

# Relazione Progetto sugli Scalogrammi

## Paolo Mazzitti 502042

### Introduzione

Questo progetto si concentra sull'analisi e la classificazione di scalogrammi, rappresentazioni multilivello risultanti dalla trasformazione di segnali. L'obiettivo dello studio è la distinzione dei segmenti di scalogrammi dominati da rumore rispetto a quelli caratterizzati da frequenze ben definite, identificando anche i livelli in cui emergono i valori più elevati, indicativi della potenza dello scalogramma tramite tecniche di Machine Learning.

### Fasi dello studio

Il progetto si è articolato in diverse fasi:

1. **Creazione del dataset:** Inizialmente, si sono combinati i dati provenienti dai file CSV e XML per creare un dataset unificato, utilizzando la libreria Pandas. Questo passaggio ha permesso di avere una base di dati omogenea per le successive analisi.
2. **Estrazione delle caratteristiche:** Dopo aver lavorato i dati grezzi, si è proceduto con l'estrazione delle feature ritenute più opportune per l'obiettivo del progetto. Questa fase di feature extraction ha determinato quali aspetti dei dati sono stati utilizzati nelle fasi successive.
3. **Trasformazione delle caratteristiche:** Si è applicata una trasformazione delle feature. Questo passaggio è fondamentale per ottimizzare l'efficienza dei modelli di Machine Learning e migliorarne la capacità predittiva.
4. **Valutazione di modelli e iperparametri:** Successivamente, si sono valutati diversi modelli di Machine Learning con vari iperparametri. Questa fase ha permesso di determinare le configurazioni più efficaci per la predizione basata sui dati a disposizione.
5. **Analisi delle Predizioni Finali:** Infine, si sono analizzate le predizioni generate dai modelli per trarre delle conclusioni. L'analisi finale delle predizioni generate dai modelli ha permesso di valutare la validità e l'efficacia delle scelte metodologiche e di interpretare i risultati ottenuti.

### Creazione del dataset

Il dataset si compone di una serie di scalogrammi, ciascuno costruito a 15 livelli, e di un corrispondente insieme di metadati. Questi dati sono organizzati in due formati distinti: i file CSV contengono le matrici degli scalogrammi, mentre i file XML associati forniscono i metadati essenziali. I metadati includono informazioni come la lunghezza dello scalogramma, il numero di livelli e dettagli importanti relativi a specifiche sezioni dello scalogramma.

In particolare, nei file XML sono presenti degli elementi denominati "*bounding-box*", che svolgono un ruolo cruciale nell'analisi. Questi bounding-box identificano sezioni specifiche all'interno degli scalogrammi, contrassegnate con etichette quali "*fragment*", "*noise*", o "*empty*". Le etichette "*fragment*" si riferiscono a segmenti con frequenze definite, nonostante la presenza di un certo grado di rumore. Al contrario, "*noise*" indica segmenti caratterizzati esclusivamente da rumore, mentre "*empty*" designa segmenti con segnale nullo.

Oltre a ciò, i bounding-box forniscono dettagli come gli indici temporali di inizio e fine di ciascun segmento e i livelli interessati. Queste ultime informazioni sono fondamentali per i segmenti etichettati come "*fragment*", poiché indicano i livelli in cui il segnale raggiunge la massima potenza, offrendo così un'importante chiave di lettura per l'analisi degli scalogrammi.

## Estrazione delle caratteristiche

Nel corso della fase di estrazione delle caratteristiche (*feature extraction*), si è proceduto alla selezione di un particolare insieme di feature, mirate specificamente per gli specifici compiti di classificazione di questo studio. Tali feature sono state scelte in ragione della loro capacità di rappresentare varie proprietà essenziali dei segnali.

Le feature selezionate comprendono:

1. **Media (Mean):** Indica il valore centrale del segmento di segnale. La media è utile per identificare il livello generale di intensità del segnale, fornendo un punto di riferimento per confrontare varie porzioni del segnale.
2. **Deviazione Standard (Std):** Misura la dispersione dei valori del segnale rispetto alla media. Una deviazione standard elevata indica una maggiore varianza nel segnale, suggerendo che i valori del segnale sono distribuiti su un intervallo più ampio.
3. **Deviazione Media Assoluta (MAD):** Rappresenta la media delle differenze assolute tra valori consecutivi nel segmento. Questa misura è particolarmente efficace nel catturare la variabilità locale all'interno del segnale, offrendo una visione dettagliata della stabilità o della volatilità del segnale.
4. **Deviazione Quadratica Media (RMSD):** Fornisce una misura della varianza nel segnale, enfatizzando le variazioni maggiori. A differenza della MAD, la RMSD tende a dare un peso maggiore alle variazioni più ampie nel segnale.
5. **Somma delle Differenze Assolute (SAD):** Calcola la somma totale delle differenze assolute tra valori consecutivi.
6. **Coefficiente di Variazione (CoV):** Si tratta di una misura normalizzata della dispersione dei dati, che si ottiene dividendo la deviazione standard per la media del segmento. Il CoV è utile per confrontare la variabilità tra segmenti di segnale con medie diverse, offrendo una visione proporzionata della dispersione dei dati.

