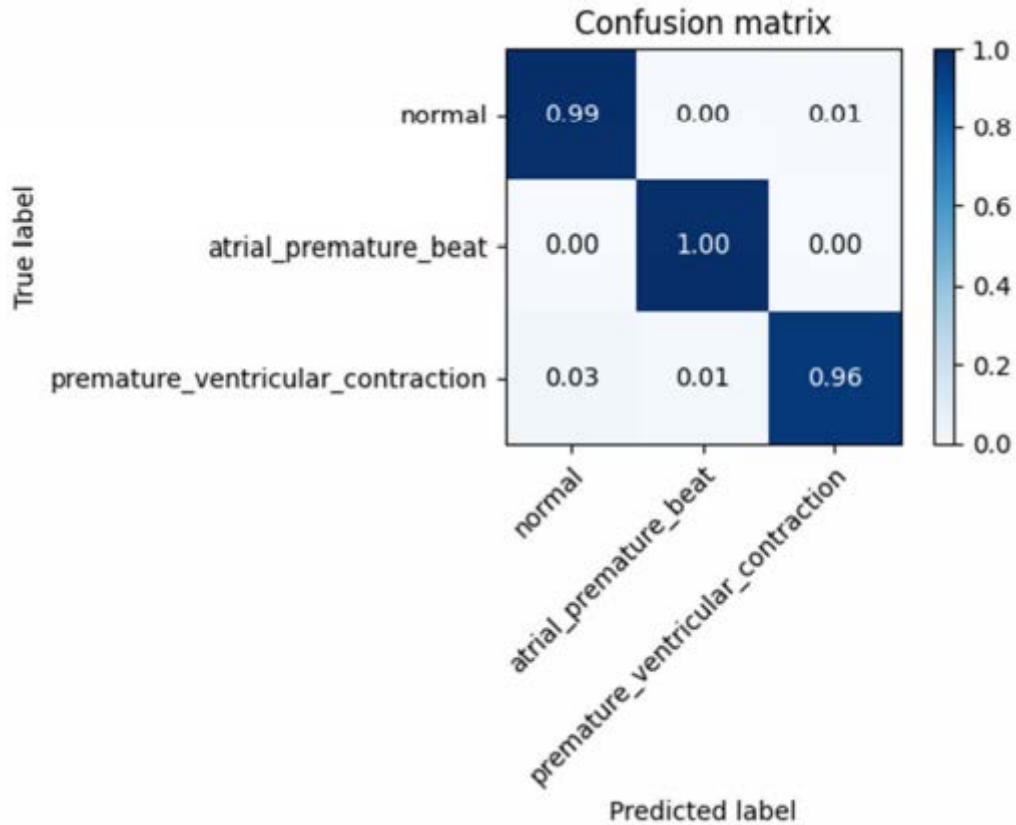


Figura 34 Valutazione di un dataset di 4000 istanze di segnale ECG ottenuti da 47 soggetti ambulatoriali: 25 maschi e 22 femmine valutato mediante reti neurali convoluzionali (CNN) per lo sviluppo di sistemi automatici di diagnosi delle malattie cardiache mediante elettrocardiogrammi. La matrice di confusione evidenzia un livello medio di precisione della classificazione del 98,33%. (31)

α	Class	TPR	TNR	FPR	FDR	F1 Score
1	Normal	99.0%	97.1%	2.9%	1%	98.0%
2	Atrial premature beat	100%	99.0%	1.0%	0%	99.5%
3	Premature ventricular contraction	96.0%	98.96%	1.04%	4%	97.5%
Mean Accuracy		98.33%	98.35%	1.65%	1.66%	98.33%

$$TPR = \frac{TP}{TP + FN} \quad TNR = \frac{TN}{FP + TN}$$

$$FPR = 1 - TNR \quad FDR = \frac{FP}{FP + TP}$$

$$F_1 = \frac{2TP}{2TP + FP + FN}$$

Figura 35 Valutazione di un dataset di 4000 istanze di segnale ECG ottenuti da 47 soggetti ambulatoriali: 25 maschi e 22 femmine valutato mediante reti neurali convoluzionali (CNN) per lo sviluppo di sistemi automatici di diagnosi delle malattie cardiache mediante elettrocardiogrammi. Parametri di classificazione statistica

Esistono altri parametri che permettono di valutare correttamente un risultato e di scegliere adeguatamente un modello come la *robustezza* che rappresenta la sua abilità nel fare previsioni corrette anche in presenza di dati errati o mancanti e la *scalabilità* caratteristica degli algoritmi efficienti non solo per piccoli insiemi di dati, ma anche per grossi database. In seguito, inoltre sono riportati alcuni esempi di ROC curve, Receiver Operating Characteristics parametro fondamentale che mette in relazione la specificity, che come detto è la percentuale di elementi negativi del testing set che sono stati classificati come negativi, con la recall detta anche sensitivity che rappresenta appunto la percentuale di istanze realmente positive classificate come tali in rapporto al variare della soglia. Il valore minimo/massimo della soglia è quello per cui sono tutti falsi positivi o tutti veri positivi.

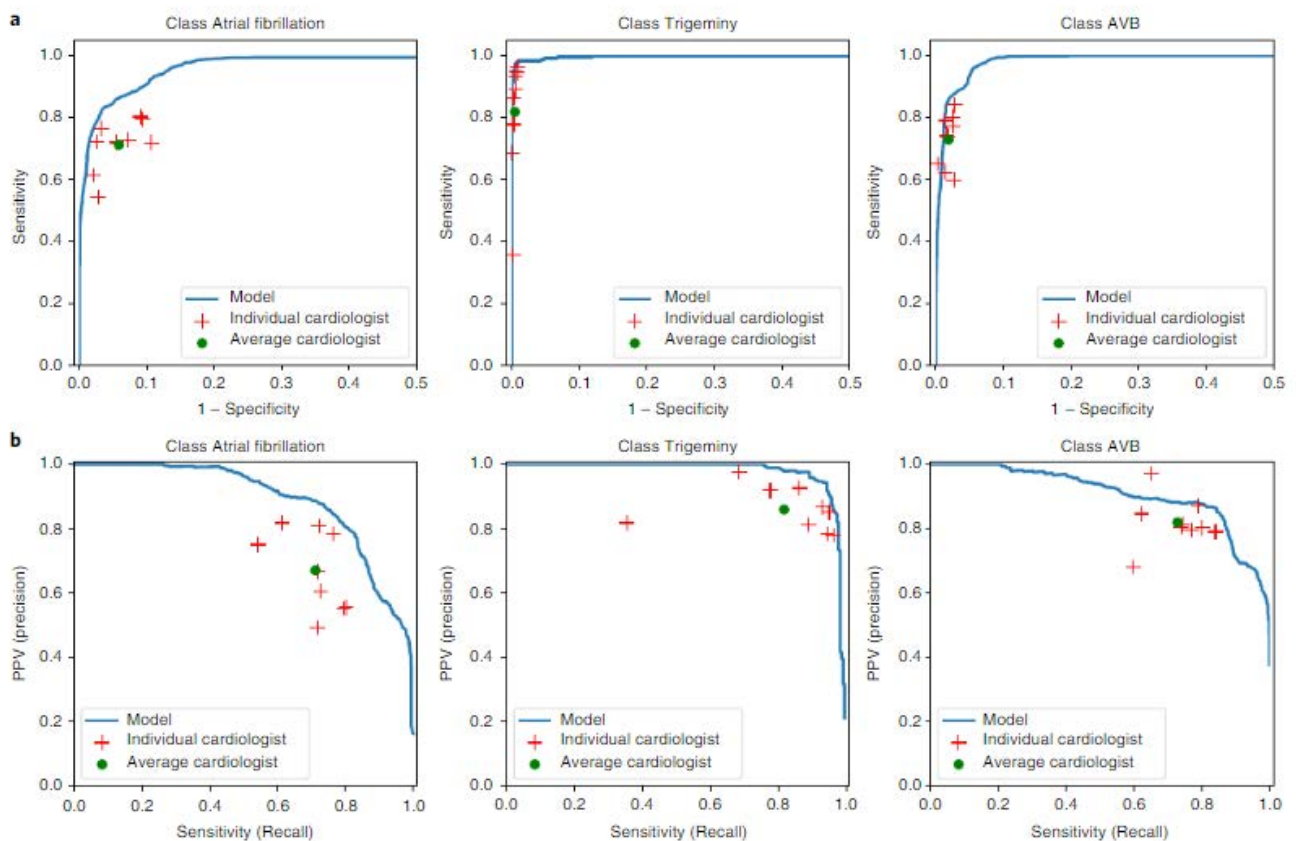


Figura 36 Esempi di curve ROC calcolate a livello di sequenza per fibrillazione atriale (AF), trigeminy e AVB. **b,** Esempi di curve di richiamo di precisione calcolate a livello di sequenza per fibrillazione atriale, trigeminy e AVB. La prestazione individuale del cardiologo è indicata dalle croci rosse e la prestazione media del cardiologo è indicata dal punto verde. La linea rappresenta il ROC (a) o la curva di richiamo di precisione (b) raggiunta dal modello (32)

4 Soluzioni sviluppate nel Progetto AMICO

Il progetto AMICO si propone di realizzare un'infrastruttura, chiamata "ambiente instrumentato", costituita dall'ambiente domestico e dalla persona, entrambi opportunamente muniti di sensori, da una piattaforma di servizi di telemedicina (in ambito Internet of Things) e da un Robot che si pone come mediatore/master tra la persona, l'ambiente circostante e l'ambiente esterno.

Tale infrastruttura può offrire sia servizi orientati alla persona nel suo ambiente domestico, monitorandone comportamenti e stato psico-fisico, sia servizi di telemedicina a supporto del monitoraggio remoto di pazienti sottoposti a terapie riabilitative da parte di medici o caregivers.

Nel seguito sono presentati, nel dettaglio, due dei principali risultati di ricerca che si basano su tecnologie di Intelligenza Artificiale.

Entrambi i risultati sono evidentemente superiori a quelli dell'attuale letteratura, dimostrando soluzioni più performanti e complete nei rispettivi settori di studio.

L'invecchiamento e il sovrappeso incrementano il numero di potenziali soggetti ad alto rischio ictus, soprattutto tra chi presenta ipertensione continua. Le tecnologie ICT possono fornire un valido supporto per la valutazione precoce, tanto più che una semplice visita medica dei pazienti affetti da ipertensione, piuttosto difficilmente riesce a prevedere la predisposizione all'ictus, e, di solito risulta particolarmente onerosa.

La soluzione studiata nell'ambito del progetto AMICO presenta un nuovo approccio basato sull'analisi di serie temporali per l'identificazione precoce degli aumenti degli stati ipertensivi, al fine di discriminare i pazienti ipertesi ad alto rischio cardiovascolare da quelli a basso rischio.

Il modello raggiunge risultati eccellenti in termini di accuracy della classificazione rispetto allo stato dell'arte riuscendo ad evidenziare le minuscole differenze tra i soggetti affetti da ipertensione.

Il secondo problema affrontato riguarda il processo di trattamento clinico che, tipicamente si articola nelle due fasi di degenza ospedaliera e di cure a domicilio dopo il ricovero. Il corretto completamento di tale processo è essenziale per la ripresa del paziente, ma diventa difficile per anziani e soggetti con

disabilità, fisica e/o cognitiva, che troppo spesso sono esposti ad errori di trattamento dovuti a dimenticanze di assunzioni di farmaci o da assunzioni di farmaci sbagliati.

Si presenta come soluzione, un sistema intelligente che, unico nella letteratura attuale, è in grado sia di supportare il paziente, quando è il momento di assumere il farmaco, con promemoria personalizzati in funzione delle sue capacità psico-fisiche (sms, messaggi audio, etc.), sia di allertarlo quando sta, erroneamente, assumendo un farmaco sbagliato, riducendo di molto il rischio di errore. Il nucleo della soluzione proposta consiste in un sistema multi-agente che si basa su algoritmi di Reinforcement Learning e di Deep Learning.

Di seguito il dettaglio delle sue soluzioni sviluppate.

4.1 Una Hybrid Deep Network per l'identificazione precoce di soggetti ad alto-rischio di ipertensione

Si stima che il 33% degli adulti soffra di ipertensione e che una persona su due tra essi *ignori* di esserne affetto. È ragionevole, di conseguenza supporre che regolari esami, per lo screening delle malattie cardiache facilitino una diagnosi precoce e riducano il rischio di ulteriori complicazioni associate a malattie cardiache, come l'ipertensione [\(34\)](#). Sebbene gli esami di screening siano una *conditio sine qua non* per la diagnosi precoce dell'ipertensione è necessario tenere presente che il processo di diagnosi è più complicato di quanto sembri [\(34\)](#). Programmi di screening per l'ipertensione hanno, certamente, dei vantaggi, ma anche questi possono presentare potenziali svantaggi come falsi positivi, lunghi tempi di attesa per appuntamenti medici, ansia, impatti psicologici e costi economici [\(35\)](#).

La diagnosi e il trattamento medico dell'ipertensione giocano un ruolo significativo nella riduzione del rischio di malattie cardiovascolari [\(36\)](#).

Il riconoscimento precoce dei sintomi nei soggetti ad alto rischio affetti da ipertensione è, tuttavia, abbastanza difficile, in quanto le anomalie, che possono essere considerate come indicatori dell'insorgenza di possibili problematiche sono di varia natura e, non sempre, facilmente quantificabili. Nella Figura 37:, ad esempio, si osservano le registrazioni del battito cardiaco di due soggetti affetti da ipertensione, uno, etichettato come *soggetto ad alto rischio*, presenta una singolarità e l'altro etichettato come *soggetto a basso rischio*. Il cerchio rosso, infatti, evidenzia un lieve cambiamento nei segnali che può essere, appunto, associato ad un'anomalia.

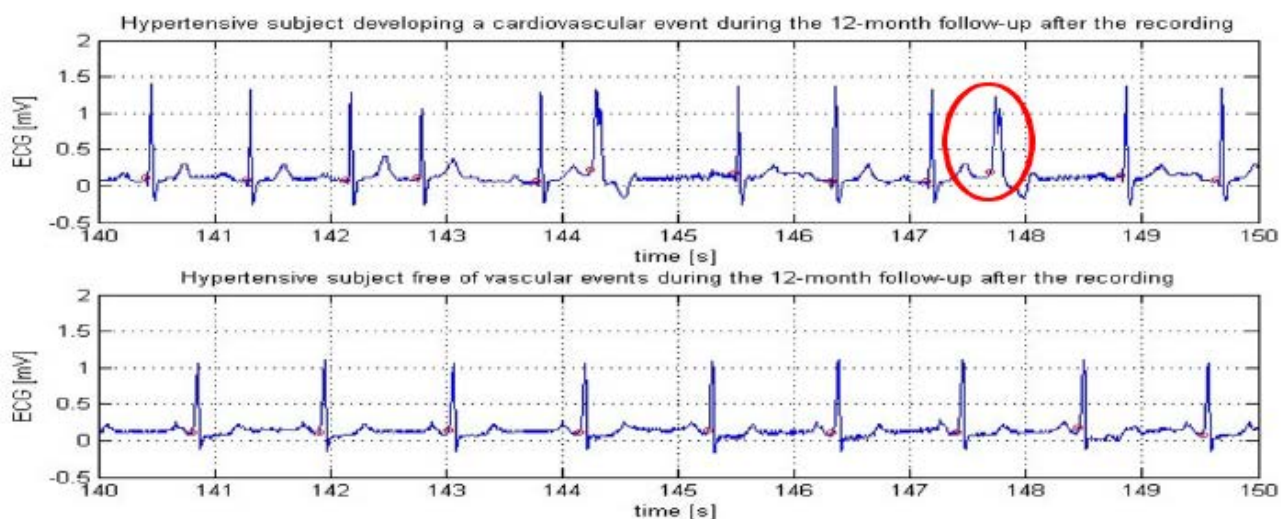


Figura 37: Confronto tra le registrazioni Holter di un soggetto ad alto rischio e uno a basso rischio (37)

Tali anomalie diventano evidenti, tuttavia, solo quando vengono valutati, quantitativamente, i segnali vitali, mediante un sistema di analisi. L'attenzione allo sviluppo di strumenti ICT per la diagnosi precoce e la prognosi delle cardiopatie è stata, pertanto incrementata.

Lo scopo di questi strumenti è *supportare* i cardiologi nelle attività diagnostiche, riducendo sia il numero di diagnosi mancati che il tempo impiegato per raggiungere decisioni.

Esistono vari esempi di tali strumenti di supporto come i questionari digitali [\(38\)](#), gli studi basati sui fattori di rischio [\(39\)](#) e le tecniche basate su parametri biologici [\(40\)](#), la maggior parte di queste tecnologie, in ogni caso adotta approcci di machine learning per estrarre informazioni utili dai dati.

Le indagini focalizzate sui parametri biologici identificano possibili anomalie dall'osservazione di specifici schemi di porzioni consecutive (denominate *window*) dei segnali fisiologici del paziente. Per tale ragione queste tecniche sono definite come un'analisi *window-by-window* dei segnali.

Sfortunatamente, progettare un modello di apprendimento automatico in grado di identificare anomalie nei segnali fisiologici può risultare complicato, non solo per la definizione poco chiara degli stessi patterns di anomalia, ma anche per il trade-off ingegneristico relativo alla progettazione delle *windows*. In questo contesto verrà presentata una nuova tecnologia, sviluppata nell'ambito del progetto AMICO, per l'analisi e la classificazione di un *intero segnale fisiologico*, in grado di risolvere, in modo ottimale, i problemi legati alla progettazione dei tradizionali approcci *window-by-window*. [\(33\)](#)

Nel seguito è riportata, in particolare, la descrizione di un nuovo approccio basato su *serie temporali* (TS) per l'identificazione precoce di uno stato ipertensivo, al fine di discriminare pazienti ipertesi ad *alto rischio* cardiovascolare da quelli a *basso rischio*. È stata, inoltre, progettata una rete ibrida di deep learning (HDN) come combinazione di tre differenti reti: un segnale di memoria a lungo termine (*Long Term Memory Signal* LTMS), una Convolution Neural Network (CNN) e una Deep Neural Network (DNN).

La *evaluation* dei modelli di previsione richiede grandi set di dati. Per questa ragione è stato preferito, il database SHARE [\(37\)](#) il cui download è stato effettuato da Physionet Repository [\(41\)](#).

Il database contiene più di 130 registrazioni ECG compresi i dati clinici di soggetti ipertesi monitorati per almeno 12 mesi.

I soggetti che hanno subito un evento vascolare - ad es. Infarto del miocardio - sono stati acquisiti come *soggetti ad alto rischio*, gli altri come *soggetti a basso rischio*.

I dati sperimentali dimostrano che il modello presentato nell'ambito del progetto AMICO raggiunge ottimi risultati in termini di accuratezza della classificazione. In aggiunta, un'elevata sensitivity e specifiche velocità sono state raggiunte nell'identificazione automatica di soggetti con esperienza di eventi vascolari entro un anno dalla registrazione ECG.

4.1.1 Stato dell'arte

Lo scopo di questa sezione è evidenziare il contributo della nostra¹¹ soluzione.

In primo luogo, presenteremo una panoramica dello *stato dell'arte* degli approcci alla valutazione dell'ipertensione, in modo da dimostrare che il nostro approccio è il primo basato su una classificazione TS dei segnali fisiologici.

Successivamente, presenteremo una panoramica di altri modelli di deep learning per la classificazione di una TS, con l'obiettivo di confrontarla con il nostro modello. I risultati sperimentali dimostreranno che la nostra soluzione presenta le migliori prestazioni rispetto agli altri approcci.

4.1.1.1 Approcci classici per la valutazione dell'ipertensione

La pressione sanguigna elevata è per lo più asintomatica, soprattutto nelle fasi iniziali e, per tale ragione è stata etichettata come *killer silenzioso* (42). L'assenza di sintomatologia, caratteristica dell'ipertensione richiede uno screening di routine della pressione sanguigna.

Il monitoraggio ambulatoriale della pressione sanguigna (Ambulatory blood pressure monitoring ABPM) è considerato un approccio gold-standard per la diagnosi di ipertensione, quando in un soggetto esaminato si riscontrata pressione alta. L'obiettivo dell'ABPM è scoprire la malattia in una sua manifestazione recente e curabile.

L'ABPM in genere richiede l'uso di una fascia di misurazione automatica e portatile, detta cuffia, che, indossata continuamente misura la pressione sanguigna ogni 15/30 minuti durante il giorno e ogni 15/60 minuti durante la notte (43).

