

Sviluppo di un modello basato su Deep Learning per la sottrazione del fondo baseline in spettri Raman non-lineari

Mattia Marfoli 1843940
Paolo Mangini 1849724
Daniele Lanciotti 1849379

14 luglio 2021

Indice

1	Descrizione del problema fisico	2
2	Analisi del dataset	2
3	Approccio di denoising	2
3.1	Scelta e struttura del modello	2
3.2	Training del modello	3
3.3	Determinazione dei picchi	3
3.4	Caratteristiche dei picchi	3
4	Analisi dei risultati	4
4.1	Analisi sulle caratteristiche dei picchi	5
4.2	Analisi sulla classificazione dei picchi	6
5	Conclusioni	6

1 Descrizione del problema fisico

Lo Scattering Raman Stimolato (SRS) è una tecnica spettroscopica, non-lineare del terzo ordine, basata sull'effetto Raman indotto da impulsi laser corti in tempo e rappresenta uno strumento potente in grado di indagare le proprietà vibrazionali della materia su scala atomica.

Nella spettroscopia Raman convenzionale, i modi roto-vibrazionali sondati hanno natura incoerente essendo attivati termicamente tramite l'assorbimento di un fotone ("Raman pulse" con frequenza ω_P).

Al contrario, la tecnica SRS stimola coerentemente i modi normali mediante due impulsi ultracorti: il "Raman pulse" (ω_P) e un secondo segnale, cioè lo "Stokes probe" di frequenza ω_S .

Quando la differenza di frequenza tra i due fotoni ($\omega_P - \omega_S$) assomiglia a quello di una specifica transizione vibrazionale (o rotazionale) il verificarsi di questa transizione è amplificato in modo risonante. L'utilizzo di un raggio laser a pompa a frequenza costante e di un raggio laser Stokes a frequenza scansionata (o viceversa) consente di svelare l'impronta spettrale della molecola ma anche di determinare il moto atomico su scale temporali del femtosecondo.

A causa della non linearità di questo fenomeno i segnali misurati sperimentalmente sono generalmente più complessi e difficili da interpretare rispetto a quelli ottenuti con il Raman spontaneo: le forme di righe appaiono sovrapposte ad un segnale di background (baseline), e possono subire forti distorsioni, con profili spettrali sia negativi che dispersivi. Uno spettro SRS consiste quindi in due componenti che non sono separabili sperimentalmente: una baseline larga spettralmente (con un offset) e una serie di caratteristiche tipiche degli spettri Raman come picchi, buche o forme dispersive.

2 Analisi del dataset

Il dataset si compone di 10K eventi (8k di training set e 2k di test set) costituiti da tensori numpy rappresentanti 400 campionamenti dello spettro Raman. Per ogni evento è disponibile sia lo spettro misurato (X_{train}/X_{test}) sia il ground truth del segnale Raman (Y_{train}/Y_{test}).

Per il dataset fornito non c'è stato bisogno di un preprocessing dei dati, infatti non sono stati necessari né la normalizzazione, poiché i dati erano già nell'intervallo $[0, 1]$, né lo splitting tra training e test set perché i dati erano già organizzati in tal modo.

Nella figura sottostante sono riportati degli esempi di come si presenta il test set.

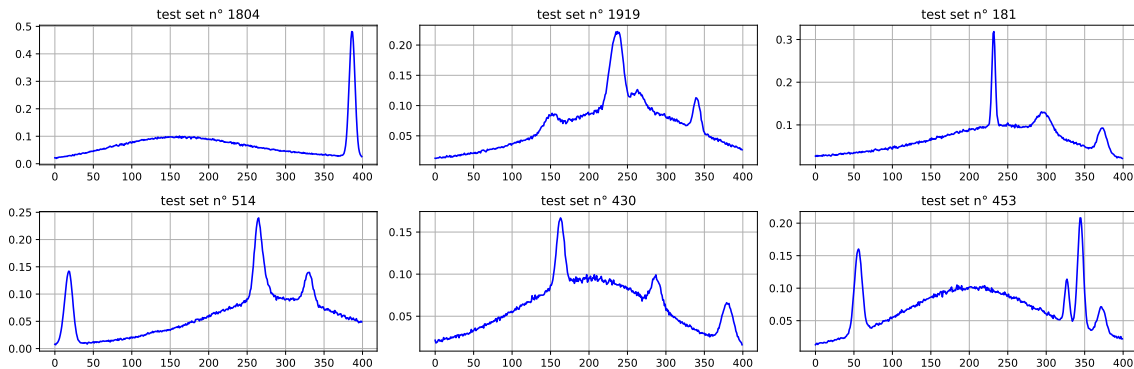


Figura 1: Campione di 6 eventi casuali del test set.