## Trasformazione delle caratteristiche

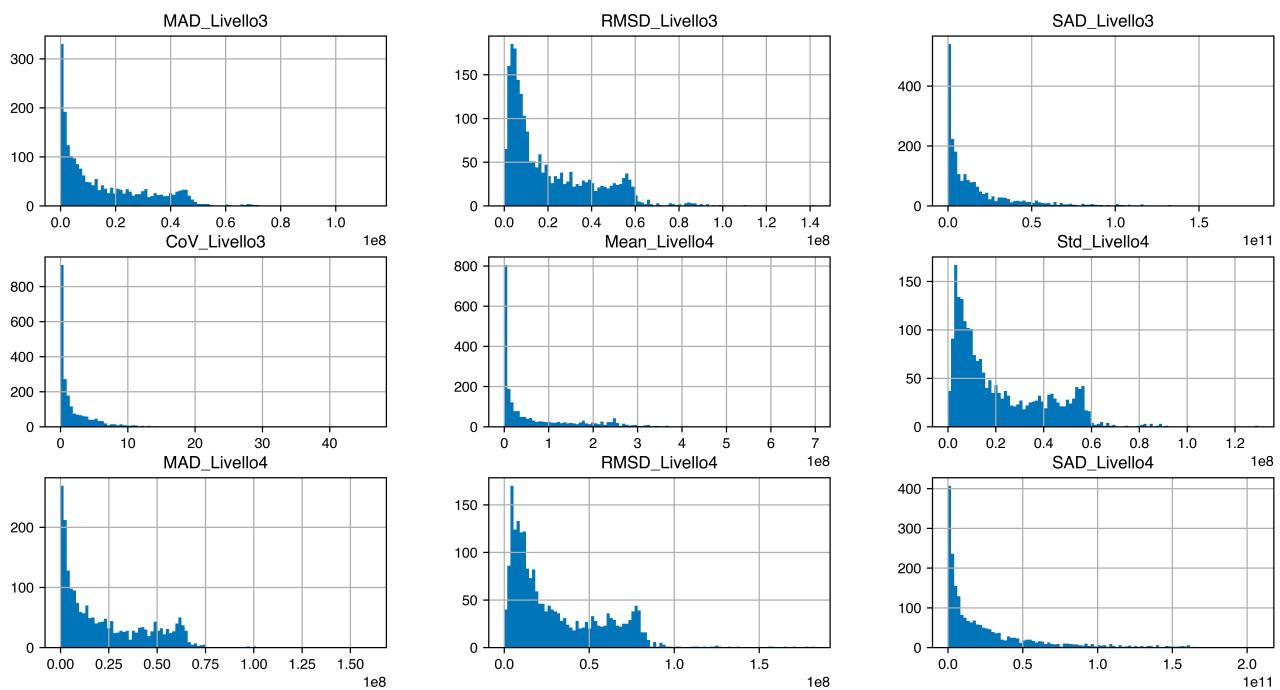
Nella fase di trasformazione e riduzione della dimensionalità, l'analisi preliminare ha incluso la generazione di grafici per esaminare la distribuzione dei dati. Questi grafici hanno rivelato una distribuzione fortemente asimmetrica dei dati, un fenomeno noto come skewness. Al fine di affrontare e mitigare questa caratteristica, che può influenzare negativamente l'efficacia dei modelli di Machine Learning, è stato sviluppato un approccio personalizzato.

Per ridurre l'asimmetria dei dati, è stata creata una classe personalizzata estendendo le classi `BaseEstimator` e `TransformerMixin` di Scikit-learn. Questo nuovo trasformatore implementa la trasformazione dei dati attraverso l'applicazione della radice quadrata o della radice cubica, a seconda dei casi specifici, per normalizzare la distribuzione dei dati e ridurne l'asimmetria.