¹¹ Da questo momento in poi è riportato fedelmente il testo dell'articolo "*A Hybrid Deep Network for the Early Identification of High-Risk Subjects with Hypertension*" presentato da Paragliola ing.G e Coronato prof. A. dell'ICAR di Napoli per il progetto AMICO (33) Si è ritenuto opportuno, pertanto esprimersi in prima persona plurale.

Nonostante la loro efficacia nella diagnosi, i monitoraggi ambulatoriali possono avere dei potenziali svantaggi come gli elevati costi di trattamento e i lunghi tempi di attesa per le visite mediche_(43). L'uso della cuffia di misurazione e di strumenti simili, inoltre può risultare scomodo sia nelle attività quotidiane che durante il sonno.

La stima dei principali fattori di rischio che influenzano l'ipertensione è un altro metodo utilizzato per supportare la diagnosi precoce.

Approcci di valutazione basati sull'apprendimento automatico sono stati utilizzati per l'analisi dei questionari e/o dei parametri biologici per la valutazione dei fattori di rischio. Ci sono molti studi che avvalorano queste tecniche come quelli condotti da Shobha & Nickolas nel 2018 _(44), da Wang et al. nel 2015 _(45), da Mohan et al. nel 2019 _(46).

L'identificazione dei fattori di rischio dipende, generalmente, da una considerazione delle caratteristiche tra cui il sesso, l'età, la pratica sportiva, l'indice di massa corporea (BMI), le abitudini alimentari, il fumo o il consumo di alcol (abuso di sostanze) e la storia familiare legata alle malattie. Lo scopo di questi metodi è definire il profilo del paziente supportando, così, una più precoce identificazione dei soggetti a più alto rischio.

Valutare i fattori di rischio è, considerevolmente, più complicato poiché possono essere significativamente differenti da una regione all'altra_(44).

Altri approcci si basano sull'applicazione di tecniche di machine learning per l'analisi dei comportamenti dinamici di un segnale vitale TS. Nel lavoro di Lehman et al. del 2015_(47) gli autori propongono un modello di previsione e stima dei rischi di mortalità nel tempo basato su TS della pressione sanguigna e della frequenza cardiaca. Nel lavoro di Melillo et al. del 2015_(37) gli autori stimano i metodi di analisi della variabilità, lineare e non lineare, della frequenza cardiaca e gli schemi di riconoscimento di pattern da registrazioni holter. Una predizione della variabilità della pressione sanguigna da dati TS, da valori della pressione sanguigna misurata a casa e da dati

acquisiti tramite una visita medica di valutazione in un ospedale è stata, invece, stimata da Koshimizu et al. nel 2020 [\[48\]](#).

Luo et al. nel 2018 [\[49\]](#) hanno sviluppato, per la previsione dell'ipertensione, una Convolution-Neural-network basata su uno schema di previsione di stati ipertensivi, utilizzando i dati sulla pressione sanguigna rilevati in un certo periodo di tempo.

In questi approcci è stato ottenuto, dai segnali TS, un gruppo di finestre temporali. La scelta della dimensione di una finestra e la sovrapposizione tra due finestre continue rappresentano un trade-off ingegneristico standard che rende il processo di progettazione complicato e dispersivo.

Approach	Data Type	Analytics Techniques	Pros	Cons
Ambulatory blood pressure monitoring	All the possible health information that a clinician would require	Medical examination	Gold-Standard	<ul style="list-style-type: none"> • Time Consuming • High costs • Long waiting time for an appointment
Risk Factors Analysis	Bio-parameters/ personal information/ History information	Questionnaires	Subject Profiling	Data strong depending on subject's native region
Vitals Signs Analysis	Subject's vital signs (Heart Rate, Blood Pressure, etc)	Windows-based Machine Learning Approaches	<ul style="list-style-type: none"> • Widely used • Strong supported in the literature 	Trade-off concerning the design of windows

Figura 38: Riepilogo dello stato dell'arte per la valutazione e la valutazione dell'ipertensione

La Figura 38: riporta una panoramica dei punti di forza e di debolezza di tutti gli approcci menzionati.

In questo documento, il nostro obiettivo è evidenziare e dimostrare che la nostra soluzione è l'unica ad adottare un approccio basato su classificazione TS per la valutazione di un aumento precoce nell'ipertensione.

4.1.1.2 Stato dell'arte dei modelli di classificazione TS

La Time Series Classification (TSC) è una sfida comunemente riscontrata. Esistono, attualmente, poche soluzioni che risolvono questo problema. Nella sezione precedente, abbiamo evidenziato che

l'applicazione di un approccio di classificazione delle serie temporali per la valutazione delle malattia da ipertensione rappresenta anche una tecnica innovativa per la stima delle stesse.

In questa sezione, presentiamo alcuni modelli ben noti per TSC, che abbiamo implementato nei nostri esperimenti per confrontarne il successo di previsione rispetto alla nostra soluzione.

I risultati dimostrano che il nostro modello ottiene i risultati migliori in termini d'identificazione precoce di un incremento in uno stato ipertensivo, al fine di discriminare i pazienti ipertesi ad alto rischio cardiovascolare dai pazienti ipertesi a basso rischio.

Fully Convolutional Neural Networks (FCN) [\[50\]](#) sono, per la maggior parte, delle reti convoluzionali (CNN) che non includono alcun strato locale di pooling (PL); la dimensione di una TS è mantenuta inalterata attraverso le convoluzioni. Una caratteristica importante delle FCN è, inoltre, la sostituzione di un strato finale Fully Connected (FC) con uno Global Average Pooling (GAP).[\[51\]](#).

Residual Network (ResNet) consiste in una rete che conta 11 strati, i cui primi 9 sono CN che continuano su un livello GAP che ne media l'output. I livelli sono raggruppati in unità logiche denominate *blocchi residui* (RB). Ogni blocco consiste di tre strati di convoluzione il cui output è sommato al seguente blocco di input e, conseguentemente, connesso al livello successivo. La rete è definita come tre blocchi residui il cui output alimenta uno strato GAP in cui l'ultimo livello è un classificatore softmax [\[50\]](#).

Encoder è una CNN ibrida profonda simile a FCN, ma con una significativa differenza. Un livello di attenzione sostituisce il livello GAP [\[52\]](#). Nello stesso modo delle FCN, questa rete è composta da tre CL con lievi modifiche. Il terzo strato convoluzionale viene fornito a un meccanismo di attenzione [\[53\]](#) che consente alla rete di scoprire in quali segmenti le TS sono significative per la classificazione.

Time Le-Net (t-LeNet) può essere considerato come una classica CNN con due strati di convoluzione che seguono una FC che alimenta un softmax. È possibile distinguere due varianti con FCN: la prima riguarda l'uso di uno strato FC invece del GAP, e la seconda riguarda l'operazione locale di max-pooling sull'output di ogni livello CNN. A differenza delle ResNet e FCN, questo approccio aumenta la quantità di parametri necessari per essere addestrata durante il processo di training, che si basa anche sulla dimensione del TS in ingresso. L'uso di reti t-LeNet è limitato poiché aumenta il numero e la lunghezza dei filtri scelti [\[54\]](#).

Multi Channel Deep Convolutional Neural Network (MCDCNN) è stata suggerita per set di dati TS multivariati (MTS) [\[55\]](#). L'architettura consiste in una CNN adattata per MTS dove l'operazione di convoluzione agisce indipendentemente per ognuna dimensione della serie temporale. Ogni dimensione dell'MTS alimenta due CL seguiti da uno strato di max-pooling, dopo di che l'output è appiattito e poi alimentato a un FC.

La rete *Time-CNN* è stata progettata da Zhao et al. nel 2017 [\[56\]](#). È possibile evidenziare tre modifiche rispetto alle altre architetture: I) Time-CNN adotta l'errore quadratico medio (MSE) come loss function al posto della category cross-entropy. II) l'ultimo strato è un FC con function di attivazioni sigmoide. III) Un livello di max-pooling locale viene sostituito con un'operazione di pooling a media locale. Tutte le architetture presentate mirano a classificare una TS. Ogni modello ha i suoi punti di forza e la letteratura mostra che ognuno di essi raggiunge buoni risultati. Tuttavia, il nostro modello include una caratteristica innovativa che lo rende diverso dalle altre architetture.

Il nostro modello è, infatti, l'unico che adotta uno strato ricorrente, la rete LSTM, come primo strato dell'intera rete. L'output di questi livelli viene connesso ad una catena di livelli CNN il cui output è, a sua volta inviato a un FC.

La giustificazione di questa scelta è correlata alla natura del TS in ingresso. Una registrazione elettrocardiografica holter è una classe di segnali in cui una sequenza di misurazioni viene costruita

campionando l'attività del cuore nel dominio del tempo. Tale sequenza può presentare una correlazione rilevante tra i vari campioni consecutivi che dovrebbero essere sfruttati per migliorare la precisione del nostro classificatore [\(57\)](#).

Tutte le architetture proposte adottano, come primi strati, una rete CNN che applica l'operatore convoluzionale a un *gruppo di campioni* grande quanto la dimensione dei filtri CNN. L'idea centrale del nostro lavoro è, in ogni caso valutare un modo diverso di farlo, analizzando la sequenza dei campioni.

Nella nostra soluzione adottiamo la rete LSTM che è naturalmente progettata per analizzare sequenze.

Tale rete, nel dettaglio elabora una sequenza di valori *un campione alla volta*, conservando, nel suo stato interno, una rappresentazione di dati che contengono informazioni riguardo tutti i passati elementi della sequenza stessa. Ogni campione (timestamp) in un input TS è, conseguentemente proiettato in uno spazio dimensionale superiore.

4.1.2 Contributo del nostro lavoro

Dopo la presentazione dello stato dell'arte relativo agli approcci, vale la pena riassumere il contributo della nostra soluzione.

- La letteratura mostra alcuni metodi incentrati sulla valutazione dell'ipertensione, soprattutto legati all'identificazione della malattia nelle sue fasi iniziali.

Uno studio della letteratura evidenzia, inoltre, la mancanza di approcci basata sul TCS dei segni vitali dei pazienti con ipertensione e per, questo motivo, nel nostro articolo proponiamo una soluzione basata su TSC per l'identificazione precoce dei soggetti ad alto rischio di ipertensione.

- Presentiamo un nuovo modello di rete profonda ibrida per affrontare TCS incorporando, all'interno della nostra rete, un componente specificamente progettato per l'analisi della sequenza di campioni.

Abbiamo confrontato il nostro modello con lo stato dell'arte delle applicazioni basate su approcci TS.

I risultati dimostrano la superiorità del nostro modello in merito al raggiungimento di prestazioni più performanti.

4.1.3 Metodo

4.1.3.1 Descrizione dei dati

Abbiamo adottato un database contenente 24 ore di registrazioni di elettrocardiografiche (ECG) holter di 139 soggetti ipertesi (49 donne e 90 uomini, di età media 72 - / + 7 anni).

I soggetti sono stati monitorati per 12 mesi, dopo le registrazioni, per prendere atto di eventi cardiovascolari maggiori (cioè ictus fatali o non fatali).

Nel follow-up dei 12 mesi successivi alle registrazioni, 17 pazienti hanno riportato un evento registrato (11 infarti miocardici, 3 ictus e 3 eventi sincopali) e per questo motivo, tali soggetti sono stati considerati ad *alto rischio* (HRP), mentre i restanti soggetti a *basso rischio* (LRP) [\(37\)](#).

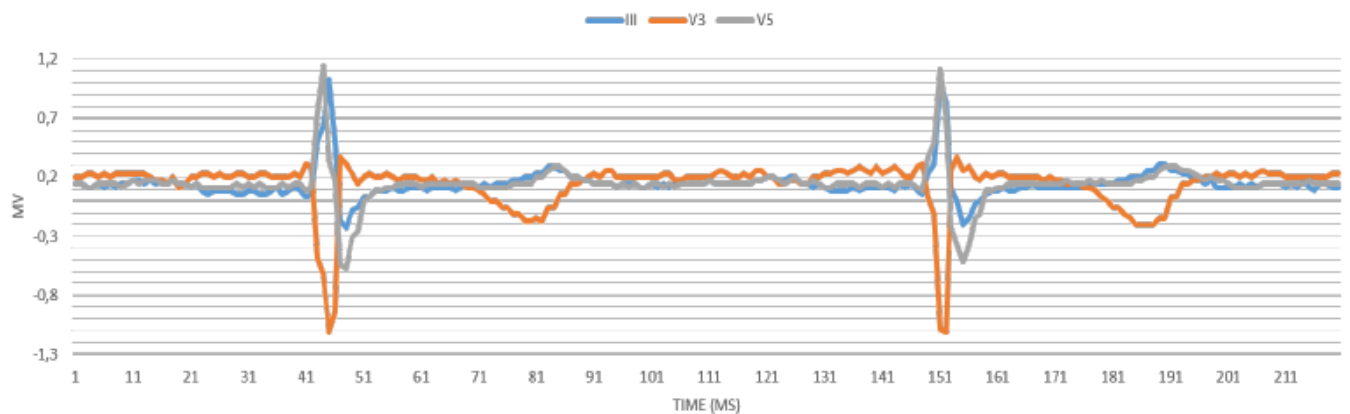


Figura 39 : Esempio di una delle forme d'onda ECG memorizzate nel dataset i

La Figura 39 mostra un esempio di estrazione dei dati registrati dai soggetti.

La Figura 38: abbiamo visto che evidenzia le minuscole differenze tra un soggetto ad alto rischio e un soggetto a basso rischio. Il cerchio rosso mostra, infatti, un leggero cambiamento nel segnale che potrebbe essere interpretato come un'anomalia nella registrazione.

La Figura 39 mostra le strutture dei dati utilizzati per il training. Il segnale è un TS multivariate (MTS) dove ogni secondo del segnale grezzo modella un time-point (o esempio di training). Ogni time-point è un vettore di features spazio-temporale ad albero in cui ogni caratteristica è una differente TS di una registrazione elettrocardiografica holter acquisita utilizzando varie tecniche [\(37\)](#).

4.1.3.2 Hybrid Deep Model

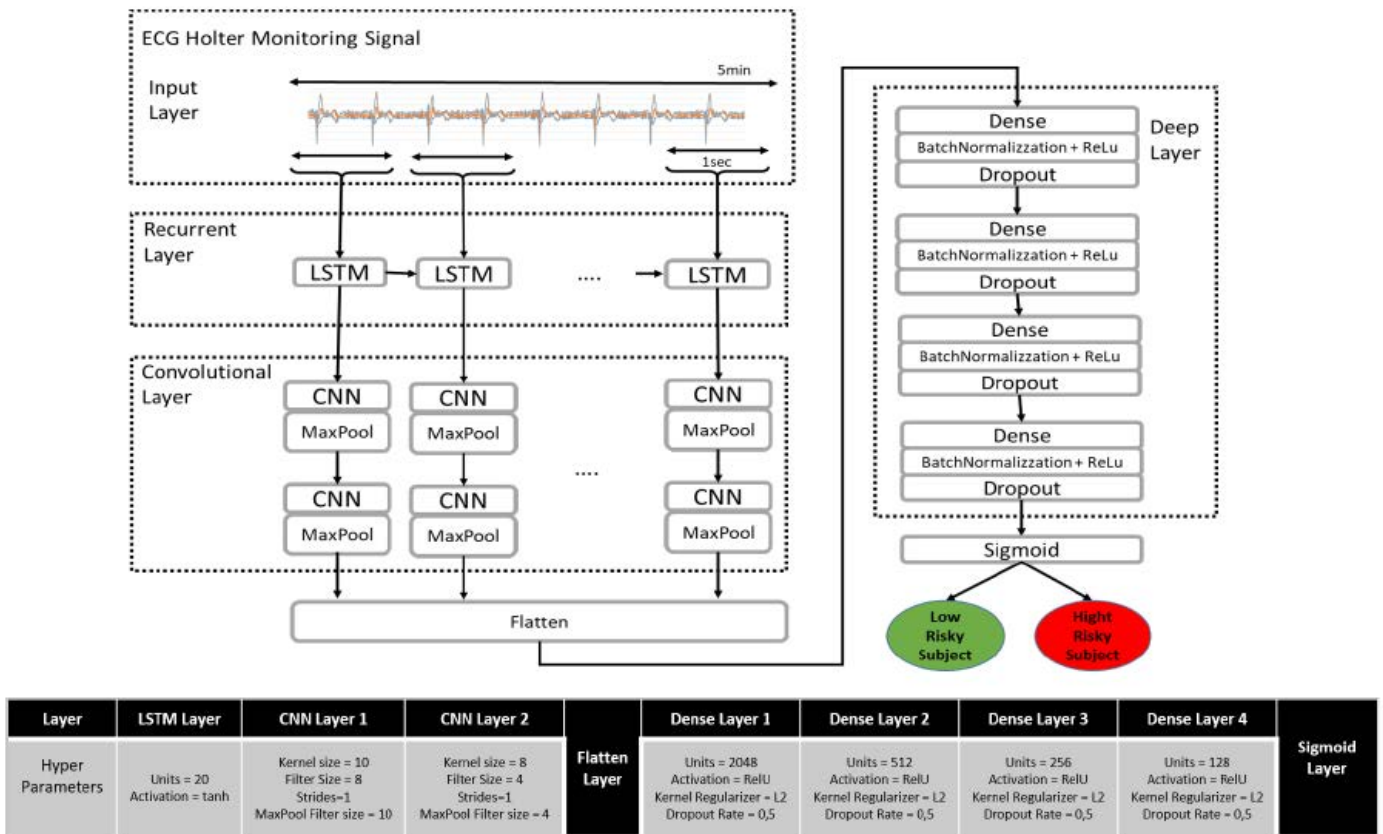


Figura 40 : Panoramica del modello ibrido e riepilogo degli iperparametri

La Figura 40 descrive una panoramica della rete ibrida proposta.

Al livello più alto, lo strato di input, il modello è alimentato dalle serie temporali TS.

Il modello è composto da tre componenti.

- Lo strato ricorrente mira ad affrontare le correlazioni temporali tra i time-points della TS. Questo livello logico è definito come una *Long Short-Term Memory network* (LSTM). Le sue capacità di modellare le correlazioni di una sequenza temporale costituiscono il motivo principale per cui abbiamo preferito tali reti per questo studio [\(58\)](#).

Le reti LSTM sono definite come una catena di celle unitarie per l'analisi di una sequenza di dati x_1, \dots, x_t dove $t \in [1, T]$, con T lunghezza della sequenza.

Il valore generico x_i in un time-point i viene elaborato da una cella unitaria.

Ogni punto temporale, sia dell'input che dell'output TS ha ancora la forma di un vettore di caratteristiche, la cui lunghezza cambia a seconda della dimensione del filtro della cella unitaria.

L'output della LSTM è stato caratterizzato considerando le features dei dati di input da ordinare per tempo.

Lo stato delle unità generiche al tempo t può essere espresso come:

$$y_t = \phi(W \cdot x_t + U \cdot h_{t-1} + b)$$

dove ϕ è la funzione sigmoide, e W , U e b sono i parametri che devono essere impostati durante il training, h_{t-1} è lo stato della precedente unità.

Una LSTM conta un livello con una dimensionalità dello spazio di output di ogni cella uguale a 10 unità. Ciascun output di una cella viene fornito come input allo strato convoluzionale.

- Lo strato convoluzionale è incaricato di generare una rappresentazione ridotta dell'input mediante l'estrazione di features più informative dai dati d'ingresso. Questo strato è definito mediante due blocchi sovrapposti, ciascuno composto di una *rete convoluzionale* (C) e di uno strato *max-polling* (P). La prima ha una dimensione di kernel di 10 unità sia per C che per P, mentre il secondo livello ha 8 unità.

Uno strato convoluzionale adotta specifici filtri per l'elaborazione delle partizioni locali di input, all'interno dei quali è applicato l'operatore convoluzionale. Questi filtri sono duplicati insieme all'intero input. Una fase di sotto-campionamento (pooling) produce una versione

con risoluzione inferiore dell'output del CL filtrando il valore massimo da diverse regioni locali.

Lo strato convoluzionale, in conclusione estrae le features, più rilevanti per ogni secondo, dall'ingresso TS.

Questo blocco può essere formalmente espresso come:

$$y_{conv} = W \otimes x + b$$

$$y = MaxPolling(y_{conv})$$

Dove \otimes è l'operatore convoluzionale e *MaxPolling* è un operatore di discretizzazione basato su sotto-campioni.

L'output dello strato di convoluzione viene immesso nel deep layer, che trasforma lo spazio delle features in modo da rendere il suo output più facile da classificare.

- Il *Deep Layer* è composto da una Deep Neural Network a quattro stacked (DNN). Ogni livello è definito come uno strato completamente connesso. Ciascuno strato *fully-connected* ha un numero di neuroni dimezzato rispetto al precedente ed è seguito da una *bath normalization* (BN) e da strati di *dropout*. Uno strato di unità lineare rettificata (ReLU) viene adottato come funzione di attivazione per prevenire la saturazione del gradiente.