3 Approccio di denoising

Il problema consiste nella separazione del segnale di baseline dagli spettri Raman: per risolverlo si è pensato di progettare ed addestrare un modello di DL supervisionato che, dopo aver appreso l'andamento del segnale di baseline, fosse in grado di identificarlo e rimuoverlo in modo da ottenere degli spettri Raman "puliti" (previsione del ground truth).

3.1 Scelta e struttura del modello

Il modello di DL scelto è stato l'autoencoder, perché, avendo a disposizione sia gli spettri Raman reali sia il ground truth, si è pensato che forzare con un encoder la rappresentazione dell'input in uno spazio latente di dimensione minore a quella iniziale avrebbe costretto il modello a imparare qualcosa che lo aiutasse in questo processo, ovvero l'andamento dei pattern dei segnali di baseline, essendo questi simili e comuni a tutti gli spettri; la parte di decoding del modello utilizzato aveva lo scopo di restituire una predizione del ground truth, dopo l'apprendimento avvenuto nella parte di encoding.

L'autoencoder realizzato è così strutturato:

L'encoder prende in input lo spettro Raman reale e consiste in 3 conv1D con attivazione Relu (da 32, 16 e 8 filtri) intervallate da dei Maxpooling1D (da 2 filtri ciascuno) che riducono l'input da un vettore di shape (400, 1) ad un vettore

nello spazio latente di shape (50, 8). (struttura vettore : (dimensione,filtri))

Il decoder prende in input l'output dell'encoder e consiste di 3 conv1D con attivazione Relu (da 8, 16 e 32 filtri) intervallate da degli Upsampling1D (da 2 filtri ciascuno) e una conv1D finale con attivazione sigmoide che restituisce un output delle stesse dimensioni del vettore di input, ovvero di shape (400, 1).

3.2 Training del modello

Il modello è stato allenato per 600 epoche perché si è osservato che con più epoche (per esempio 1000) aumentavano significativamente le oscillazioni sulla mse del validation set, invece con meno epoche (per esempio 300) la mse del validation set si fermava a valori maggiori di quelli ottenuti con più epoche: in entrambi i casi le prestazioni del modello peggioravano.

Il valore scelto per il batch size è stato 32.

Il validation set utilizzato durante la fase di addestramento è stato preso come il 20% del training set.

La loss utilizzata è stata la "binary cross-entropy", poiché è la funzione di perdita corrispondente ad un "decoding layer" con funzione di attivazione sigmoide. Come ottimizzatore è stato scelto "Adam" e come metrica si è optato per la "mse", perché stabile e consistente nel monitorare la distanza tra il risultato vero e quello predetto da una rete neurale. Nel grafico a destra sono riportati gli andamenti della loss e della mse per training e validation set durante l'allenamento iterato per 600 epoche.

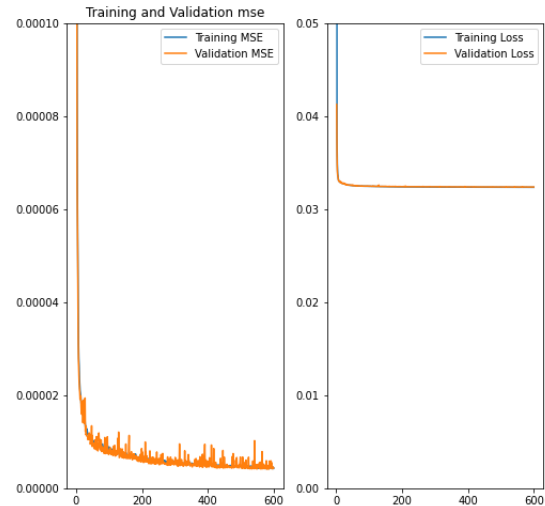


Figura 2: Grafico della loss e della mse per il training e validation set.