Lo strato di normalizzazione è applicato per accelerare la velocità di convergenza e coadiuvare il miglioramento della generalizzazione, mentre gli strati di dropout per aumentarne la capability intesa, appunto, come capacità di generalizzazione.

Un livello di base completamente connesso è formalizzato come

$$y_{fc} = W x + b$$

$$y_{bn} = BN(y_{fc})$$

$$y = ReLU(y_{bn})$$

Il *Deep Layer* termina con uno strato sigmoideo che classifica l'input come soggetto ad alto rischio oppure come soggetto a basso rischio.

Un'analisi completa dell'intera struttura di tutti gli strati è al di fuori dello scopo di questo documento. Per maggiori dettagli, il lettore può fare riferimento agli studi di Lipton del 2015 (58), di Orbach del 1962 (59) e di Goodfellow et al. del 2016 (60).

4.1.4 Risultati

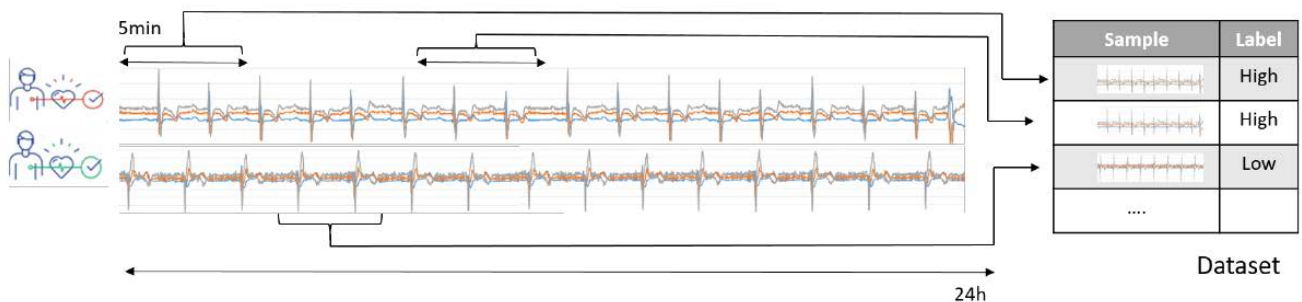


Figura 41: Estrazione del campione di training dal dataset

4.1.4.1 Training

In questa sezione introduciamo sia il training dataset che il processo di apprendimento del nostro modello.

I dati di training sono estratti dal dataset scaricato dalla repository di Physionet.

Le registrazioni personali hanno una durata di circa 24 ore ciascuna e contengono tre segnali ECG, ciascuno campionato a 128 campioni al secondo.

Un frammento di 5 minuti registrato durante il giorno è stato selezionato casualmente, senza replica, per ogni soggetto. Un punto è stato selezionato casualmente da ogni secondo.

Ogni frammento è un singolo campione del training dataset. Inoltre, ogni campione è stato etichettato con un tag High o Low in base al soggetto da cui è stato selezionato (Figura 41)

Il numero di campioni selezionati da ciascuna registrazione è stato differenziato tra soggetti ad alto rischio e soggetti a basso rischio, per garantire un giusto bilanciamento tra le due classi.

Da quelle a *basso rischio* abbiamo selezionato 40 campioni. Da quelle ad *alto rischio*, invece, abbiamo selezionato 160 campioni in modo che il training set sia composto da più di 7000 campioni (ovvero serie temporali).

Abbiamo ripetuto questo processo sia per il training set che per il test set.

Abbiamo adottato un ciclo di convalida incrociata di 10 volte eseguito per il modello di ottimizzazione (ovvero un'ottimizzazione di iperparametri di Tuning), mentre un hold-out test set è stato utilizzato per ottenere stime imparziali delle prestazioni di classificazione reali.

Per il test set, abbiamo selezionato un campione da ciascuna registrazione dei soggetti a basso rischio e tre campioni da ciascuna di quelle dei soggetti ad alto rischio. Pertanto, il test set è composto da 177 campioni.

4.1.4.2 Valutazioni delle performance

Vale la pena ricordare che i risultati dimostrano l'alta qualità del nostro modello nella classificazione di un TS come soggetto ad alto rischio o soggetto a basso rischio.

Il nostro modello si presenta come una classificazione binaria.

Per la fase di addestramento, abbiamo suddiviso, in modo casuale, l'intero training dataset definendo un training set come il 90% del set originale mentre il validation set è stato definito come 10%.

Per la valutazione del nostro modello sono state utilizzate quattro metriche: precision, recall, accuracy e F1-Score (JML).

Queste misurazioni sono definite come segue:

$$\begin{aligned}
 \text{Recall} &= \frac{TP}{TP + FN} \\
 \text{Precision} &= \frac{TP}{TP + FP} \\
 \text{Accuracy} &= \frac{TP + TN + FP + FN}{TP + TN + FP + FN} \\
 F1 &= 2 * \frac{\text{Precision} + \text{Recall}}{\text{Precision} * \text{Recall}}
 \end{aligned}$$

dove

- TP: veri positivi (cioè il numero di soggetti ad alto rischio identificati)
- FN: falsi negativi (ovvero soggetti ad alto rischio erroneamente identificati come soggetti a basso rischio)
- TN: veri negativi (cioè soggetti a basso rischio adeguatamente identificati)
- FP: falsi positivi (ovvero soggetti a basso rischio erroneamente identificati come soggetti ad alto rischio)

L'AUC (Area under the ROC Curve) è tutta l'area bidimensionale sotto l'intera curva ROC.

Per gli esperimenti comparativi abbiamo implementato tutte le architetture TSC presentate nei relativi lavori basati sulle indicazioni riportate nei rispettivi documenti.

Lo scopo di questa sezione è confrontare le prestazioni del nostro modello rispetto agli altri modelli.

Il training set ed il test set sono stati impostati nello stesso modo per tutte le architetture.

Per ottenere le migliori prestazioni abbiamo cercato la migliore configurazione della nostra rete attraverso il tuning degli iperparametri delle reti.

La messa a punto (tuning) del nostro modello è stata ponderata mediante un approccio grid-search con una view per impostare i parametri in:

- Neurone per strato: {512;1024;2048;4096}
- Learning rate: {0.01; 0.05; 0.001; 0.005; 0.0001}
- Epoche: {1000; 1500; 2000, 2500}
- Dimensione del convolutional kernel: {5, 10, 15}
- Dimensione del polling filter: {5, 10, 15}

Un approccio di analisi sulla griglia consiste in una ricerca, molto approfondita, in un dato sottoinsieme dello spazio iperparametrico.

Architecture	AUC			Accuracy			Sensitivity			Specificity			F1-Score		
	Worst	AVG	Best	Worst	AVG	Best	Worst	AVG	Best	Worst	AVG	Best	Worst	AVG	Best
(Melillo et al. (2015b))	-	-	0.88	-	-	0.85	-	-	0.71	-	-	0.87	-	-	-
FCN (Wang et al. (2016))	0.79	0.84	0.86	0.72	0.73	0.74	0.78	0.78	0.79	0.8	0.8	0.81	0.77	0.77	0.78
ResNet (Wang et al. (2016))	0.8	0.84	0.875	0.80	0.8	0.81	0.8	0.8	0.8	0.78	0.79	0.8	0.75	0.75	0.76
Encoder (Serra et al. (2018))	0.77	0.85	0.88	0.8	0.81	0.82	0.75	0.76	0.76	0.85	0.87	0.88	0.76	0.75	0.77
LeNet (Guenec et al. (2016))	0.77	0.83	0.87	0.75	0.77	0.8	0.72	0.73	0.74	0.8	0.83	0.85	0.74	0.75	0.75
MDCNN (Zheng et al. (2014))	0.85	0.87	0.90	0.85	0.86	0.88	0.72	0.74	0.76	0.93	0.91	0.9	0.77	0.79	0.8
Time-CNN (Zhao et al. (2017))	0.88	0.93	0.95	0.88	0.9	0.92	0.87	0.88	0.9	0.87	0.9	0.92	0.82	0.85	0.87
Our Solution	0.97	0.97	0.98	0.97	0.98	0.99	0.96	0.97	0.98	0.97	0.97	0.98	0.96	0.97	0.99

Figura 42: Panoramica dei Performance Results

La Figura 42 mostra i risultati della classificazione.

Per ogni architettura riportiamo i migliori risultati ottenuti, i peggiori e la media.

I migliori risultati si riferiscono al caso di test in cui la fase di addestramento è stata fissata a 2500 epoche, mentre i peggiori a 1000 epoche. La colonna AVG descrive la media di tutti i casi di test eseguiti.

La prima riga riporta i risultati degli studi di Melillo et al. del 2015 [_\(37\)](#). Gli autori adottano un approccio non-TSC, ma abbiamo ritenuto, comunque opportuno includerli, perché hanno sviluppato il dataset ed eseguito, per primi, esperimenti su di esso, in modo da tracciare una linea di base per il nostro lavoro. Non è riportato un valore per il dato di F1-score in quanto gli autori non lo hanno calcolato.

Considerando il caso migliore, i risultati mostrano che il nostro modello supera tutte le altre soluzioni raggiungendo una precisione del 98%. Il miglioramento medio dell'accuracy è di circa il 13% e va da un minimo del 6,12% rispetto ai risultati di Zhao et al. del 2017 [_\(56\)](#) fino ad un massimo del 18,37% rispetto a quelli di Guennec et al. del 2016 [_\(54\)](#).

L'alta qualità del modello è, per di più confermata anche dai valori di recall e precision, entrambi superiori di circa il 95%!

La F1-score raggiunge il 96% confermando, così, un buon comportamento del modello rispetto ai falsi positivi e ai falsi negativi. Considerando il caso peggiore, i risultati sono comunque migliori con un aumento medio dell'accuracy del 16%.

Di seguito sono riportati i risultati ottenuti nei migliori casi di test.

A) Our Model		
	T _{Predict}	F _{Predict}
T _{True}	53	2
F _{True}	2	120

B) Time-CNN		
	T _{Predict}	F _{Predict}
T _{True}	51	4
F _{True}	6	116

C) MDCNN		
	T _{Predict}	F _{Predict}
T _{True}	43	12
F _{True}	10	112

D) ResNet		
	T _{Predict}	F _{Predict}
T _{True}	47	8
F _{True}	12	110

E) FCN		
	T _{Predict}	F _{Predict}
T _{True}	45	10
F _{True}	8	114

F) LeNet		
	T _{Predict}	F _{Predict}
T _{True}	45	10
F _{True}	10	112

G) Encoder		
	T _{Predict}	F _{Predict}
T _{True}	44	11
F _{True}	10	112

Figura 43 : Panoramica delle Confusion Matrix di ogni architettura

La Figura 43 mostra la matrice di confusione di ciascuna architettura.

La matrice A riporta i risultati medi della classificazione sul test set del nostro modello. Esso risulta in grado di discernere tra le due classi con un'elevata precisione.

Il test set è composto da 156 campioni. Il nostro modello identifica i soggetti a basso rischio (i casi falsi) con un'accuratezza del 98% (120 su 122) e quelli ad alto rischio (i casi veri) con una precisione del 96% (53 campioni su 55).

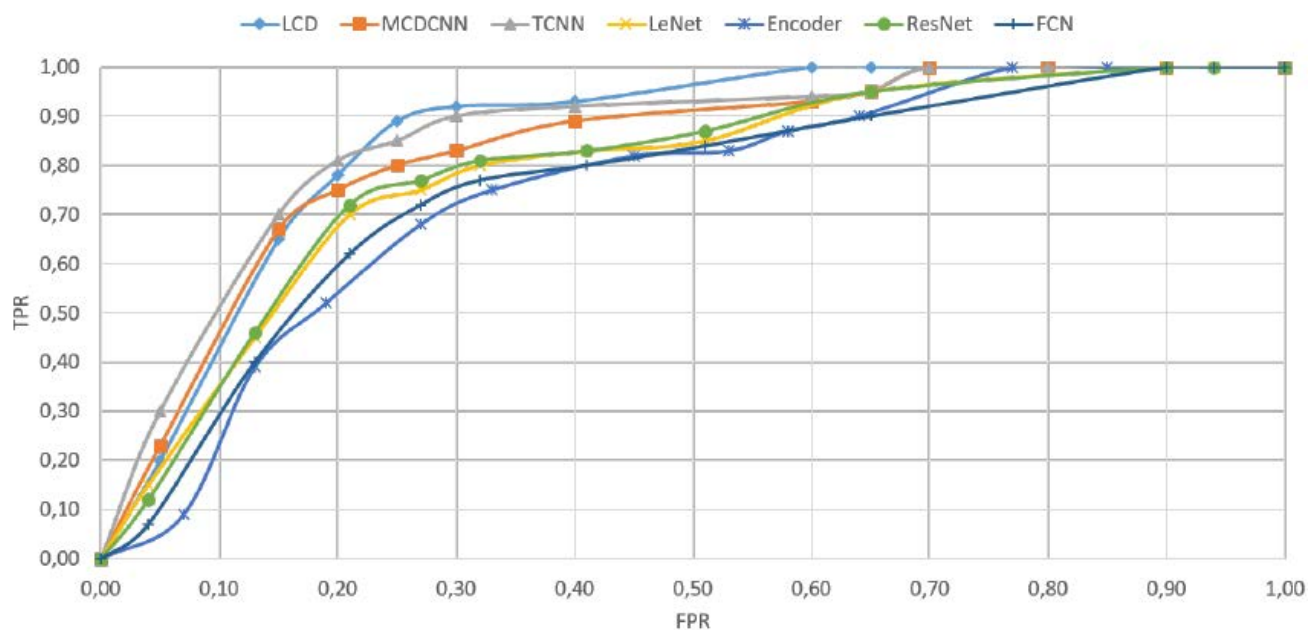


Figura 44 : Overview delle ROC curves

La Figura 44 mostra la curva di ROC che evidenzia le prestazioni del nostro modello di classificazione.

Le curve di tutte le soluzioni hanno un andamento molto simile. Tuttavia, il nostro modello manifesta un'AUC più grande rispetto a tutte le altre architetture.

4.1.5 Discussione dei Risultati Ottenuti

Abbiamo confrontato il nostro approccio con altre ben note architetture per l'identificazione di individui ad alto rischio in una popolazione di pazienti ipertesi. I risultati mostrano l'elevata qualità della nostra soluzione. Tuttavia, una discussione onesta, considerando i punti di forza e di debolezza, è necessaria.

Nella Figura 44 è riportata la curva ROC.

Il nostro modello raggiunge i valori AUC più elevati, pari al 98%, superiori al 95% ottenuto da Zhao et al. nel 2017 [\[56\]](#). L'AUC della nostra soluzione è, di conseguenza, migliore del 3%, in aggiunta ad un ulteriore miglioramento medio del 9,7%.

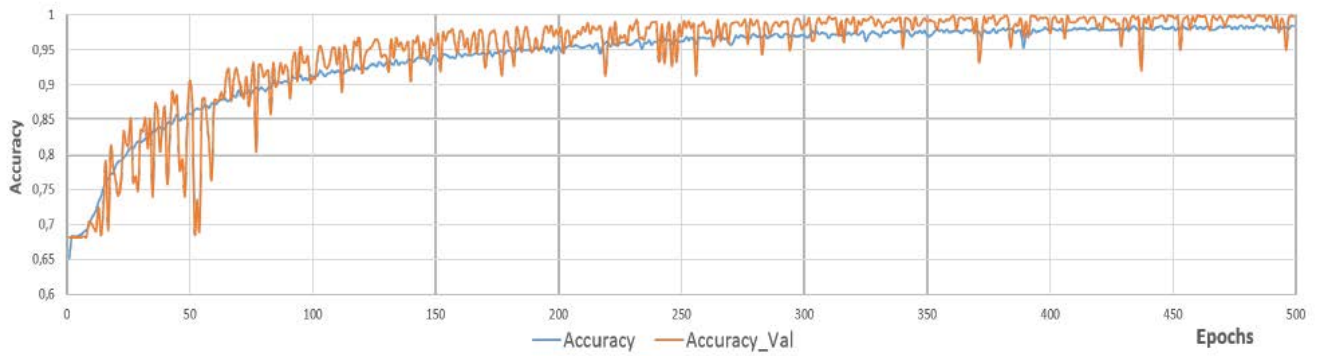
È importante notare che esiste un punto d'incrocio nell'andamento dei Falsi Positivi (FPR), equivalente al 25%. Successivamente a questo punto, il nostro modello funziona meglio con un aumento netto dello FPR e, di conseguenza, una risposta, in termini di recall, migliore. Questo dimostra che il nostro modello è in grado di evitare meglio i Falsi Positivi, cioè i soggetti ad alto rischio che sarebbero, erroneamente interpretati come soggetti a rischio basso.

Come già indicato, prima di questo punto la nostra soluzione raggiunge risultati molto simili a quelli di Zhao et al. [\[56\]](#) e di Zheng et al. del 2014 [\[55\]](#) con, comunque, dei miglioramenti, seppur trascurabili.

Dalle matrici di confusione di ciascuna architettura, riportate nella Figura 43, vale la pena notare che il nostro modello (matrice A) raggiunge una buona prestazione per quanto riguarda i Falsi Positivi. La matrice riporta, infatti, solo 2 Falsi Positivi che producono un miglioramento di oltre il 200% rispetto al miglior risultato dell'altra architettura Zheng et al. (matrice B) con 6 Falsi Positivi.

Il numero medio di Falsi Positivi è intorno a 9 e, di conseguenza, il nostro modello raggiunge, in generale, risultati migliori.

Per quanto riguarda i Veri Negativi non si evince un chiaro miglioramento, ne consegue che il nostro modello risultata molto vicino alle altre architetture.



(a) Trend of the Accuracy

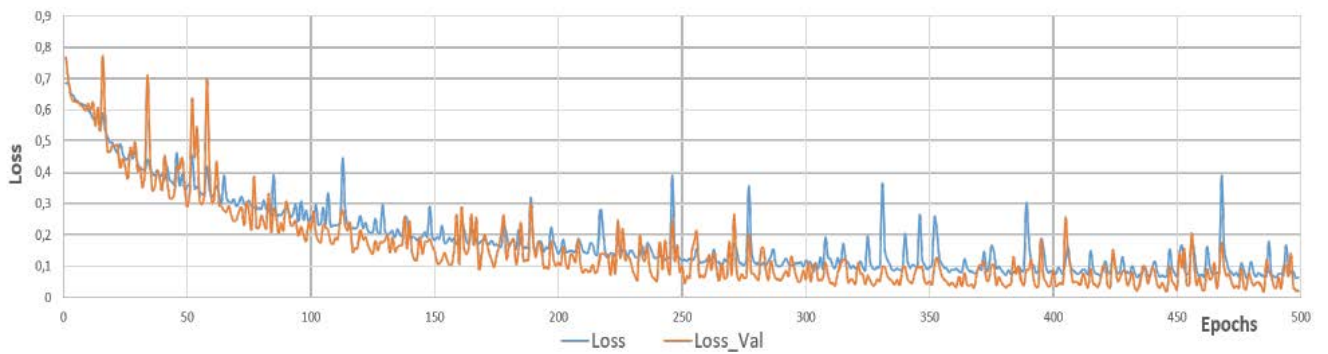


Figura 45 : Trend di Loss e Accuracy del Modello Comparato con il Validation Set

Le Figura 45 *a* e *b* mostrano le tendenze dell'accuracy e della loss function del nostro modello durante il training. E' possibile osservare che la curva del training set e quella del validation set sono molto vicine tra loro.

Questo risultato dimostra che i modelli non sono affetti da problemi di overfitting.

All'inizio del training process, il learning process mostra un trend di oscillazione che diventa più lineare dopo la 250a epoca.

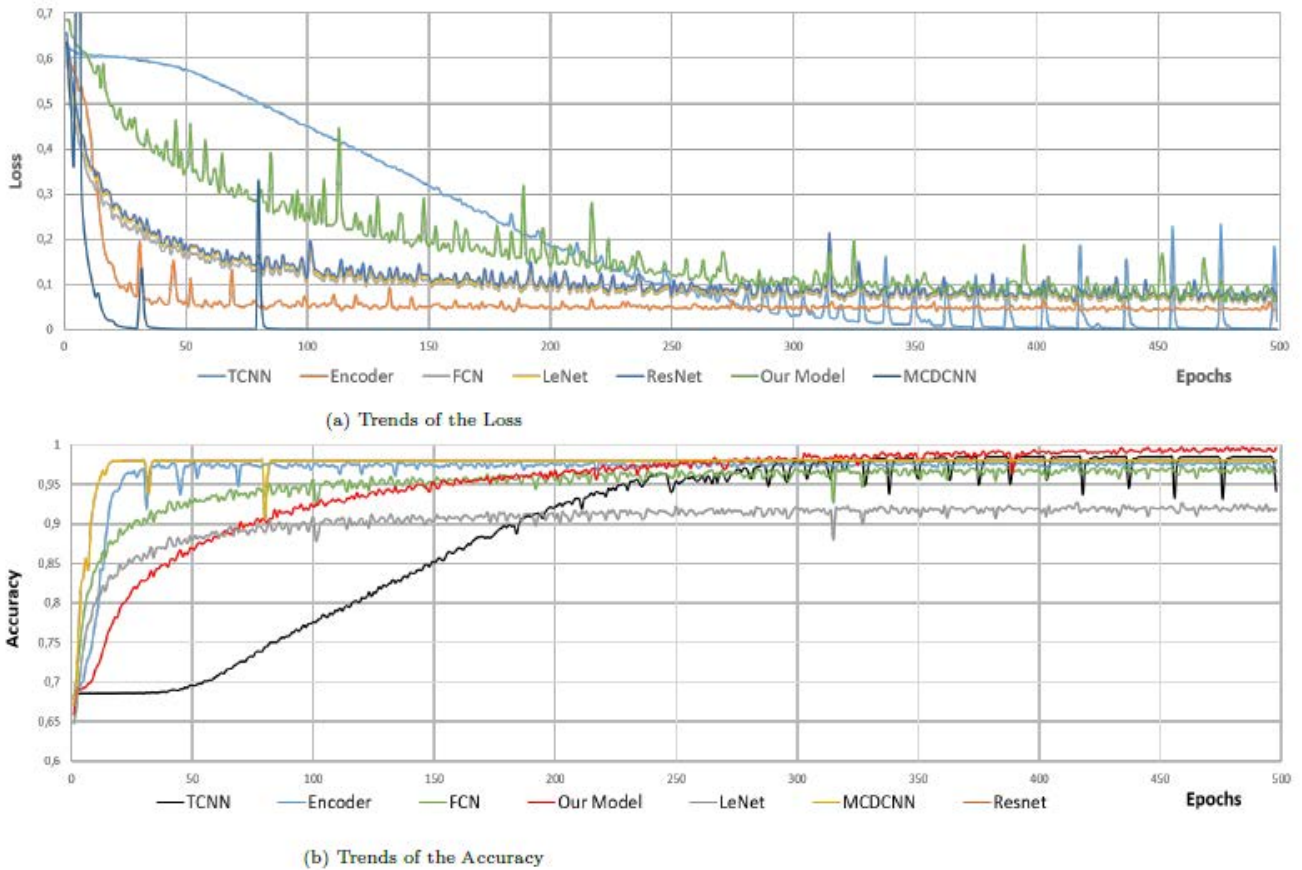


Figura 46 : Confronto tra le tendenze di Loss and Accuracy di tutte le architetture

Le Figura 46 *a* e *b* mostrano le tendenze dell'accuracy e della loss function della nostra soluzione rispetto alle altre architetture.

Le figure dimostrano che il nostro modello converge più lentamente degli altri.

Da questo risultato si evince che il nostro modello raggiunge un minimo locale più lentamente e, da quel punto richiede più epoche.

Dopo 100 epoche il valore di loss del nostro modello è più elevato della maggioranza degli altri: + 99% (MDCNN), + 76% (Encoder), + 48% (FCN), + 45% (LeNet), + 43% (ResNet) e -82% (TCNN).

Ogni modello raggiunge un punto stabile dopo 250 epoche, ma anche a questo livello il nostro modello non mostra la loss migliore. Nell'ultima epoca il gap-loss risulta + 90% (MDCNN), + 73% (TCNN), 36% (Encoder), 5% (FCN), -0,08% (LeNet), -4% (ResNet).

La lenta convergenza del nostro risultato è, nettamente ricompensata da una migliore accuracy.

La Figura 46b mostra come, alla fine del training, l'accuracy raggiunta dal nostro modello sia, evidentemente, la migliore.

Per quanto riguarda il miglioramento della classificazione dell'accuracy, la nostra soluzione è migliore di: + 5% (TCNN), + 1.7% (Encoder), + 2.9% (FCN), + 7% (LeNet) +4% (ResNet) e +1.6%(MDCNN).

Architecture	#Trainable Parameters	Time Duration(hh)
FCN	267,905	4,34
ResNet	524,365	8,68
Encoder	1,091,585	14,45
LeNet	786,684	8,07
MDCNN	473,672	2,35
Time-CNN	396,485	3,37
Our Solution	1,254,509	18,64

Figura 47 : Panoramica della durata del tempo di addestramento di ciascuna rete

La Figura 47 racchiude una panoramica della durata del tempo di training di ciascuna rete. La colonna *Trainable Parameters* riporta il numero di parametri interni di ognuna di esse generati durante il training. Il nostro modello è la rete più grande con più di 1,2 milioni di parametri, e di conseguenza prevede un tempo di training più lungo.

Questi risultati evidenziano come il nostro modello sia in grado di ottenere le migliori prestazioni in termini di discernimento tra soggetti ad alto rischio e soggetti a basso rischio. Tuttavia, mostra un trend lento durante il training. Questo aspetto incrementa la quantità totale di tempo necessaria al modello stesso per raggiungere un punto di minimo locale, lì dove, le altre soluzioni raggiungono più velocemente un punto sub-ottimale.

Dagli esperimenti abbiamo appreso alcuni punti che esamineremo in profondità nelle attività future, tra cui:

- ridurre il numero di parametri. L'elevato numero di parametri rende il processo di training molto lungo. Questa rappresenta la principale questione, relativa al nostro approccio che stiamo progettando di indagare nel nostro prossimo lavoro. Un ventaglio di soluzioni includerebbe la modifica della struttura della rete attraverso la modifica degli strati.
- aumento della dimensione del dataset. Il set di dati utilizzato per il training è un buon punto di partenza. Tuttavia, stiamo progettando di testare il nostro modello con altri dataset più grandi che descrivono la stessa classe di dati.

4.1.6 Conclusioni

In questo lavoro abbiamo presentato un approccio basato sulle Serie Temporal per l'identificazione precoce di soggetti ad alto rischio affetti da ipertensione.

Tra gli altri approcci per la valutazione dell'ipertensione, il nostro è il primo che adotta un TSC per differenziare i soggetti ad alto da quelli a basso rischio.

Abbiamo confrontato il nostro modello con altre architetture di rete profonda per la classificazione della TS ed i risultati sperimentali dimostrano che il nostro modello ottiene prestazioni migliori in termini di accuracy della classificazione. L'elevato numero di parametri da noi utilizzati comporta, tuttavia, un maggior impiego di tempo per completare il training.

L'analisi dell'intera TS mostra risultati promettenti in termini di evidenziazione delle minuscole differenze tra i soggetti affetti da ipertensione.

4.2 Sistema intelligente per il supporto di pazienti con disabilità nel trattamento domiciliare basato su approcci di Reinforcement Learning e Deep Learning

La recente politica dei sistemi sanitari nazionali tende a migliorare la qualità della vita dei malati e a limitare i costi sanitari, riducendo il tempo di permanenza in ospedale e aumentando l'assistenza domiciliare. Ciò richiede, da parte dei pazienti, la possibilità di seguire i propri piani terapeutici anche a casa in modo controllato. Purtroppo, l'aderenza alla terapia, che è definita come *"la misura in cui il paziente segue le istruzioni mediche"* è compresa, orientativamente, solo tra il 50% e lo 80% (62).

L'Organizzazione Mondiale della Sanità, ha stimato che l'aderenza alla terapia, in particolare nelle case di cura è profondamente minata da errori terapeutici (63) cioè da *"ogni evento avverso, indesiderabile, non intenzionale, prevenibile che può causare o portare ad un uso inappropriato del farmaco o ad un pericolo per il paziente."*

La stessa OMS ha classificato diversi tipi di errori terapeutici riconducibili a più fattori - quali l'omissione o l'assunzione di un farmaco sbagliato; la somministrazione in base alla frequenza, alla dose, alla modalità; lo stesso paziente - e ha esaminato diversi studi che riportano dati sugli effetti di tali errori nelle case di cura (63) dichiarando che, in particolare, *"nella popolazione anziana si possono anche riscontrare specifici problemi legati agli errori di somministrazione della terapia farmacologica. Le persone che vivono in case di riposo, infatti sono spesso fragili a causa di molteplici condizioni di salute e assumono, per questo, più farmaci."* Tra i fattori che contribuiscono agli errori terapeutici sono stati inclusi lo stato mentale del paziente, la mancanza di conoscenza del farmaco e le menomazioni fisiche che rendono difficile la somministrazione della terapia.

Capire quale sia il modo migliore per ricordare di assumere il farmaco specifico al paziente, nonché verificare la sequenza di azioni da lui compiute per l'assunzione ed in particolare verificare il farmaco che sta, effettivamente, assumendo può realmente incrementare il grado di aderenza al trattamento pianificato e ridurre, così, il numero di errori terapeutici e, quindi, di eventi avversi ad essi legati.

A tal fine è necessario personalizzare un sistema di supporto in funzione delle attitudini¹² dello specifico paziente.

Il contributo del seguente documento affronta questo problema.

E' stato preferito un approccio ibrido che adotta sia tecniche di Reinforcement Learning che di Deep Learning, per addestrare una serie di Agenti Intelligenti sia per supervisionare il processo di trattamento domiciliare degli anziani e dei malati con disabilità, che per migliorare l'aderenza alla terapia farmacologica.

In particolare, tra gli altri, l'approccio include un *self-learning agent*, chiamato *Tutor*, che è in grado sia di auto-apprendere le attitudini del soggetto che di auto-adattare il modo in cui comunica con lo stesso.

Questo agente dipende da un algoritmo di apprendimento per rinforzo che gli permette di personalizzare il modo in cui comunicare con uno specifico paziente in base alle sue attitudini.

Un altro agente, chiamato *Checker*, fa affidamento su un classificatore basato su Deep Learning e può identificare il farmaco che il paziente sta per assumere e avvisarlo in caso sia sbagliato.

¹² Il termine inglese *skill* può assumere, in italiano un ventaglio di significati, più profondi rispetto alla semplice traduzione di *abilità*. In questo contesto, che tratta, prevalentemente, di persone anziane e/o con disabilità, il termine individuato di *attitudine* è inteso sia come competenza, ad esempio tecnico/culturale per l'uso di una tecnologia quanto, più semplicemente, come capacità psico-fisica di approcciare ad un trattamento, ma anche come *predisposizione comportamentale* a seguire le regole e le abitudini.

4.2.1 Motivazioni¹³

L'impatto degli errori farmacologici nei piani di trattamento terapeutico è stato riesaminato e dimostrato in molte indagini, ad esempio da Barber et al. nel 2009 (64) o da Matthew Perri et al. nel 2005 (65).

Tali studi dimostrano la necessità di soluzioni in grado di sorvegliare i processi di trattamento farmacologico domestici. Il compito, tuttavia, è impegnativo a causa della diversità dei soggetti; gli anziani soprattutto, o coloro che presentano disabilità cognitive possono incontrare difficoltà nel seguire un trattamento specifico. Le disabilità possono ridurre la capacità di comprendere il senso di un messaggio scritto, così come la perdita di memoria o problemi di udito possono inficiare il senso di un messaggio vocale e questi sono solo alcuni esempi di possibili ostacoli che soggetti anziani o pazienti con deficit cognitivi possono incontrare.

Barber et al. (64) ha monitorato 256 ospiti, reclutati in 55 case di cura, prendendo una media di 8.0 trattamenti. Per *centosettantotto* (69,5%) degli ospiti si sono riscontrati uno o più errori terapeutici. Un valore medio di 1,9 errori per ospite. Il potenziale danno medio derivante da prescrizione, monitoraggio, somministrazione, ed errori di dispensazione è stato stimato rispettivamente come 2.6, 3.7, 2.1 e 2.0 su una scala da 0 = nessun danno a 10 = morte. Gli autori hanno sottolineato quanto sia, effettivamente, motivo di preoccupazione che due terzi degli ospiti sia stato esposto a uno o più errori terapeutici. Perri et al. (65) ha esaminato 1117 cartelle cliniche di pazienti di 15 case di cura in Georgia. Un totale di 519 (46,5%) pazienti ha subito almeno un trattamento inappropriato, e 143 (12,8%) pazienti hanno manifestato almeno un evento avverso

¹³ In questo contesto è riportato fedelmente il testo dell'articolo "*A Reinforcement Learning and Deep Learning based Intelligent System for the Support of Impaired Patients in Home Treatment*" presentato da Nacem ing. M, Paragliola ing.G. e Coronato prof. A. dell'ICAR di Napoli per il progetto AMICO (61). Si è ritenuto opportuno, pertanto esprimersi in prima persona plurale.

allo stato di salute. Gli autori hanno osservato come un uso inappropriato di farmaci fosse comune nell'ambito del loro studio.

L'uso inappropriato di terapie farmacologiche negli anziani è associato ad un aumento del rischio di aggravamento dello stato di salute.

In generale, si stima che gli errori terapeutici danneggino 1,5 milioni di persone ogni anno negli Stati Uniti (United States Institute of Medicine 2006), uccidano 7.000 pazienti all'anno e rappresentino circa 1 ricovero ospedaliero su 20 in US. E' probabile che l'incidenza sia simile in Europa (66). L'impatto degli errori terapeutici varia a seconda degli studi e dei Paesi. Si stima che la percentuale di pazienti soggetti a errori terapeutici in ospedale è compresa tra il 2% e il 14%, con la conseguenza che circa l'1-2% di loro ne viene danneggiato. Inoltre, la non aderenza è anche un fattore di rischio per una serie di successivi esiti negativi sulla salute, fino addirittura ad annoverarsi a 125.000 decessi ogni anno (67), (68). L'OMS afferma che *"le statistiche mostrano che le strategie per ridurre il tasso di eventi avversi nella sola Unione Europea porterebbero alla prevenzione di più di 750.000 errori medici dannosi per anno, che a loro volta si traducono in 3,2 milioni di giorni di degenza ospedaliera in meno, 260.000 incidenti di invalidità permanente in meno e 95.000 decessi in meno per anno"* (69).

4.2.2 Stato dell'arte

Le tecnologie di Intelligenza Artificiale stanno trasformando e potenziando radicalmente i servizi di assistenza sanitaria. Approcci di Reinforcement Learning e Deep Learning sono utilizzati, al giorno d'oggi, per definire e realizzare servizi e applicazioni sanitarie innovative come, ad esempio, l'imaging medico (Jin et al. del 2017 (70), Abdelaziz Ismael et al. del 2020 (71) e Coronato et al. del 2020 (72)); l'identificazione degli eventi avversi (Paragliola & Coronato del 2018 (73) e Coronato et al. del 2014 (74)); terapie personalizzate come, ad esempio, in Tseng et al. del 2017 (75) e Raghu

et al. del 2017 (76); la gestione del rischio (Paragliola & Naeem del 2019 (77) e Coronato & Cuzzocrea nel 2020 (78)); i servizi di sanità elettronica basati sul linguaggio naturale (Gargiulo et al. nel 2018 e 2019 (79)).

Per quanto riguarda il supporto dei pazienti durante i trattamenti farmacologici a casa sono stati elaborati diversi sistemi di promemoria. In questa sezione, presentiamo lo stato dell'arte ed evidenziamo le principali caratteristiche e differenze tra queste soluzioni e l'approccio da noi proposto.

In Foreman et al. del 2012 (80) gli autori valutano l'aderenza alla terapia per i pazienti che ricevono promemoria testuali. Gli autori hanno definito sei tipi di messaggi. Ai soggetti viene notificata solo una combinazione.

In questa soluzione non esiste alcuna distinzione tra pazienti affetti da menomazioni fisiche o cognitive e pazienti "normali". Non esiste una fase di verifica per accertarsi che l'utente segua il promemoria, oppure no.

In Park et al. del 2014 (81), gli autori esaminano l'efficacia di un intervento di eHealth¹⁴ utilizzando messaggi di testo, per migliorare l'aderenza ad antiaggreganti e statine, tra i pazienti con una storia di infarto miocardico e/o intervento coronarico percutaneo. L'obiettivo principale è quello di confrontare l'aderenza terapeutica tra diversi gruppi di pazienti.

Il sistema proposto definisce diversi tipi di messaggi per ogni gruppo di pazienti. Nella soluzione presentata non esiste una definizione dinamica dei messaggi; infatti, essi sono fissati per ogni gruppo.

In Abdul Minaam & Abd-ELfattah del 2018 (82), gli autori presentano un portapillole *smart* in grado di ricordare al paziente di prendere un medicinale. Il portapillole emette suoni e luci per

¹⁴ Per "eHealth", o "Sanità in Rete" s'intende l'utilizzo di strumenti basati sulle tecnologie dell'informazione e della comunicazione per sostenere e promuovere la prevenzione, la diagnosi, il trattamento e il monitoraggio delle malattie e la gestione della salute e dello stile di vita.

richiamare l'attenzione dell'utente, mentre un messaggio viene visualizzato anche sul suo smartphone. Il sistema proposto non prevede alcuna fase di verifica. Di conseguenza, la certezza che il farmaco sia preso correttamente è a carico del paziente stesso o del suo badante.

In Wu et al. (2017) (83), gli autori propongono un portapillole intelligente potenziato da un sistema di Intelligenza Artificiale che genera promemoria sensibili al contesto. Le informazioni sul contesto vengono acquisite da un sistema multi-sensore inserito in un ambiente smart-home. E' presente un modulo per il riconoscimento delle attività svolte dal paziente. L'approccio proposto è in grado di adattare il messaggio di promemoria in base alla posizione del paziente stesso, mentre non tiene conto delle sue attitudini.

Un altro portapillole intelligente è presentato da Wu et al. nel 2015 (84). Tale soluzione è in grado di tirar fuori dalla scatola un sacchetto contenente il farmaco, all'orario previsto. Se il paziente non estrae il sacchetto, la scatola avvisa il caregiver tramite Skype. Il sistema fornisce una sorta di fase di verifica, ma necessita di una connessione Internet disponibile.

In M. Hayakawa, Y. Uchimura, K. Omae, K. Waki, H. Fujita (87), gli autori presentano un sistema di autogestione dei farmaci, in grado di ricordare e monitorarne l'assunzione in tempo reale. Il sistema utilizza lo smartphone personale del paziente per memorizzare una cronologia di somministrazione, ma non vi è alcun adattamento del servizio a diversi tipi di pazienti; cioè, ancora una volta, viene fornita la notifica a tutti i pazienti allo stesso modo, né viene implementata una fase di verifica.

Patil nel 2019 (88) propone un sistema che si occupa della programmazione della terapia. Per rendere i promemoria facili da usare, uno schermo LCD mostra un timer con l'indicazione della prossima assunzione. Lo stesso tipo di messaggio è consegnato a tutte le classi di pazienti. Nella soluzione proposta non viene fornita alcuna verifica delle ipotesi.

Il problema dell'interazione con gli anziani viene affrontato in Teixeira et al. nel 2016 (89). Gli autori, infatti presentano "*una metodologia centrata sugli anziani*" che è dimostrata, rispetto alla progettazione di un assistente terapeutico. Nel loro caso di studio, tuttavia, adottano messaggi testuali statici insieme ad immagini, per comunicare con il paziente, senza tenere conto delle sue particolari attitudini. Ancora una volta, la fase di verifica non è implementata.

In Tucker et al. del 2015 (90), gli autori hanno proposto un sistema data-driven per allarmi tempestivi circa la sicurezza del paziente. Hanno progettato un sistema di lettura dei dati dell'intero corpo in movimento, in grado di discriminare i pazienti tra soggetti che assumono o non assumono la terapia. L'articolo si concentra sulla verifica della corretta adesione al trattamento farmacologico. Tuttavia, non affronta la problematica di indicare quale farmaco il paziente deve assumere.

In He et al. (2018) (91), gli autori hanno sviluppato un'applicazione per cellulari per migliorare l'aderenza alla terapia con messaggi di testo come promemoria. La soluzione proposta, tuttavia, non si adattata alle attitudini del paziente.

In Buono et al. (2018) (92), gli autori hanno definito un linguaggio formale per la specifica del comportamento di un'infrastruttura di devices IoT¹⁵ utilizzati nel monitoraggio della terapia domestica. L'applicazione non affronta il problema di aderenza alla terapia.

In Beauchemin et al. del 2019 (93), gli autori hanno proposto una Web App per supportare l'adesione a cure mediche per i malati di HIV. L'App può inviare promemoria, ma non può personalizzarli a seconda delle capacità dell'utente.