3.3 Determinazione dei picchi

Per la determinazione dei picchi degli spettri Raman (ground truth), sia predetti che reali, si è utilizzata una funzione custom dato che le funzioni già esistenti, come ad esempio `scipy.signal.find_peaks`¹, spesso non riuscivano nemmeno ad individuare correttamente tutti i picchi negli spettri più semplici da analizzare, ovvero i ground truth reali: questa situazione rendeva inapplicabile l'analisi dei risultati tramite una discussione sui picchi poiché la maggior parte delle volte la differenza trovata tra i picchi degli spettri reali e predetti era dovuta all'inefficacia del modo di determinarli che effettivamente ad un errore del denoiser. La funzione custom creata individua i picchi come dei massimi locali che vengono selezionati come i punti tali che siano maggiori sia dei 5 punti precedenti che dei 5 successivi: non ci si è limitati al solo confronto con i due valori adiacenti per evitare che delle oscillazioni presenti nel test set fossero riconosciute come massimi locali.

Inoltre è stata impostata una soglia, posta uguale a 0, in modo tale che venissero scartati punti critici non interpretabili fisicamente come dei picchi.

Infine è stata messa un'altra condizione nell'individuazione dei massimi locali, ovvero la distanza minima tra un massimo e l'altro, infatti i picchi hanno una loro larghezza e abbiamo voluto dare questa informazione all'algoritmo per ridurre al minimo output privi di significato fisico.

Per migliorare l'efficacia nel trovare i picchi predetti, la funzione custom è stata applicata agli spettri Raman predetti solo dopo aver effettuato su questi ultimi un processo di regolarizzazione mediante la funzione `scipy.signal.savgol_filter`². Questo processo non è stato necessario per i ground truth reali poiché funzioni già abbastanza regolari.

3.4 Caratteristiche dei picchi

Per determinare le caratteristiche dei picchi quali larghezza, altezza e prominence sono stati usati approcci diversi: per determinare larghezza e prominence sono state usate due funzioni apposite della libreria "SciPy" di python (rispettivamente `scipy.signal.peak_widths`³ e `scipy.signal.peak_prominences`⁴), mentre l'altezza dei picchi è stata determinata semplicemente come il valore dell'ordinata del test set (spettri Raman reali) corrispondente alle ascisse dei massimi locali già note perché restituite dalla funzione illustrata nella sezione precedente.

¹Documentazione SciPy di `scipy.signal.find_peaks` ([link](#))

²Documentazione SciPy di `scipy.signal.savgol_filter` ([link](#)) utilizzata con parametri `window_length = 13` e `polyorder = 3`.

³Documentazione SciPy di `scipy.signal.peak_widths` ([link](#))

⁴Documentazione SciPy di `scipy.signal.peak_prominences` ([link](#))

4 Analisi dei risultati

Per verificare che, una volta addestrato il modello, questo avesse realmente imparato a individuare il segnale di baseline in uno spettro Raman, sono state effettuate delle previsioni sul campione di test (set nuovo al modello). Nella figura sottostante sono mostrati i grafici relativi a sei eventi (scelti casualmente tra i 1861 casi in cui il numero di picchi predetti e quelli veri è risultato lo stesso), i cui spettri Raman di partenza sono esposti in figura 1. Per ognuno di questi eventi sono riportati, da sinistra a destra, la previsione del modello sul ground truth, il corrispondente ground truth reale, e il pattern del segnale di baseline predetto sovrapposto a quello vero (trovato come $X_test - Y_test$).

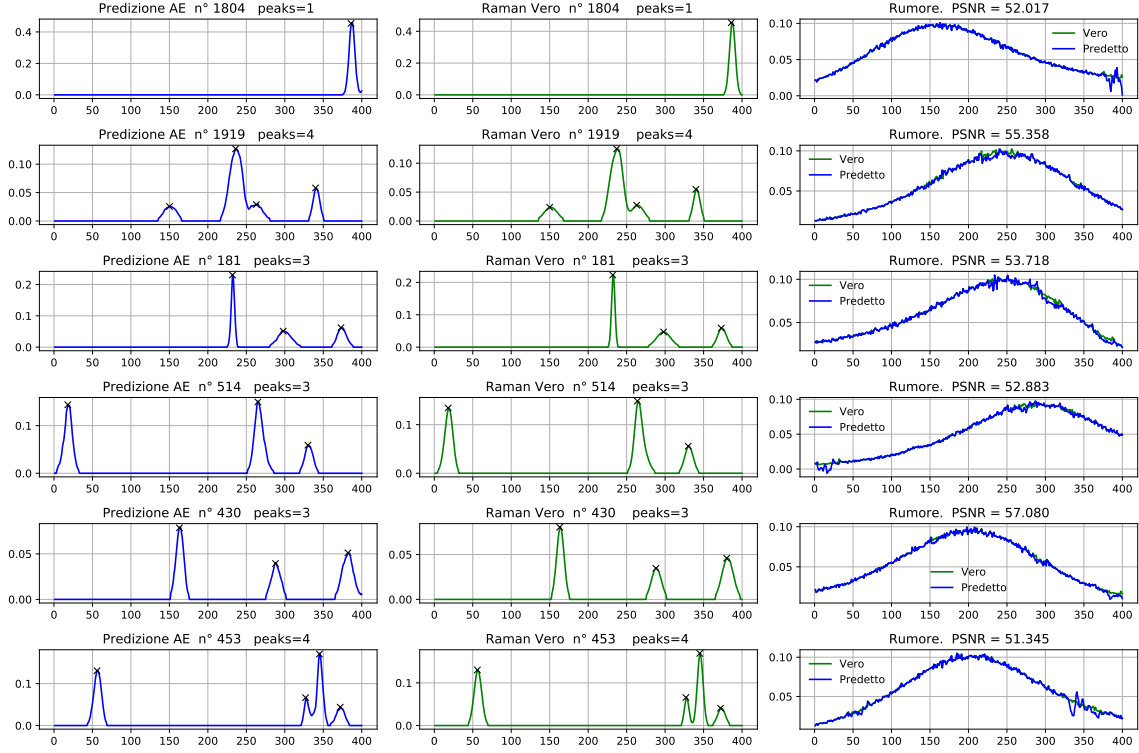


Figura 3: Campione (corrispondente a quello in figura 1) di predizioni corrette del ground truth del modello messe a confronto con i relativi spettri veri e plot del segnale di baseline.

Nella figura 4 (a) sottostante sono riportati cinque grafici, scelti casualmente, tra gli spettri che, dopo il processo di denoising, hanno riscontrato una differenza tra il numero di picchi predetti e reali pari ad uno (135 casi). Nella figura 4 (b) sottostante vengono mostrati i soli 4 casi in cui la differenza del numero di picchi tra gli spettri predetti e reali è pari a due; si può osservare come la differenza di picchi sia un sintomo di imprecisione del modello nel predire quei particolari segnali di baseline, infatti, in questi grafici ad ogni errore di classificazione dei picchi corrisponde una fluttuazione non trascurabile del segnale di baseline predetto rispetto a quello reale.

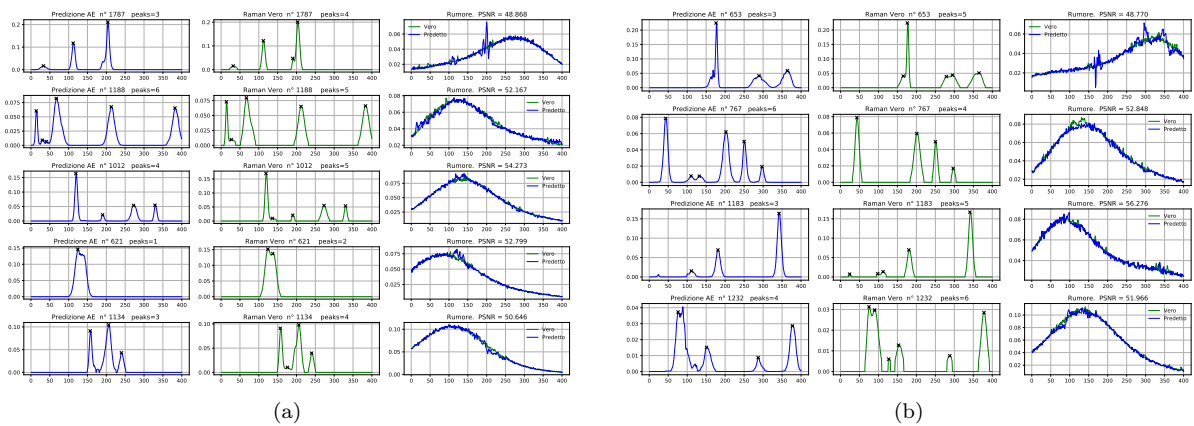


Figura 4: Predizioni del ground truth del modello con differenze di un picco (a) e due picchi (b), messe a confronto con gli spettri veri e plot del segnale di baseline.