Al fine di confrontare tali approcci con i nostri sono stati selezionati cinque tipi di funzionalità:

- *Tipo di messaggio*: descrive il modo in cui il sistema interagisce con il paziente (ad es. messaggio di testo, messaggio audio, ecc.)

¹⁵ Internet of Things. Nelle telecomunicazioni è un neologismo riferito all'estensione di Internet al mondo degli oggetti e dei luoghi concreti per cui le "cose" si rendono riconoscibili e acquisiscono intelligenza grazie al fatto di poter comunicare dati su se stessi e accedere ad informazioni aggregate da parte di altri.

- *Messaggio dinamico*: considera se il messaggio è costruito staticamente, o dinamicamente (ad esempio, la notifica può essere personalizzata o è identica per ogni tipo di paziente)
- *Tipo di soggetto*: descrive il tipo di paziente
- *Verifica*: considera se è presente una fase di verifica per controllare l'aderenza alla terapia (il paziente sta prendendo la pillola giusta o no?)
- *Tipo di dispositivi*: descrive i tipi di dispositivi utilizzati per ricordare e avvisare il paziente

La Tabella 1 mostra un riepilogo del nostro confronto.

4.2.2.1 Limiti dello stato dell'arte

Sebbene i lavori correlati, presentati nel precedente paragrafo, forniscano alcune soluzioni di base per l'assistenza alle terapie sanitarie domestiche è possibile evidenziare alcune limitazioni.

- 1) Quasi tutte le soluzioni emettono messaggi statici; cioè i tipi di messaggi sono definiti in fase di progettazione. In particolare, nessuna soluzione prevede una *personalizzazione del messaggio* in base alle attitudini del paziente. Dal nostro punto di vista, questa è una forte limitazione, perché soggetti diversi possono avere differenti capacità visive, uditive e cognitive. Ad esempio, potrebbe essere difficile riconoscere le informazioni da un'immagine per i pazienti che hanno menomazioni visive.
- 2) Non esiste una fase di verifica che confermi che il paziente ingerisca la pillola giusta. I documenti esaminati limitano il loro campo di applicazione al ricordare la pillola, ma non viene eseguita alcuna verifica dell'aderenza alla terapia. In alcune soluzioni è il paziente stesso a notificare l'assunzione del farmaco.

4.2.3 Contributo soluzione proposta

Nel nostro approccio, affrontiamo le limitazioni precedenti come segue:

- 1) Per quanto riguarda la *personalizzazione del messaggio*, un agente software intelligente apprende, in modo dinamico e autonomo, il metodo migliore per comunicare con il paziente. In particolare, un approccio basato sul Reinforcement Learning viene acquisito dall'agente per individuare un modo per comunicare e osservarne l'efficacia. Dopo un certo numero di tentativi, l'agente apprenderà la struttura del messaggio più efficace per comunicare con lo specifico paziente.
- 2) Per fornire una *fase di verifica*, abbiamo definito un altro agente intelligente in grado di riconoscere se il paziente sta maneggiando la scatola della pillola che gli è stata ricordata di prendere, oppure no. Subito dopo aver emesso il promemoria, l'agente inizia ad analizzare le immagini catturate dalla fotocamera e classifica gli oggetti gestiti dal paziente. Il feedback audio e visivo viene generato quando il paziente prende il portapillole giusto.

Tabella 1: Panoramica delle opere correlate

Riferimento	Tipo di messaggio	Messaggio Dinamico	Tipo di soggetto	Verifica	Tipo di dispositivo
Foreman et al.(2012) (80)	Testo	Statico	Normale	NO	Smartphone
Park et al. (2014) (81)	Testo	Diversi tipi di messaggi di testo statici	Normale	NO	Smartphone
Abdul Minaam & Abd-ELfattahm (2018) (82)	Audio e immagini	Statici	Soggetti con malattie cardiache	NO	Smart Dispenser
Wu et al (2017) (83)	Audio e video	Dinamici-Basatii sull'ambiente	Normale	SI	Ambiente Intelligente, IoT
Wu et al (2015) (84)	Fisico (estrazione della scatola dei medicinali)	Statico	Normale	SI	Portapillole Smart
Ramljak (2017) (85)	Testo tramite cellulare e Audio+Video tramite strutture per smart-home	Statiche	Normale	SI	Smartphone
Nijiya Jabin Najeeb et al. (2018) (86)	Testo	Statico	Normale	NO	Portapillole intelligente
M.Hayakawa,Y. Uchimura, K. Omae,	Testo e audio	Statico	Normale	NO	Portapillole intelligente

K. Waki, H.Fujita (87)					
Patil (2019) (88)	Testo	Statico	Normale	NO	Portapillole Smart
Teixeira et al.(2016) (89)	Immagine e Testo	Statico	Normale	NO	Smartphone
Tucker et al.(2015) (90)	Testo	Statico	Malati di Parkinson	SI	App autonoma
He et al. (2018) (91)	Testo	Statico	Normale	NO	Mobile App
Buono et al.(2018) (92)	Testo	Statico	Normale	NO	Mobile App
Beauchemin et al.(2019) (93)	Testo	Feedback su misura	Malati di HIV	NO	Web App
Nostra Soluzione nell'ambito del progetto AMICO	Testo, Audio, Immagini (più qualsiasi combinazione di questi canali)	Dinamica – Basata sulle attitudini del paziente	Normale, Lieve cognitivo compromesso, Visivo/Uditivo compromesso	SI	Smartphone e/o Robot e/o Web

4.2.4 Background tecnico

Prima di entrare nei dettagli della soluzione proposta, è necessario richiamare parte della teoria relativa agli approcci di Reinforcement Learning (94)

Un problema di Reinforcement Learning (RL) è descritto formalmente per mezzo di un processo decisionale di Markov (MDP).

Definizione 1. Un Processo Decisionale di Markov (MDP) è una tupla $\langle S, A, R, P, \gamma \rangle$; dove S è un insieme (finito) di stati; A è un insieme (finito) di azioni; R è una funzione di ricompensa che modella l'obiettivo dell'agente RL; P è una probabilità di transizione di stato; e γ è un fattore di sconto (discount): $\gamma \in [0, 1]$.

Si assume la proprietà di Markov. Essa afferma che il futuro (prossimo stato) dipende solo dallo stato attuale e non da quelli precedenti.

Un MDP è specificato dall'elenco delle sue azioni, dall'elenco dei suoi stati e dalle dinamiche dell'ambiente. Per ogni coppia stato-azione (s, a) , la probabilità di ogni coppia stato successivo-ricompensa (s', r) , è definita come:

$$p(s', r | s, a) \doteq \Pr\{S_{t+1} = s', R_{t+1} = r \mid S_t = s, A_t = a\}$$

Informalmente, l'obiettivo dell'agente è massimizzare la ricompensa (premio) ottenibile nel tempo. Pertanto, indicando la sequenza delle ricompense come R_{t+1}, R_{t+2}, \dots dopo il passo temporale t , l'obiettivo è massimizzare la funzione $G_t = R_{t+1} + R_{t+2} + \dots + R_T$ in cui T è l'ultimo passo temporale.

Definizione 2. G_t è la somma delle ricompense scontate raccolti al tempo t .

$$G_t = \sum_{k=0}^T \gamma^k R_{t+k+1}$$

In cui $0 \leq \gamma \leq 1$ è il *discount rate*. Ciò consente un trade-off tra due esigenze, il tentativo di ottenere la massima ricompensa cumulativa e il desiderio di ottenere un beneficio sufficiente entro un tempo ragionevole.

Definizione 3. Una policy π è una distribuzione di probabilità sulle azioni dati gli stati.

$$\pi(a|s) \doteq P[A_t = a | S_t = s]$$

Data la policy π e il valore di ritorno G_t , è possibile specificare due *value functions* relative al rendimento atteso: la funzione stato-valore e la funzione azione-valore.

Definizione 4. La *state-value function* $v_\pi(s)$ è il ritorno atteso partendo dallo stato s e seguendo π .

$$v_\pi(s) \doteq E_\pi [G_t | S_t = s] = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s \right]$$

Allo stesso modo, la funzione azione-valore $q_\pi(s, a)$ di un MDP è il ritorno atteso dallo stato s , selezionando l'azione a , e seguendo π .

Su diverse value functions è possibile definire quelle ottimali; cioè, quelle che ottengono le migliori prestazioni in termini di ritorno.

Definizione 5. La *funzione stato-valore ottimale* $v_*(s)$ è la *value function* massima su tutte le policies.

$$v_*(s) = \max_{\pi} v_\pi(s), \forall s \in S$$

Dopo aver definito lo MDP, il passo successivo è ottenere una policy ottima. RL è un'ampia classe di algoritmi per la risoluzione di MDP. I metodi RL possono essere classificati come model-based vs. model-free e value-function vs. metodi policy-based.

Il seguente lavoro è basato sul RL Bayesiano. Nel Bayesian RL, un'inferenza su una Variabile Random (RV) X viene creata utilizzando una distribuzione di probabilità per RV X (95). Quindi, le inferenze possono essere estratte dalla distribuzione di probabilità. Ad esempio, si consideri che RV X sia nascosta e si possano fare osservazioni solo su una RV Y correlata. Il compito dell'apprendimento bayesiano è dedurre X dai campioni di RV Y ; ad esempio, quando X è una quantità fisica e Y è la misura del rumore. Un processo di inferenza bayesiana può essere realizzato come segue:

1. Si considera la distribuzione a-priori $P(X)$ che rappresenta la conoscenza su RV X senza l'osservazione dei dati;
2. Si sceglie un modello statistico $P(Y|X)$ che esprima la conoscenza su RV Y quando X è data. Anche $P(Y|X)$ riflette la dipendenza statistica tra RVs X e Y ;
3. Si osserva il dato $Y = y$;
4. Si calcola la distribuzione a-posteriori per aggiornare la conoscenza su RV X utilizzando la regola di Bayes.

$$P(X|Y = y) = \frac{P(y|X)P(X)}{\int P(y|X')P(X')dX'}$$

4.2.5 Architetture del sistema e degli Agenti Software

4.2.5.1 Panoramica delle Architetture

L'architettura del sistema, come mostrato nella Figura 48 è composta da tre agenti software intelligenti (ovvero il *Planner*, il *Tutor* e il *Checker*), che collaborano nel ricordare la terapia al paziente e nel prevenire errori terapeutici.

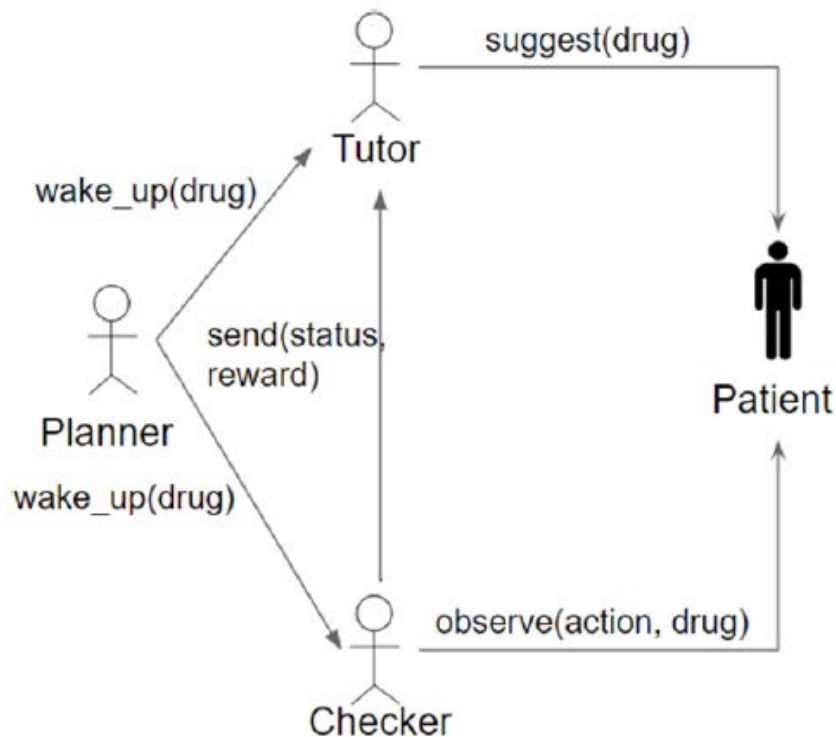


Figura 48: Schema a blocchi del sistema

- *Funzionalità e definizione del Planner*: Il *Planner* è un agente incaricato di schedulare la terapia per il paziente. È stato implementato come thread autonomo, che invia notifiche sincronizzate sia al *Tutor* che al *Checker*, secondo il piano terapeutico del paziente, al fine di attivare questo secondo agente.

- *Funzionalità e training del tutor*: il *Tutor* è incaricato di inviare un messaggio appropriato all'utente. La caratteristica, più rilevante, di questo agente è di essere in grado di apprendere il tipo di messaggio più efficace, in funzione delle attitudini del paziente. Considerando che il tutor impara dal profilo di quest'ultimo, abbiamo adottato una classe di metodi RL model-based per addestrare l'agente. La policy di apprendimento on-line garantisce un approccio di learning più conservativo. In particolare, se c'è il rischio di una grande ricompensa negativa associata ad una possibile policy, l'agente tende ad evitare tale policy in vista di una più sicura. Il tutor apprende il grado di abilità del paziente (in termini di audio, capacità visive e cognitive) attraverso dei tentativi. In fine, comunica al paziente per mezzo del tipo di messaggio che risulta essere il modo più appropriato in base alle attitudini del paziente stesso. Abbiamo condotto alcuni studi

preliminari per realizzare un agente software in grado di apprendere capacità cognitive (96) e capacità uditive e visive (97). Nel nostro precedente lavoro, abbiamo adottato approcci classici di Reinforcement Learning, mentre in questo documento abbiamo integrato un Bayesian Reinforcement Learning, che ha dimostrato di raggiungere prestazioni superiori. Il tutor si espone come thread autonomo che riceve come input la notifica del medicinale da suggerire al *Planner* e stabilisce come inviare un messaggio personalizzato al paziente combinando i tipi di base di messaggi riportati nella Tabella 2. Maggiori dettagli sulla definizione di tutor sono riportati nella sezione Tutor.

- *Funzionalità e training del Checker*: il Checker è incaricato di verificare che il paziente - dopo un sollecito - prenda la pillola corretta. Utilizza una telecamera per monitorare le azioni dei pazienti durante la manipolazione del portapillole. La fase di training di questo

Tabella 2: Tipi di messaggi di base

No	Code	Message Type
1	C1	Messaggio Audio Scientifico
2	C2	Messaggio Audio Semplice
3	C3	Immagine Visiva del portapillole
4	C4	Messaggio di testo Scientifico
5	C5	Messaggio di testo Semplice

agente mira proprio a renderlo capace di riconoscere la scatola. Il training viene svolto offline rispetto agli altri agenti a causa del lungo tempo di formazione. L'agente è progettato come classificatore multi-classe per il quale abbiamo adottato una rete VGG-16, principalmente, per risparmiare tempo per il training. Il VGG-16 è una ben nota rete pre-addestrata con un buon trade-off, tra tempo di formazione e complessità, rispetto ad altre architetture come AlexNet o GoogleNet. Il checker si espone come thread attivato dopo che il tutor ha inviato un messaggio al paziente. Questo agente prende le immagini del portapillole e fornisce l'etichetta prevista della scatola. Maggiori dettagli sulla formulazione del checker sono nella sezione Checker.

Per il resto del documento, ci concentreremo sul Tutor e sul Checker, in quanto il Planner è un componente software che non contiene particolari meccanismi intelligenti.

4.2.5.2 Tutor

Il Tutor è un agente basato sull'apprendimento per rinforzo che interagisce con il paziente. È stato definito un modello del paziente tenendo in considerazione le seguenti quattro attitudini codificate tramite cinque parametri che possono variare nel range [0- > 1]:

- Percezione Uditiva: questa attitudine è rappresentata dal parametro *Audio Skill* (AU_s). Un valore basso di questo parametro implica che la probabilità che il paziente sia in grado di ascoltare un messaggio audio è bassa. Di conseguenza, un messaggio audio (ad esempio, un promemoria audio) probabilmente non verrà afferrato dal paziente.
- Percezione Visiva: questa capacità è rappresentata dal parametro *Visual Skill* (VS_s). In questo caso, minore è la capacità visiva, minore è la probabilità di visualizzare un messaggio visivo (ad esempio, un'immagine di una scatola di un farmaco o un messaggio di testo sul display).
- Working Memory: questa attitudine è codificata per mezzo di due parametri, il *Working Memory High Skill* (WM_{hs}) e *Working Memory Low Skill* (WM_{ls}). Minore è il WM_{hs} , minore è la probabilità di comprendere un messaggio "complesso". Ad esempio, se un'alta capacità di memoria di lavoro¹⁶, WM_{hs} , di un paziente è in buone condizioni, allora questo può capire i messaggi scientifici come l'intero nome scientifico del farmaco. Invece, maggiore è la WM_{ls} e maggiore è la probabilità di comprendere un messaggio "semplice" audio o visivo. In alcuni casi, un paziente potrebbe non essere in grado di riconoscere il nome scientifico complesso del farmaco, ma può capire un messaggio semplificato come, ad esempio, "*prendi la pillola*

¹⁶ In psicologia la memoria di lavoro è la parte della memoria a breve termine che si occupa di elaborazione percettiva e linguistica cosciente immediata. In sostanza è l'abilità di mantenere temporaneamente informazioni in mente e manipolarle.

bianca dalla scatola gialla". Una corretta combinazione di questi due parametri può emulare situazioni diverse.

- **Attenzione:** questa attitudine è rappresentata dal parametro *Attention Skill* (AT_s). Minore è la capacità, maggiore è la probabilità di ignorare il messaggio. L'attenzione del paziente è fondamentale per il completamento con successo del compito. In effetti, un paziente potrebbe essere in grado di riconoscere messaggi audio e / o visivi, ma non di mantenere il livello di attenzione richiesto finché l'attività non sia terminata, causandone il fallimento.

Abbiamo definito cinque tipi di messaggi di base come si può vedere nella Tabella 2 **Errore**. **L'origine riferimento non è stata trovata.** Il Tutor può combinare questi tipi di base per trovare la struttura del messaggio più efficace per il paziente. In altre parole, se per un paziente il modo più efficace per ricordare una pillola è, ad esempio, un messaggio audio che riporta il nome scientifico del farmaco (tipo C1); per un altro il modo migliore potrebbe essere la combina dei tipi C1C3C5 (cioè il nome scientifico riprodotto dallo speaker audio insieme all'immagine della scatola del farmaco e un messaggio di testo descrittivo mostrato da un display).

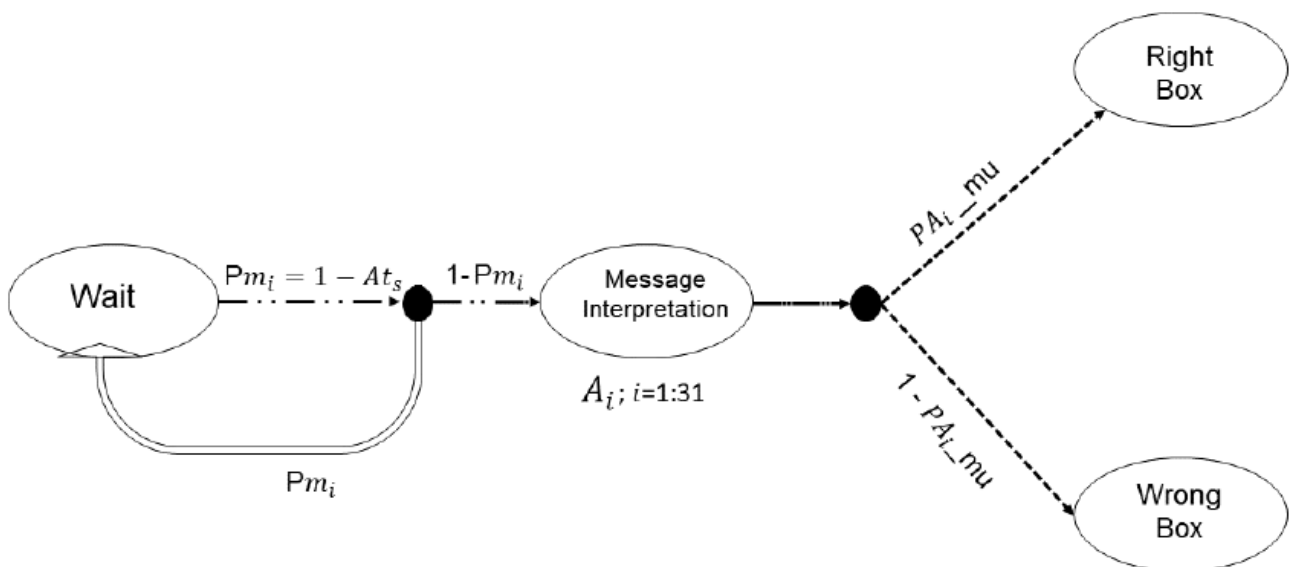


Figura 49: Il processo decisionale di Markov che descrive il comportamento del paziente

Per quanto riguarda il MDP riportato in Figura 49 esso descrive il comportamento del paziente in termini di stati e transizioni di stato. Tali transizioni dipendono dall'azione eseguita dal Tutor (A_i) e due distribuzioni di probabilità: la probabilità che il messaggio sia compreso dal paziente (P_{mu}) e la probabilità che il messaggio venga ignorato dal paziente (P_{mi}).

Ogni azione (A_i) rappresenta un promemoria basato su una specifica combinazione di tipi di messaggio base che il Tutor può creare e fornire dinamicamente al paziente. Dopo aver ricevuto il messaggio, il paziente può ignorarlo o può provare ad interpretarlo.

L'equazione

$$P_{mi} = f(AT_s)$$

indica che la probabilità che un messaggio venga ignorato dipende dalla sua Attention Skill. D'altra parte, la probabilità che un messaggio sia compreso correttamente dal paziente dipende dalle altre abilità e quindi:

$$P_{mu} = f(AU_s, VS_s, WM_{hs}, WM_{ls})$$

Le due equazioni precedenti sono formalizzate dalle due successive:

$$P_{mi} = 1 - AT_s$$

$$P_{mu} = \min (1, C1 * AU_s * WM_{hs} + C2 * AU_s * WM_{ls} + C3 * VS_s * WM_{hs} + C4 * VS_s * WM_{ls} + C5 * VS_s * WM_{hs} + C6 * VS_s * WM_{ls})$$

L'obiettivo dell'agente è massimizzare il completamento, con successo, del compito del promemoria; cioè, il paziente prende la pillola giusta. Per raggiungere questo obiettivo, abbiamo implementato un metodo bayesiano di apprendimento per rinforzo.

Gli algoritmi bayesiani per RL sono stati ampiamente studiati, producendo approcci per incorporare la conoscenza precedente nei modelli di inferenza. In parole semplici, adottando un metodo RL bayesiano è possibile far partire l'agente da una conoscenza iniziale pregressa invece che da zero. Di conseguenza, la fase di training può trovare la policy ottimale (e perfezionare il modello) in un minore numero di interazioni rispetto ai metodi classici RL.

L'equazione che dobbiamo risolvere per il Bayesian RL è:

$$V^{\pi^*}(x, b) = \max_a \sum_{x'} \Pr(x'|x, b, a) [X'_r + \gamma V^{\pi^*}(x', b_{xax'})]$$

Dove X rappresenta l'insieme di stati (nel caso del nostro MDP: *Wait*, *Message interpretation*, *Right box* e *Wrong box*), b rappresenta una distribuzione (o conoscenza) rispetto a θ non conosciuto che viene utilizzato per l'exploration e $\Pr(x'|x, a, b)$ rappresenta la probabilità di transizione.

Ogni volta che il Tutor comunica un promemoria, il paziente può prendere la scatola giusta o sbagliata a seconda delle probabilità P_{mu} e P_{mi} . Queste probabilità possono essere modellate come distribuzione a-posteriori $P(\theta|x)$ utilizzando θ basato su informazioni o osservazioni iniziali utilizzando una regola di Bayes come segue:

$$P(\theta|x) = \frac{P(x|\theta)P(\theta)}{P(x)}$$

La funzione di verosimiglianza $P(x|\theta)$ segue la distribuzione di Bernoulli; mentre apriori $P(\theta)$ segue la distribuzione Beta

Nel nostro caso, usiamo come probabilità iniziali P_{mi} e P_{mu} - indicate nelle rispettive equazioni - e applichiamo la regola di Bayes per modellare la distribuzione a-posteriori $P(\theta|x)$. Una soluzione analoga è stata esposta da Sutton per il problema dei banditi multi-armati (98)

Nel nostro problema, poiché $P(\theta)$ è Beta distribuita e $P(x|\theta)$ è Bernoulli distribuita, anche il termine $P(\theta|x)$ è Beta distribuita, il che significa che quando un paziente raggiunge lo stato *Right box*, la distribuzione a-posteriori ($P(\theta|x)$) diventerà $\text{Beta}(\alpha+1, \beta)$. Al contrario, se il paziente raggiunge lo stato *Wrong box*, l'a-posteriore sarà $\text{Beta}(\alpha, \beta+1)$.

Usiamo l'algoritmo Thompson Sampling (TS) (99) per risolvere il problema exploration-exploitation. TS è un metodo per le attività decisionali online dove le azioni sono selezionate sequenzialmente in un modo da bilanciare l'esperienza (exploiting) - ciò che è già noto per massimizzare la ricompensa immediata - e l'esplorazione (exploration) - il modo in cui l'agente cerca nuove possibilità per aumentare la ricompensa totale futura.

In TS, per ogni messaggio, la probabilità $P(\theta|x)$ viene campionata dall'inizio e, successivamente viene selezionato dal Tutor, come promemoria per il paziente, il messaggio con la più alta probabilità campionata.

4.2.5.3 Checker

L'agente Checker è definito come un classificatore di immagini che è stato realizzato adottando un'architettura di rete ben conosciuta: la rete pre-addestrata VGG-16 (100).

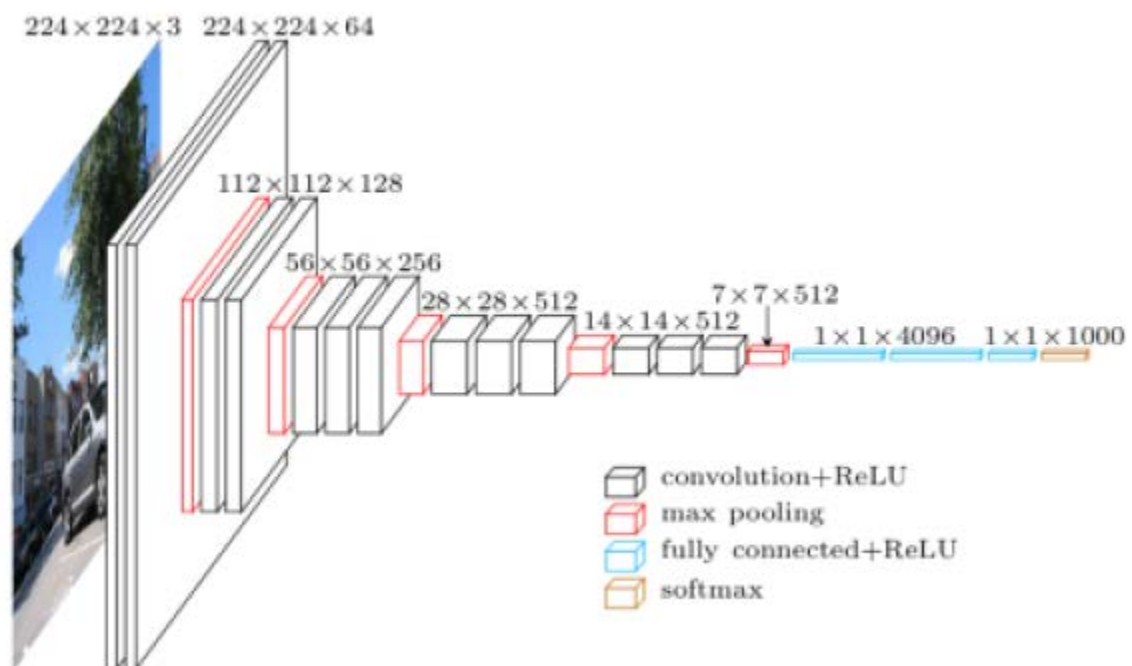


Figura 50: Modello originale VGG-16 copiato da Pyimagesearch.com.

La Figura 50 mostra una panoramica della rete, costituita da 13 strati convoluzionali e 3 livelli completamente connessi, tutti i livelli sono stati addestrati a-priori. Per il training del nostro modello, abbiamo congelato gli strati di convoluzione del modello VGG-16 originale.

Per ottenere un buon training, abbiamo creato un dataset di immagini di scatole di farmaci. Per gli scopi di questo documento, abbiamo scelto dodici farmaci ampiamente utilizzati in Italia. Questi dodici farmaci sono scelti, casualmente, al solo scopo di test. Per ogni scatola di medicinali raccogliamo una grande

varietà di immagini scattate sotto differenti condizioni di luce, con diverse angolazioni e sfondi da entrambi i lati anteriore e posteriore.

Successivamente, abbiamo applicato tecniche di data augmentation per aumentare il numero di immagini per ogni scatola medica, ottenendo 500 immagini per ogni medicinale. In totale, abbiamo organizzato un set di dati di 6000 immagini.



Figura 51: Alcune immagini del farmaco "Tachipirina" dopo l'aumento.

Alcune immagini di un medicinale dopo l'aumento dei dati sono mostrate nella Figura 51.

Su 6000 immagini, 4500 immagini sono utilizzate per l'addestramento e le restanti 1500 per il test e quindi è stato adottato l'80% dei nostri campioni di dataset per il training e il restante 20% per i test.

Abbiamo normalizzato tutti i dati nelle dimensioni di $224 \times 224 \times 3$.

Per gli esperimenti, abbiamo adottato due versioni della rete VGG-16. Nella prima versione, abbiamo congelato tutti gli strati di convoluzione dell'originale VGG-16 e riaddestrato gli ultimi due livelli fully connected impostando 1024 neuroni per livello invece di 4096. Abbiamo chiamato questa configurazione *model-1*.

Per migliorare l'accuracy, abbiamo deciso di riaddestrare gli ultimi quattro strati convoluzioni della rete originale (l'ultimo 28x28x512 e i tre livelli 14x14x512) in quanto il nostro dataset personalizzato è diverso dall'originale su cui è stata addestrata la rete originaria VGG-16. Abbiamo chiamato questa configurazione *model-2*.

Inoltre, per evitare overfitting, abbiamo aggiunto uno strato di dropout con un drop rate di 0,5 tra livelli fully connected e sostituito 1000 classi con 12 classificate per il nostro caso.

Abbiamo utilizzato come loss function nel nostro modello la categorical cross-entropy. Per l'ottimizzazione è stato richiesto un tasso di apprendimento di 0,0001 e una quantità di moto di 0,9 alla discesa del gradiente stocastico.

4.2.6 Configurazione e risultati sperimentali

Per addestrare il Tutor, non abbiamo fatto interagire l'agente artificiale direttamente con il paziente.

Invece, abbiamo acquisito preliminarmente l'iniziale profilo del paziente tra più di cento persone fra i medici coinvolti nel Progetto AMICO.

I medici hanno stimato, approssimativamente, per ognuno di essi, il valore dei cinque parametri che nel nostro modello descrivono le attitudini del paziente. È importante chiarire che tali stime rappresentano la conoscenza iniziale fornita all'agente RL, in accordo con l'approccio Bayesian Reinforcement Learning.

Il Tutor, invece, dopo aver elaborato il profilo iniziale continuerà ad apprendere e perfezionare il suo modello sulla base delle effettive interazioni (ricorda / avvisa) con il paziente.

È anche importante chiarire che una tale exploitation del profilo iniziale è opzionale. In effetti, un classico agente RL imparerebbe da zero interagendo direttamente con il paziente, ma, in questo modo, il Tutor apprende la policy ottima (ovvero il modo migliore per comunicare con il paziente) in un numero decisamente più breve di interazioni.

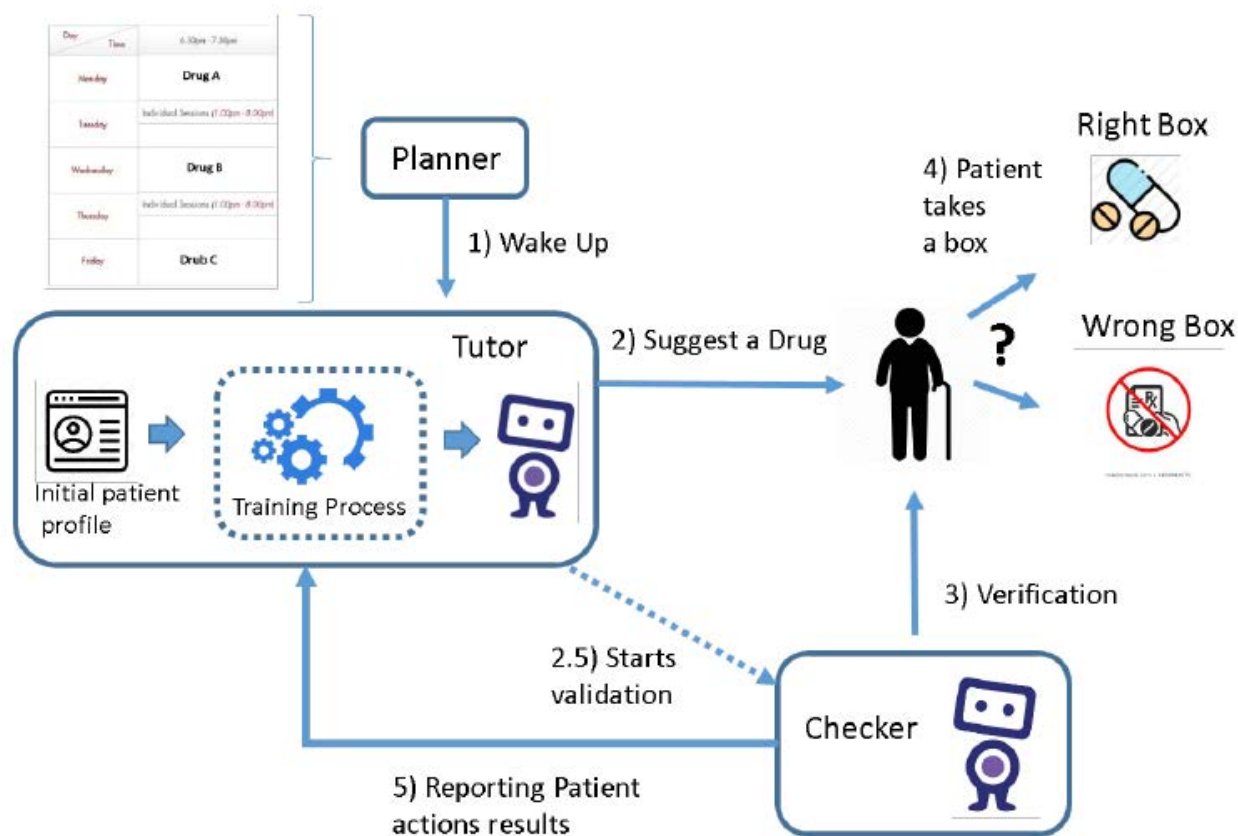


Figura 52: Esecuzione del processo di promemoria

La Figura 52 mostra il flusso di lavoro eseguito per impostare un processo di promemoria per uno specifico paziente.

Successivamente all'addestramento, con o senza un profilo iniziale del paziente, il processo consisterà nelle seguenti fasi ripetute ogni volta che il paziente dovrà prendere una pillola:

- 1) Wake Up:** quando è il momento per il paziente di assumere la terapia, il Planner attiva il Tutor per notificare una pillola specifica.
- 2) Suggest a Drug:** il Tutor invia un promemoria al paziente. Se il sistema è ancora in "modalità allenamento", il Tutor attenderà una ricompensa fornita dal Checker. In base alla ricompensa ricevuta, il Tutor raffinerà il suo modello.

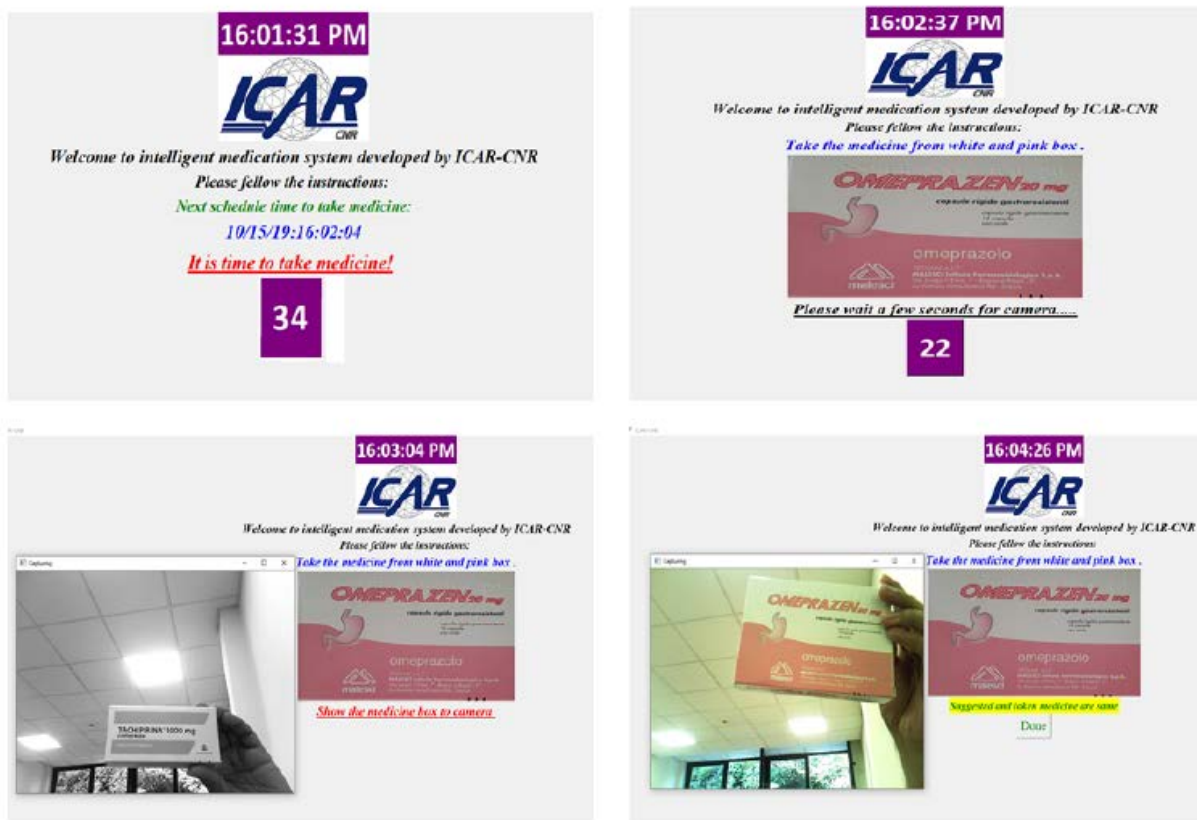


Figura 53 : Dimostrazione dal vivo

- 3) *Starts verification*: il Planner avvisa anche il Checker del farmaco appena ricordato.
- 4) *Patient takes a box*: Il paziente prende una scatola.
- 5) *Verification*: una telecamera cattura l'immagine della scatola del medicinale per verificare se il paziente ha preso la scatola della pillola giusta. Dopodiché, fornisce un feedback al paziente (avviso audio / video, immagine in bianco&nero o a colori, applausi, ecc.) a seconda della situazione.
- 6) *Reporting*: Il Checker annota il risultato dopo il completamento del compito.

La Figura 53 mostra il sistema di interfacce (abbastanza approssimative in questa versione prototipale) lungo il processo di promemoria.

La riga in alto è relativa al Tutor. L'immagine più a sinistra nella riga superiore mostra l'interfaccia relativa al momento in cui ricordare una pillola. In particolare, questa è la situazione un minuto prima dell'orario previsto per la terapia. Il sistema suona e inizia il conto alla rovescia. Dopodiché, il sistema

emette il promemoria (immagine più a destra). In questo esempio, il messaggio fornito al paziente è una composizione C1C3 (messaggio audio scientifico e immagine della scatola di pillole).

La riga inferiore è relativa alle attività che coinvolgono il Checker. La fotocamera è, infatti, attivata.

Nell'immagine più a sinistra la fotocamera cattura un'immagine con la scatola della pillola sbagliata.

L>alert per il paziente viene generato come uno specifico audio-suono. Inoltre, il colore di sfondo dell'immagine della telecamera è in bianco e nero.

Nell'immagine più a destra, invece, la fotocamera cattura l'immagine del farmaco corretto.

Di conseguenza, un applauso è riprodotto dal riproduttore audio e lo sfondo della scena catturata dalla telecamera diventa a colori.