Per effettuare un'analisi critica dei risultati si è pensato che una stima quantitativa delle prestazioni del modello nell'identificare i segnali di diverse tipologie potesse essere portata avanti attraverso lo studio e il confronto dei picchi degli spettri Raman (ground truth) predetti e reali, in particolare ci si è concentrati su quantità, larghezza, altezza e vicinanza dei picchi.

Inoltre un'altra grandezza utilizzata per stimare la bontà del processo di denoising effettuato è il PSNR (Peak signal-to-noise ratio): il valore medio del PSNR indica quanto la previsione del ground truth assomigli effettivamente a quello reale, più alto è questo valore maggiore è la qualità di apprendimento, da parte del modello, del pattern del segnale di baseline.

I valori del PSNR calcolati per i vari tipi di predizione effettuate dal modello sono riportate nella tabella sottostante.

predizioni	n° spettri	PSNR medio
tutte	2000	54.74 ± 3.37
$\Delta\text{picchi} = 0$	1861	54.93 ± 3.30
$\Delta\text{picchi} = 1$	135	52.10 ± 3.12
$\Delta\text{picchi} = 2$	4	52.47 ± 2.67

Tabella 1: Valori di PSNR medio per ogni tipo di predizione.
 Δpicchi è la differenza tra numero di picchi predetti e reali.

Il risultato aspettato è coerente con le aspettative infatti il PSNR più alto è quello per le previsioni con stesso numero di picchi poiché l'individuazione corretta dei picchi è strettamente legata alla qualità del processo di denoising.

Il valore del PSNR per le predizioni sbagliate (per uno o due picchi) è approssimativamente uguale, anche se teoricamente dovrebbe essere leggermente maggiore il PSNR delle predizioni con un solo errore di picco, questo non avviene probabilmente è dovuto all'esigua quantità di predizioni con differenza di due picchi con cui è stata calcolata la media del PSNR.

4.1 Analisi sulle caratteristiche dei picchi

Per stimare quantitativamente l'efficacia dell'individuazione dei picchi dei ground truth predetti si sono utilizzati degli istogrammi, i quali mostrano di quanto le previsioni si discostino dai risultati reali nel caso in cui il numero dei picchi determinato nel caso predetto e quello reale sia lo stesso (1861 casi).

Negli istogrammi sottostanti sono riportati la distanza media e assoluta e la differenza media e assoluta di altezza, larghezza e prominence tra i picchi individuati negli spettri Raman predetti e in quelli reali (con assoluta si intende la somma delle differenze delle grandezze riguardanti i picchi dello spettro Raman corrispondente, mentre per media si divide la differenza assoluta per il numero di picchi presenti nello spettro).

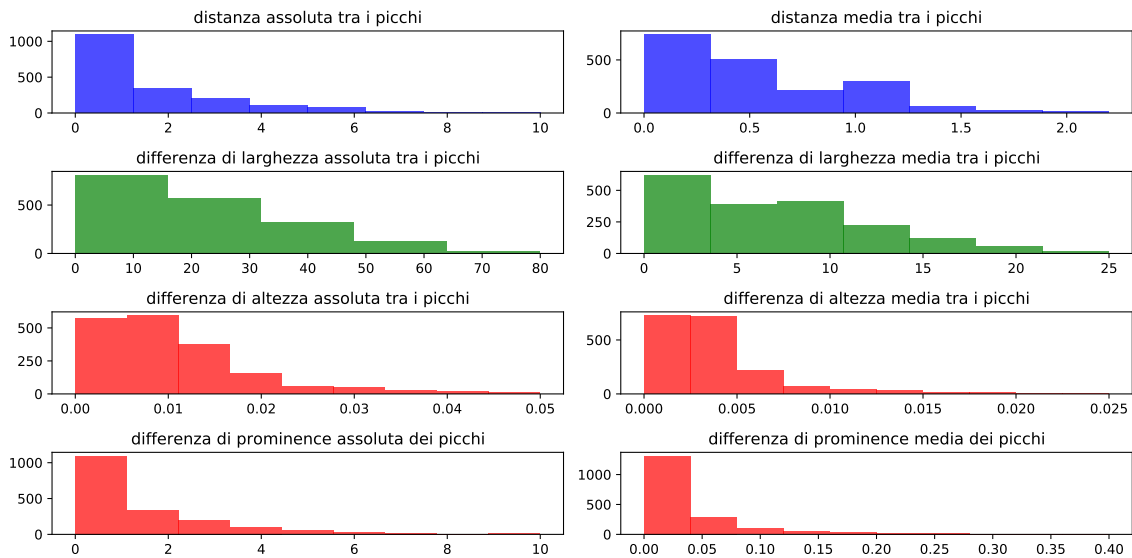


Figura 5: Istogrammi della differenza media e assoluta di altezza, distanza, larghezza e prominence.