4.2.6.1 Performance del Tutor

Al fine di mostrare l'efficacia del metodo proposto tramite risultati numerici, abbiamo confrontato l'algoritmo Thompson sampling con i metodi Boltzman sampling, Epsilon-Greedy, Random, Softmax, e Upper Confidence Bound (UCB).

Per prima cosa, presentiamo la performance del tutor in merito al suggerimento di un messaggio di promemoria a diversi pazienti virtuali.

Successivamente, quantifichiamo le prestazioni dell'algoritmo di Thompson sampling, utilizzato per il training del tutor, attraverso metriche come il learning rate e il regret.

Alla fine, valutiamo quale algoritmo raggiunge il miglior trade-off tra exploration e exploitation.

È importante notare che per questo setup sperimentale abbiamo adottato emulatori virtuali dei pazienti.

Pertanto, ci aspettiamo che nella configurazione della convalida clinica l'algoritmo avrà bisogno di alcune interazioni in più per convergere all'ottimo.

Abbiamo considerato diversi pazienti virtuali con capacità fisiche variabili e verificato che il tutor (agente RL) sia in grado di scegliere un messaggio adatto in funzione di determinate attitudini. La Tabella 3 mostra la scelta dei messaggi rispetto a diversi pazienti virtuali con attitudini diverse. La scelta dei ranges è puramente dimostrativa per la convalida dell'approccio.

Tabella 3: Prestazioni di tutor verso diversi pazienti virtuali

AU_s	VS_s	WM_{hs}	WM_{ls}	AT_s	Combinazione selezionata da Tutor
0.56	0	0.56	0.44	0.56	C_1C_2
0.64	0.24	0.64	0.36	0.64	$C_1C_2C_3C_4$
0.8	0.48	0.8	0.2	0.8	C_1C_3
1	1	1	0	1	3 C_3
0.08	0.56	0.44	0.56	0.56	$C_1C_3C_4C_5$
0.32	0.64	0.36	0.64	0.64	$C_3C_4C_5$
0.4	0.72	0.28	0.72	0.72	C_4C_5
0.64	0.56	0.44	0.56	0.56	$C_1C_2C_3C_4$
0.88	0.96	0.04	0.96	0.96	C_2C_4

Nella metà superiore della Tabella 3, consideriamo i pazienti con capacità visive variabili mantenendo le altre attitudini allo stesso livello. Nel seguito riportiamo la descrizione dei risultati della Tabella 3:

1. Quando la capacità visiva del paziente è al livello più basso ($VS_s = 0$) e le altre attitudini sono a livello intermedio, la scelta del tutor è C_1C_2 (cioè la combinazione di messaggio audio semplice e scientifico).

Vale la pena notare che il tutor non seleziona alcun messaggio visivo perché la capacità visiva del paziente è molto debole e inoltre non seleziona solo messaggi scientifici perché la working memory è a livello intermedio; in caso di $VS_s = 0.24$, la scelta del tutor per il messaggio è $C_1C_2C_3C_4$ (cioè la combinazione di quattro messaggi di base) in accordo al fatto che la capacità visiva non è debole, ma comunque inferiore alla media e le altre attitudini sono al di sopra della media, ad eccezione della memoria di lavoro;

2. In caso di $VS_s = 0.48$, la scelta del tutor per il messaggio è C_1C_3 (cioè, la combinazione di un messaggio audio scientifico e di un'immagine della scatola del medicinale) poiché la capacità visiva è al livello intermedio e le altre attitudini sono a livelli superiori alla media, eccettuata la working memory low skill;

3. Quando tutte le capacità del paziente virtuale sono al massimo livello tranne che la working memory che è al livello più basso, il tutor seleziona un singolo messaggio di tipo C3 che è l'immagine della scatola del farmaco.

Inoltre, la metà inferiore della Tabella 3 mostra esempi per pazienti con variabile capacità uditiva.

In tutti gli esempi, il tipo di promemoria selezionato è la combinazione di più di un messaggio. Ciò è dovuto al fatto che in uno qualsiasi di questi tre esempi, nessuna delle abilità audio o visive è al livello massimo. Possiamo concludere che il tutor seleziona un solo messaggio della Tabella 2, quando ogni capacità audio o visiva è al massimo livello.

Il processo di apprendimento di un agente RL è il fattore chiave nella evaluation delle sue prestazioni.

L'algoritmo Bayesiano RL che abbiamo adottato (ovvero Thompson sampling) richiede solo poche interazioni con il paziente per apprendere in modo ottimale, come si può vedere nella Figura 54.

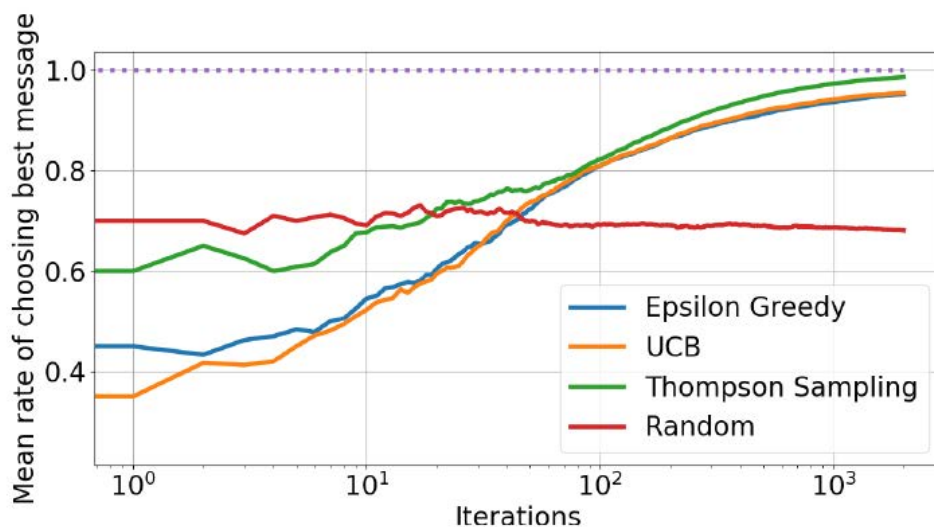


Figura 54: Apprendimento degli agenti per selezionare le migliori azioni

La figura mostra che l'algoritmo Thompson Sampling converge, relativamente rapidamente, nella scelta della migliore azione (messaggio). Entro circa 60 prove, il tasso medio di selezione del messaggio migliore è di circa l'80%. Questo è rilevante rispetto ai metodi UCB, Epsilon-greedy e metodi casuali.

Un'altra metrica per la valutazione delle prestazioni di un algoritmo di RL è il *Regret* (101) - (102). Il regret è la differenza tra la migliore azione possibile e quella selezionata. Più basso è il valore di regret, migliori sono le prestazioni di un algoritmo.

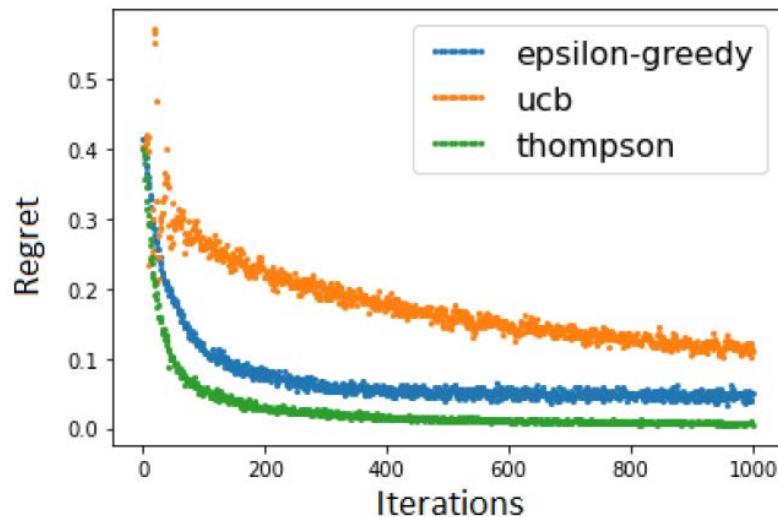


Figura 55: Tasso di convergenza del regret di ogni algoritmo

Possiamo vedere nella Figura 55 che l'algoritmo di Thompson sampling converge verso il minimo regret molto meglio rispetto agli algoritmi ϵ -Greedy e UCB. Il regret si riduce al minimo e converge meglio con l'algoritmo Thompson sampling.

Un equilibrio tra exploration ed exploitation è importante in un algoritmo RL per ottenere buone prestazioni. Poiché abbiamo un totale di 31 azioni (come la combinazione di tutti i messaggi base), l'agente RL deve esplorare il più possibile lo spazio d'azione per accumulare ricompense complessive e allo stesso tempo sfruttare le conoscenze già disponibili.

Per misurare il trade-off tra exploitation ed exploration abbiamo misurato il grado di exploitation utilizzando la Ricompensa Cumulativa Media (ACR) e quello di exploration utilizzando il Root Mean Square Error (RMSE). Quando entrambi i valori ACR e RMSE sono bassi, l'agente RL è in modalità exploration, al contrario, quando ACR e RMSE sono alti, l'agente RL è in modalità exploitation.

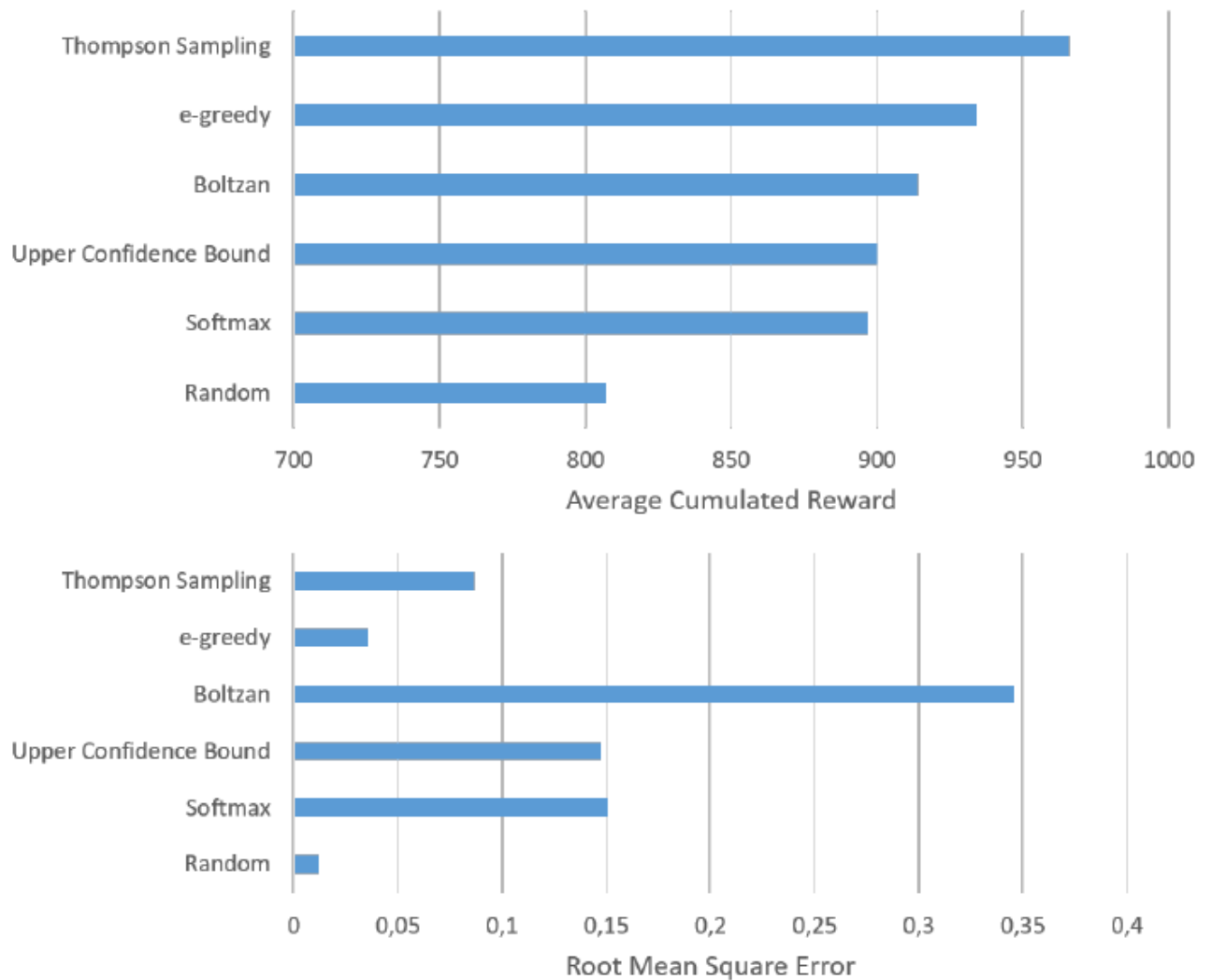


Figura 56: Valori ACR e RMSE di diversi algoritmi

La Figura 56 mostra i valori ACR e RMSE tra la probabilità iniziale di distribuzione e la probabilità stimata.

Il valore RMSE più basso dell'agente Random nella Figura 56 indica uno sbilanciamento dell'algoritmo Random in direzione dell'exploration rispetto all'exploitation.

Ogni volta l'algoritmo ϵ -Greedy sceglie l'azione più gratificante con probabilità $p = 1 - \epsilon$ per la exploitation, mentre un'azione casuale è selezionata con una probabilità $q = \epsilon$ per l'exploration. Per questo motivo, il valore ACR dell'algoritmo ϵ -Greedy è superiore a quello del Random ed agli agenti greedy e presenta un valore RMSE basso a causa della sua exploration casuale.

Il metodo UCB mostra prestazioni simili all'algorithmo ϵ -Greedy in termini di un trade-off tra exploration ed exploitation.

I valori ACR per gli algoritmi Softmax e Boltzman sono inferiori all' ϵ -Greedy e all'UCB. Tuttavia, mostrano valori RMSE più elevati.

Il Thompson sampling ha valori ACR più elevati tra tutti gli algoritmi e allo stesso tempo bassi valori RMSE e, di conseguenza sembra essere, comparativamente il miglior metodo per bilanciare exploitation ed exploration.

4.2.6.2 Performance del Checker

Il modello viene valutato sul validation dataset durante il training e il modello addestrato viene testato sull'applicazione GUI sviluppata. Sono adoperate l'accuracy della classificazione e la loss function come metrica delle prestazioni. La

Figura 57 e Figura 58 mostrano rispettivamente accuracy e loss function per le configurazioni model-1 e model-2.

Model-2 ha prestazioni migliori in quanto è stata raggiunta un'accuracy del 98.00% sui dati di convalida rispetto al model-1 che raggiunge un'accuracy del 90.67%. Inoltre, model-1 presenta una loss di 0.3575 alla convalida dei dati mentre model-2 ha una loss di 0.0583 di molto inferiore a quella del model-1.

Per quanto riguarda il tempo di convergenza del checker, inizialmente, abbiamo addestrato il nostro modello su CPU e ciò ha richiesto circa 29 ore. Nei seguenti passaggi abbiamo addestrato

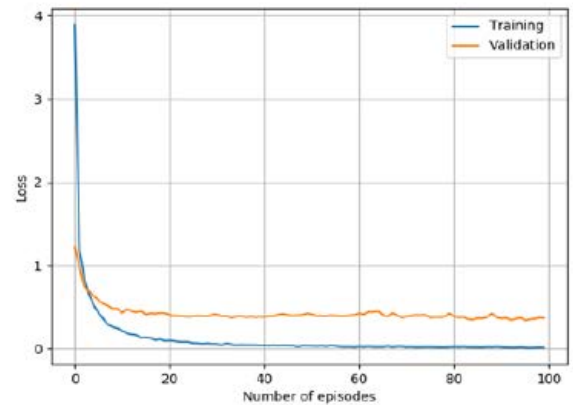
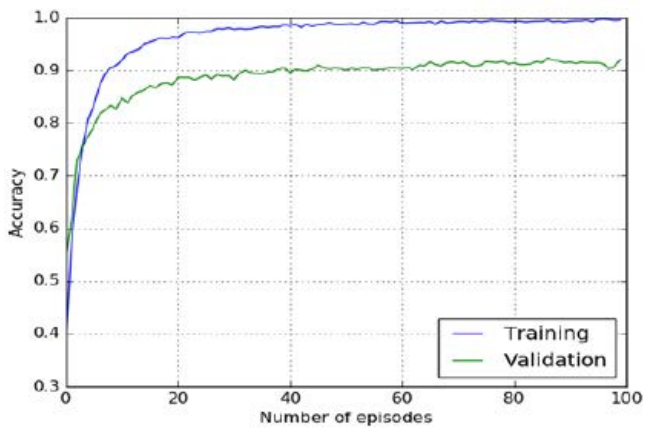


Figura 57: Confronto fra loss

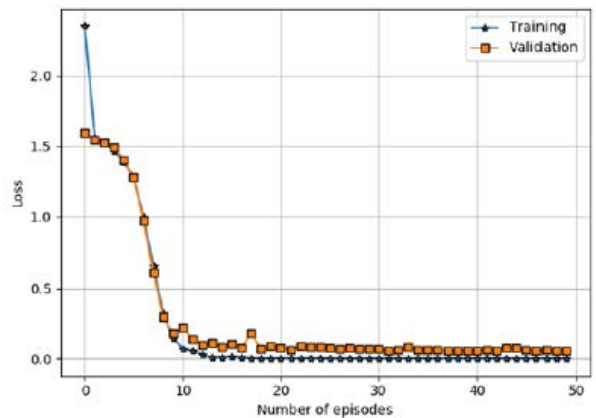
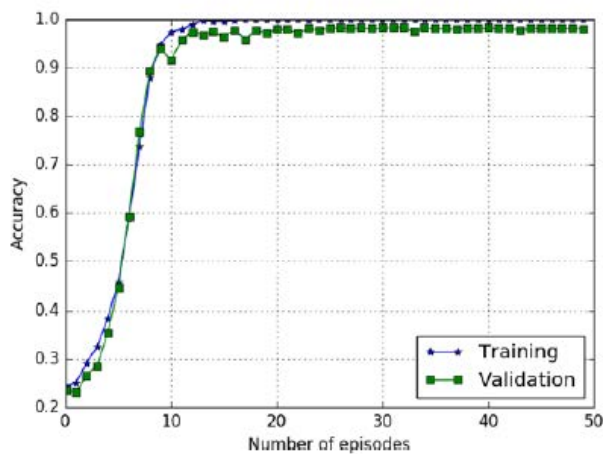


Figura 58: Confronto fra accuracy

lo stesso modello su un'infrastruttura cloud Google impiegando 1 ora e 17 minuti per completare il processo di apprendimento.

4.2.7 Conclusione e sviluppi futuri

Questo documento ha presentato un sistema intelligente per il supporto nel trattamento terapeutico domiciliare di soggetti anziani o pazienti con disabilità, riducendo la discontinuità delle cure e potenziali eventi avversi.

Il sistema è in grado di auto-adattarsi rispetto alle attitudini del paziente ed è composto da tre agenti principali. Sono state adottate diverse tecnologie di Intelligenza Artificiale, come il Bayesian Reinforcement Learning e il Deep Learning.

Futuri lavori saranno indirizzati al miglioramento della fase di verifica, ovvero ai meccanismi di controllo che assicurano l'aderenza alla terapia. Prima di tutto, verrà sperimentato un set più ampio di farmaci. Ciò richiede la costruzione di un dataset più grande di immagini di scatole di pillole. L'obiettivo futuro più rilevante riguarda, in ogni caso, l'integrazione di altri tipi di meccanismi per la verifica dell'assunzione del farmaco. In effetti, l'attuale implementazione non può assicurare che il paziente abbia effettivamente ingerito la pillola dopo aver maneggiato il corretto portapillole. Il lavoro futuro mirerà a riconoscere le azioni relative all'estrazione di una pillola dalla scatola e la sua effettiva assunzione.

Bibliografia

1. *Evaluation of Data Mining Classification Models*. Abdalla M. EL-HABI, Mohammed El-Ghareeb. s.l. : IUG Journal of Natural and Engineering Studies, 2014, Vols. Vol.22, No.1, pp 151-165.
2. *The IT Leader's Definitive Guide to Predictive Modeling*. Hoogenhout, Michelle. s.l. : PLUTORA BLOG - BUSINESS INTELLIGENCE, RELEASE MANAGEMENT, VALUE STREAM MANAGEMENT, FEBRUARY 18, 2020, //www.plutora.com/blog/predictive-modeling.
3. *Data Mining: tecniche di trasformazione dei dati*. Moschese, Giuseppe. s.l. : APOGEO, 20 Ottobre 2004.
4. *Materiale Didattico messo a disposizione*. s.l. : dall' ALMA MATER STUDIORUM Università degli Studi di Bologna per il corso di Machine Learning – anno accademico 2019/2020 e dalla FEDERICO II Università degli Studi di Napoli per il corso Metodi Statistici per i Data Mining.
5. *Training and Testing Data Sets*. microsoft. s.l. : microsoft.com, 05/08/2018.
6. *Data Preprocessing*. techopedia.com. s.l. : tecnopedia, Vols. definition - 14650.
7. *Data Mining Process: Cross-Industry Standard Process for Data Mining*. Patidar, Shailna. s.l. : DZone - A DEVADA MEDIA - PROPERTY, Aug. 13, 18 .
8. *L'intelligenza artificiale ci aiuta a vivere meglio: come il robot Prisca*. Tomaso, Vittorio Di. s.l. : CORRIERE DEL MEZZOGIORNO, Feb 6, 2020.

9. *INTELLIGENZA ARTIFICIALE: COS'È, COME FUNZIONA E A COSA SERVE?* s.l. : Il portale dedicato all'Intelligenza Artificiale.
10. *Deep Learning: Methods and Applications*. L. Deng, D.Yu. 2014.
11. *Machine Learning IBM Limited Edition*. Judith Hurwitz, Daniel Kirsch. s.l. : John Wiley & Sons, Inc, 2018.
12. *Classification and regression trees*. Monterey. Leo, Breiman, et al. s.l. : Chapman and Hall/CRC, 1 edition - January 1, 1984.
13. *Artificial Intelligence A Modern Approach -Third Edition* . Stuart Russell, Peter Norvig. 2010, Vols. Stuart Russell and Peter Norvig, Editors.
14. *A Top Machine Learning Algorithm Explained: Support Vector Machines (SVMs)*. Harrad, William. s.l. : Velocity Business Solutions, February 27th, 2020.
15. *Support Vector Machines*. tutorial, scikit-learn-. s.l. : scikit-learn developers (BSD License), 2019.
16. *The Elements of Statistical Learning - Data Mining, Inference, and Prediction*. Trevor Hastie, Robert Tibshirani, Jerome Friedman. s.l. : Springer Series in Statistics, Second Edition .
17. *Algoritmo k-means: cos'è e come funziona?* GOVONI, LORENZO. s.l. : LORENZO GOVONI - Business e Tecnologia, 2020.
18. *Fuzzy Logic* . s.l. : WolframComputation Meets Knowledge, 2004.
19. *Density-Based Clustering*. Manco, Giuseppe. Mercoledì, 24 novembre 2004, Vols. Chapter 8, Han and Kamber - Chapter 14, Hastie , Tibshirani and Friedman.
20. *Neural Networks and Deep Learning*. Nielsen, Michael. s.l. : Michael Nielsen's project announcement mailing list, Dec 2019.
21. *Understanding The Recurrent Neural Network*. Shekhar, Amit. s.l. : MindOrks, MindOrks.
22. *Neural Networks as Cybernetic Systems*. Cruse, Holk. s.l. : ResearchGate, January 2006.
23. *Guida alle architetture di reti profonde*. Sala, Joyce Xu - Traduttrice: Sabrina. s.l. : Deep Learning Italia, 5 Giugno 2018.
24. *Understanding Encoder-Decoder Sequence to Sequence Model*. Kostadinov, Simeon. s.l. : Towards Data Science, Feb 5, 2019.
25. *Introduction to Autoencoders*. Hubens, Nathan. s.l. : DeepLarningItalia , 8 May 2018.
26. *Validazione efficace dei modelli di machine learning*. Biscaglia, Nicola. s.l. : Milliman, 22 Maggio 2017.
27. *Cosa significa Overfitting e Underfitting*. Soriano, Domenio. s.l. : Soriano Domenico, 22 Febbraio, 2020.

28. *Difference between ml and deep learning with respect to splitting of the dataset into*. Shimpi, Tanmay. s.l. : mc.ai, 1. APRIL 2019, Vol. Source: Deep Learning on Medium.
29. *What is Amazon Machine Learning? Services*, Amazon Web. s.l. : AWS, 2020, Vol. Amazon Machine Learning.
30. *Artificial Intelligence & Deep Learning for Medical Diagnosis*. Kharkovyna, Oleksii. s.l. : Towards Data Science, Nov 13, 2019 .
31. *Automatic ECG Diagnosis Using Convolutional*. Roberta Avanzato, Francesco Beritelli. Department of Electrical, Electronic and Computer Engineering, University of Catania, 95125 Catania, Italy; : Electronics - MDPI, 8 June 2020.
32. *Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network*. Awni Y. Hannum, Pranav Rajpurkar, Masoumeh Haghpanahi, Geoffrey H. Tison, Codie Bourn, Mintu P. Turakhia, Andrew Y. Ng. s.l. : nature medicine, January 2019.
33. *A Hybrid Deep Network for the Early Identifi*. Paragliola, G. - Coronato, A. Napoli : AMICO project partnership with National Programs (PON) of the Italian Ministry of Education, Universities and Research (MIUR), 2020.
34. *Early diagnosis saves lives: focus on patients with hypertension*. Gulec, S. 3 , 332 - 334., 2013, Vol. Kidney International Supplements.
35. *The 2012 canadian hypertension education program recommendations for the management of hypertension: Blood pressure measurement, diagnosis, assessment of risk, and therapy*. Daskalopoulou, S. S., Khan, N. A., Quinn, R. R., Ruzicka, M., McKay, D. W., Hackam, D. G., Rabkin, S. W., Rabi, D. M., Gilbert, R. E., Padwal, R. S., & et al. 28 , 270 - 287, 2012, Vol. Canadian Journal of Cardiology.
36. *What is the evidence base for diagnosing hypertension and for subsequent blood pressure treatment targets in the prevention of cardiovascular disease?* . Schwartz, C. L., & McManus, R. J. 13, 2015, Vol. BMC Medicine.
37. *Automatic prediction of cardiovascular and cerebrovascular events using heart rate variability analysis* . Melillo, P., Izzo, R., Orrico, A., Scala, P., Attanasio, M., Mirra, M., De Luca, N., & Pecchia, L. 10, e0118504, 2015, Vol. PLOS ONE .
38. *Analysis of importance of pre-processing in prediction of hypertension* . Shobha, K., & Nickolas, S. 6, 209-2014, 2018, Vol. CSI Transactions on ICT .
39. *A hypertension risk prediction model based on bp neural network* . Zhao, H., Ma, Z., & Sun, Y. s.l. : IEEE, 2019, Vol. In 2019 International Conference on Networking and Network Applications (NaNA).
40. *A machine-learning-based prediction method for hypertension outcomes based on medical data*. Chang, W., Liu, Y., Xiao, Y., Yuan, X., Xu, X., Zhang, S., & Zhou, S. 9, 178, Vols. Diagnostics, .

41. *PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals.* Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., & Stanley, H. E. 101, e215-e220, 13 giugno 2000, Vol. Circulation Electronic .
42. *New approaches in hypertension management: a review of current and developing technologies and their potential impact on hypertension care.* . Kitt, J., Fox, R., Tucker, K. L., & McManus, R. J. (2019). 21, 2019, Vols. Current Hypertension Reports,.
43. *Implementing abpm into clinical practice.* Hinderliter, A., Voora, R. A., & Viera, A. J. 20, 2018, Vol. Current Hypertension Reports.
44. *Analysis of importance of preprocessing in prediction of hypertension.* Shobha, K., & Nickolas, S. 6, 209-214, 2018, Vols. CSI Transactions on ICT,.
45. *Predicting hypertension without measurement: A non-invasive, questionnaire-based approach.* Wang, A., An, N., Chen, G., Li, L., & Alterovitz, G. 42 , 7601-7609, 2015, Vol. Expert Systems with Applications.
46. *Effective heart disease prediction using hybrid machine learning techniques.* Mohan, S., Thirumalai, C., & Srivastava, G. 7,81542-81554., 2019, Vol. IEEE Access.
47. *A physiological time series dynamicsbased approach to patient monitoring and outcome prediction.* Lehman, L.-w. H., Adams, R. P., Mayaud, L., Moody, G. B., Malhotra, A., Mark, R. G., & Nemati, S. 19,1068-1076, 2015, Vol. IEEE Journal of Biomedical and Health Informatics.
48. *Prediction of blood pressure variability using deep neural networks.* Koshimizu, H., Kojima, R., Kario, K., & Okuno, Y. 136,104067, 2020, Vol. International Journal of Medical Informatics.
49. *The prediction of hypertension based on convolution neural network.* Luo, Y., Li, Y., Lu, Y., Lin, S., & Liu, X. (pp. 2122{2127}), 2018, Vol. 2018 IEEE 4th International Conference on Computer and Communications (ICCC).
50. *Time series classification from scratch with deep neural networks: A strong baseline.* Wang, Z., Yan, W., & Oates, T. 2016.
51. *Learning deep features for discriminative localization.* Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. 2015.
52. *Towards a universal neural network encoder for time series.* Serrà, J., Pascual, S., & Karatzoglou, A. 2018.
53. *Neural machine translation by jointly learning to align and translate.* Bahdanau, D., Cho, K., & Bengio, Y. 2014.
54. *Data augmentation for time series classification using convolutional neural networks.* Guennec, A. L., Malinowski, S., & Tavenard, R. 2016.

55. *Time series classification using multi-channels deep convolutional neural networks*. In F. Li, G. Li, S.-w. Hwang, B. Yao, & Z. Zhang (Eds.). Zheng, Y., Liu, Q., Chen, Y., Enhongand Ge, & Zhao, J. L. (pp. 298-310), 2014, Vols. Web-Age Information Management.
56. *Convolutional neural networks for time series classification*. . Zhao, B., Lu, H., Chen, S., Liu, J., & Wu, D. 28, 162-168, 2017, Vol. Journal of Systems Engineering and Electronics.
57. *Machine learning for sequential data: A review*. In T. Caelli, A. Amin, R. P. W. Duin, D. de Ridder, & M. Kamel (Eds.). Dietterich, T. G. pp 15-30, August 6-9,2002, Vols. Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshops SSPR 2002 and SPR 2002 Windsor, Ontario, Canada,August 6{9, 2002 Proceedings.
58. Lipton, Z. C. (2015). *A critical review of recurrent neural networks for sequence learning*. *abs/1506.00019*, 2015, Vols. CoRR.
59. 1962). *Principles of neurodynamics. perceptrons and the theory of brain mechanisms*. Orbach, J. 7,218, 1962, Vols. Archives of General Psychiatry.
60. *Deep Learning*. MIT Press. Goodfellow, I., Bengio, Y., & Courville, A. 2016.
61. *A Reinforcement Learning and Deep Learning based Intelligent System for the Support of Impaired Patients in Home Treatment*. Muddasar Naeem, Giovanni Paragliola, Antonio Coronato. October 8, 2020.
62. *Adherence to Long-Term Therapies: Evidence for Action*. WHO. 203.
63. *Medication Errors*. WHO. 2016.
64. *Care homes' use of medicines study: Prevalence, causes and potential harm of medication errors in care homes for older people*. Barber, N., Alldred, D., Raynor, D., Dickinson, R., Garfield, S., Jesson, B., Lim, R., Savage, I., Standage, C., Buckle, P., Carpenter, J., Franklin, B., Woloshynowych, M., & Zermansky, A. 18,341-346, 2009, Vol. Quality & safety in health care.
65. *Adverse Outcomes Associated with Inappropriate Drug Use in Nursing Homes*. 39,405-411, 2005, Vol. Annals of Pharmacotherapy.
66. *Streaming ema public communication on medication*. AGENCY, EMA - EUROPEAN MEDICINES. s.l. : Stakeholders and Communication Division, 27 November 2015, Vol. Science Medicines health.
67. *Noncompliance with directly observed therapy for tuberculosis*. Burman, W. J., Cohn, D. L., Rietmeijer, C. A., Judson, F. N., Reves, R. R., & Sbarbaro, J. A. 111,1168-1173, 1997, Vol. Chest.
68. *Medication nonadherence and the outcomes of patients with quiescent ulcerative colitis*. Kane, S., Huo, D., Aikens, J., & Hanauer, S. 114,39-43, 2003, Vol. The American Journal of Medicine.
69. *Europe, World Health Organization - Regional Office for*.
70. *Deep Convolutional Neural Network for Inverse Problems in Imaging*. . Jin, K. H., McCann, M. T., Froustey, E., & Unser, M. 26,4509-4522, 2017, Vol. IEEE Transactions on Image Processing.

71. *An enhanced deep learning approach for brain cancer MRI images classification using residual networks.* . Abdelaziz Ismael, S. A., Mohammed, A., & Hefny, H. 102,101779, 2020, Vol. *Artificial Intelligence in Medicine*.
72. *Reinforcement learning for intelligent healthcare applications: A survey.* Coronato, A., Naeem, M., De Pietro, G., & Paragliola, G. 109,101964, 2020, Vol. *Artificial Intelligence in Medicine*.
73. *Gait anomaly detection of subjects with parkinson's disease using a deep time series-based approach.* . Paragliola, G., & Coronato, A. 6,73280-73292, 2018, Vol. *IEEE Access*.
74. *A situation-aware system for the detection of motion disorders of patients with autism spectrum disorders.* Coronato, A., Pietro, G. D., & Paragliola, G. 41,7868-7877, 2014, Vol. *Expert Syst. Appl.*
75. *Deep reinforcement learning for automated radiation adaptation in lung cancer.* . Tseng, H.-H., Luo, Y., Cui, S., Chien, J.-T., Ten Haken, R. K., & Naqa, I. E. 44 , 6690-6705, 2017, Vol. *Medical Physics*.
76. *Deep Reinforcement Learning for Sepsis Treatment.* Raghu, A., Komorowski, M., Ahmed, I., Celi, L. A., Szolovits, P., & Ghassemi, M. abs/1711.0 , 2017, Vol. *CoRR*.
77. *Risk management for nuclear medical department using reinforcement learning algorithms.* . Paragliola, G., & Naeem, M. 5,105-113, 2019, Vol. *Journal of Reliable Intelligent Environments*.
78. *An innovative risk assessment methodology for medical information systems.* . Coronato, A., & Cuzzocrea, A. pp. 1-1, 2020, Vol. *IEEE Transactions on Knowledge and Data Engineering*.
79. *A clustering based methodology to support the translation of medical specifications to software models.* Gargiulo, F., Silvestri, S., & Ciampi, M. 71,199-212., 2018, Vol. *Applied Soft Computing*.
80. *Impact of a Text Messaging Pilot Program on Patient Medication Adherence.* Foreman, K. F., Stockl, K. M., Le, L. B., Fisk, E., Shah, S. M., Lew, H. C., Solow, B. K., & Curtis, B. S. 34, 1084-1091 , 2012, Vol. *Clinical Therapeutics*.
81. *A text messaging intervention to promote medication adherence for patients with coronary heart disease: A randomized controlled trial.* . Park, L. G., Howie-Esquivel, J., Chung, M. L., & Dracup, K. 94,261-268, 2014, Vol. *Patient Education and Counseling*.
82. *Smart drugs:Improving healthcare using Smart Pill Box for Medicine Reminder and Monitoring System.* Abdul Minaam, D. S., & Abd-Elfattah, M. 3,443-456, 2018, Vol. *Future Computing and Informatics Journal* .
83. *AI empowered context-aware smart system for medication adherence.* Wu, Q., Zeng, Z., Lin, J., & Chen, Y. 1,102-109, 2017, Vol. *International Journal of Crowd Science*.
84. *A smart pill box with remind and consumption confirmation functions .* Wu, H., Wong, C., Liu, P., Peng, S., Wang, X., Lin, C., & Tu, K. pp. 658-659, s.l., 2015, Vol. *IEEE 4th Global Conference on Consumer Electronics (GCCE)* .
85. *Smart home medication reminder system.* Ramljak, M. pp.1-5, 2017, Vols. *2017 25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)* .

86. *Pill Care-The Smart Pill Box with Remind, Authenticate and Confirmation Function*. Nijiya Jabin Najeeb, P., Rimna, A., Safa, K., Silvana, M., & Adarsh, T. pp. 1-5, 2018, Vol. In *2018 International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR)*.
87. *A Smartphone-based Medication Self-management System with Realtime Medication Monitoring*. M. Hayakawa, Y. Uchimura, K. Omae, K. Waki, H. Fujita, K. O. Vol. 4.
88. *The Autonomous Pill Dispenser with Alarm and Mobile Notifications*. Patil, J. 7,2502-2508, 2019, Vol. *International Journal for Research in Applied Science and Engineering Technology*.
89. *Design and development of Medication Assistant: older adults centred design to go beyond simple medication reminders*. Teixeira, A., Ferreira, F., Almeida, N., Silva, S., Rosa, A. F., Pereira, J. C. & Vieira, D. 16,545-560, 2016, Vol. *Universal Access in the Information Society*.
90. *Machine learning classification of medication adherence in patients with movement disorders using non-wearable sensors*. . Tucker, C. S., Behoora, I., Nembhard, H. B., Lewis, M., Sterling, N. W., & Huang, X. 66,120-134, 2015, Vol. *Computers in Biology and Medicine*.
91. *Improving medication adherence with adjuvant aromatase inhibitor in women with breast cancer: study protocol of a randomised controlled trial to evaluate the effect of short message service (sms) reminder*. . He, Y., Tan, E. H., Wong, A. L. A., Tan, C. C., Wong, P., Lee, S. C., & Tai, B. C. 18, 2018, Vol. *BMC Cancer*.
92. *Eudroid: a formal language specifying the behaviour of iot devices*. Buono, P., Cassano, F., Legretto, A., & Piccinno, A. 12,425-429, 2018, Vol. *IET Software*.
93. *A multi-step usability evaluation of a self-management app to support medication adherence in persons living with hiv*. . Beauchemin, M., Gradilla, M., Baik, D., Cho, H., & Schnall, R. 122,37-44, s.l., 2019, Vol. *International Journal of Medical Informatics*.
94. Sutton, R. S., & Barto, A. G. 1998 *Reinforcement Learning: An Introduction*, Vol. {MIT} Press.
95. *Foundations and Trends in Machine Learnin*. M. Ghavamzadeh, J. P., S. Mannor, & Tamar, A. pp 102-109,
96. *A Reinforcement Learning Based Intelligent System for the Healthcare Treatment Assistance of Patients with Disabilities*. In C. Esposito, J. Hong, & K.-K. R. Choo (Eds.). Coronato, A., & Naeem, M. pp. 15-28, 2019, Vols. *Pervasive Systems, Algorithms and Networks*.
97. *Adaptive Treatment Assisting System for Patients Using Machine Learning*. Naeem, M., Coronato, A., & Paragliola, G. 2019, Vols. In *2019 6th International Conference on Social Networks Analysis, Management and Security, SNAMS 2019*.
98. *Reinforcement Learning: An Introduction*. Sutton, R. S., & Barto, A. G. 1998, Vol. fMITg Press.
99. *Model-Based Bayesian Exploration*. Dearden, R., Friedman, N., & Andre, D. 2013, Vols. *CoRR*, abs/1301.6 .

100. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. Simonyan, K., & Zisserman, A. 2015, Vol. *International Conference on Learning Representations*.
101. *Reinforcement learning: A survey*. Kaelbling, L. P., Littman, M. L., & Moore, A. W. 1996, Vols. *CoRR*, cs.AI/9605103.
102. *A tutorial on thompson sampling*. Russo, D., Roy, B. V., Kazerouni, A., & Osband, I. 2017, Vols. *CoRR*, abs/1707.02038.
103. Finlay, Steven. *Predictive Analytics, Data Mining and Big Data: Myths, Misconceptions and Methods*. s.l. : Palgrave Macmillan, 2014.
104. *Lo studio del clima..LE RETI NEURALI..ma.. cosa sono?* Langone, Raffaele. s.l. : *Rivista online - L'informazione che non informa*, 7 Novembre 2009.
105. *Unsupervised Bump Hunting Using Principal Components*. Daniel A Díaz-Pachón, Jean-Eudes Dazard† J. Sunil Rao. September 2014.
106. *Mode-finding algorithms revisited*. M. Kaykobad, Rezaul Alam Chowdhury, MD. Sanaul Hoque. s.l. : *International Journal of Computer Mathematics* , 19 Mar 2007, Vol. 69.
107. *Fully Connected Deep Networks*. Bharath Ramsundar, Reza Bosagh Zadeh. s.l. : *O'Reilly home*, 2020, Vols. *TensorFlow for Deep Learning - Chapter 4*.
108. *Cognitive Machine-Learning Algorithm for Cardiac Imaging*. Partho P. Sengupta, Yen-Min Huang, Manish Bansal, Ali Ashrafi, Matt Fisher, Khader Shameer, Walt Gall, and Joel T. Dudley. Jun 2016, Vol. *Circulation: Cardiovascular* .