Com'è possibile vedere dagli istogrammi, la distribuzione degli eventi per le caratteristiche dei picchi ha un massimo per il bin che contiene lo 0 e degrada procedendo verso destra, ovvero per differenze maggiori; l'andamento del primo istogramma certifica che la determinazione dei picchi è molto accurata per i ground truth predetti, mentre gli altri

istogrammi mettono in risalto il fatto che i picchi determinati, oltre ad essere nelle stesse posizioni, hanno anche le stesse caratteristiche fisiche, quali altezza, larghezza e prominence.

4.2 Analisi sulla classificazione dei picchi

È stata effettuata anche una catalogazione dei picchi degli spettri predetti: i picchi sono stati divisi tra picchi isolati, doppi, tripli e quadrupli identificati come n picchi vicini con una distanza tra gli n inferiore ad una di soglia (posta uguale a 30 perché maggiore delle più grandi fluttuazioni di larghezza). L'obiettivo era quello di valutare se, quando il numero di picchi dei ground truth predetto era uguale al numero di picchi di quelli reali (1861 casi), effettivamente il processo di denoising risultasse sufficientemente buono da far riconoscere all'algoritmo correttamente tutti i picchi isolati, doppi, tripli e quadrupli evitando casi in cui il picco dello spettro predetto sia classificato erroneamente.

Sono state considerate predizioni corrette i casi in cui sia il numero sia il tipo dei picchi predetti e reali sono risultati coincidenti e sono risultati 1823. Sono state considerate predizioni sbagliate i 177 casi in cui il numero di picchi predetti e reali è risultato diverso, dato che l'errore nella determinazione dei picchi nascondeva una predizione, da parte del modello, del segnale di baseline con qualche fluttuazione nei punti critici dello spettro (ovvero i picchi). Tra le 177 predizioni errate, ci sono 38 spettri che, pur avendo lo stesso numero di picchi sia nella predizione che nella realtà, sono erroneamente considerati errati, a causa della diversa classificazione dei picchi tra spettro predetto e reale dovuta al fatto che in certe casistiche le fluttuazioni di larghezza dei picchi (osservabili nell'istogramma verde della sezione precedente) risultavano confrontabili con la distanza scelta per la classificazione. Grazie a quest'accorgimento le predizioni corrette risultano 1861 e le predizioni errate 139 (su 2000 spettri analizzati): il modello risolve il proprio compito correttamente nel 93.05% dei casi mentre sbaglia nel restante 6.95%.

Inoltre si è voluto vedere quanto il modello fosse performante nell'identificare le varie tipologie di picchi (isolato, doppio, triplo, quadruplo) tra 5079 picchi analizzati, i risultati ottenuti sono riportati nella tabella sottostante.

tipo di picco	predizioni corrette	predizioni sbagliate	percentuale corretta [%]
isolato	4536	31	99.32
doppio	403	52	88.57
triplo	45	8	84.91
quadruplo	3	1	75.00

Questo tipo di analisi è fortemente soggetta alle fluttuazioni di larghezza dei picchi, ma ha l'obiettivo di sottolineare la maggiore difficoltà del modello di assolvere il proprio compito per spettri con più picchi vicini.

5 Conclusioni

L'obiettivo di questo progetto era quello di isolare i ground truth di spettri Raman ottenuti mediante SRS.

Il compito è stato approcciato con tecniche di Machine Learning, in particolare di Deep learning supervisionato: infatti si sono utilizzate delle reti convoluzionali per costruire un autoencoder che fosse in grado di apprendere l'andamento del segnale di fondo comune agli spettri del dataset di partenza.

I risultati ottenuti con questo approccio si sono rivelati più che soddisfacenti perché il modello ricostruisce, mediante le sue predizioni, i ground truth reali con una notevole accuratezza, la quale è certificata anche dall'analisi sui picchi degli spettri discussa nella sezione precedente.