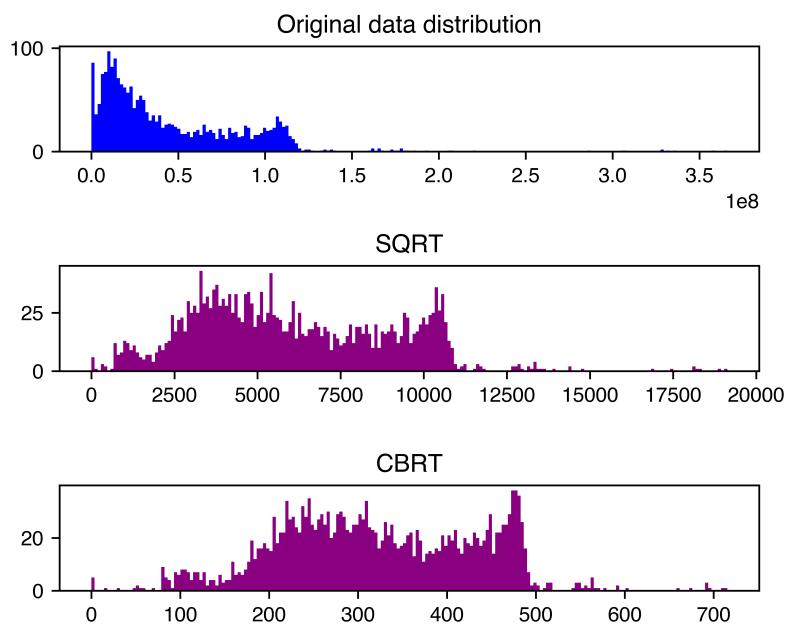
Successivamente, per assicurare che i dati trasformati fossero adeguatamente scalati per l'analisi, è stati valutati diversi metodi di standardizzazione (*Standard Scaler*, *Robust Scaler*, *Quantile*

*Transformer*), per normalizzare i dati rispetto alla media e alla deviazione standard. Questo passaggio è cruciale per garantire che le varie feature abbiano lo stesso peso nei modelli di Machine Learning, evitando che caratteristiche con ampi intervalli di variazione dominino il processo di apprendimento.

La selezione dell'operazione di trasformazione più efficace non è stata determinata a priori, ma attraverso un processo empirico. Creando una pipeline che incorpora diverse configurazioni di iperparametri e trasformazioni dei dati, è stato possibile valutare vari scenari in modo sistematico. Alla fine di questo processo, la scelta della trasformazione ottimale si basa su evidenze concrete derivanti dai risultati ottenuti, garantendo che la decisione finale sia informata dalle caratteristiche specifiche dei dati e dalle esigenze del modello.



*Distribuzione di alcune caratteristiche*



*Didascalia*

## Selezione dei modelli e iperparametri

Nella fase di valutazione dei modelli, si è optato per l'impiego di diversi algoritmi di Machine Learning disponibili nella libreria Scikit-learn, tra cui Support Vector Classifier (SVC), K-Neighbors Classifier (KNC), Random Forest, AdaBoost e Gradient Boosting Classifier. Per identificare la configurazione ottimale degli iperparametri per ciascuno di questi modelli, è stata utilizzata la tecnica di Halving Random Search.

Questo metodo consente una ricerca efficiente nello spazio degli iperparametri attraverso una selezione iterativa che riduce il pool di candidati ad ogni passaggio, basandosi su una frazione ridotta del dataset di training e incrementando successivamente la dimensione del dataset valutato per le configurazioni più promettenti.

Parallelamente, per garantire la generalizzabilità dei modelli e una stima affidabile delle loro prestazioni, è stata applicata la cross validation. Questo processo prevede la divisione del dataset in diverse parti, alternando il training e il test su ciascuna di queste, per assicurare che ogni segmento del dataset venga utilizzato sia per l'addestramento che per la validazione. Tale approccio aiuta a minimizzare il rischio di overfitting e a fornire una valutazione robusta delle capacità predittive dei modelli su dati non visti.

L'impiego congiunto di Halving Random Search e cross validation ha permesso di determinare con precisione le migliori configurazioni di iperparametri per i modelli analizzati, massimizzando così l'efficacia della classificazione degli scalogrammi nel contesto del progetto.

Per l'addestramento e la valutazione dei modelli, è stata privilegiata l'opzione di addestrare un classificatore specifico per ogni target, piuttosto che adottare la strategia del MultiOutput Classifier. Quest'ultimo, offerto da Scikit-learn, permette di gestire più output simultaneamente attraverso un unico modello, addestrando un classificatore indipendente per ciascuna label. Sebbene questa soluzione possa semplificare l'addestramento su task multi-label, può risultare meno precisa a causa delle diverse esigenze di ciascun target in termini di modellazione e ottimizzazione degli iperparametri.

Optando per classificatori dedicati, si mira a una personalizzazione maggiore, adeguando le tecniche e le configurazioni ai singoli target per ottenere prestazioni ottimali. Questa decisione è stata guidata dalla necessità di massimizzare l'accuratezza della classificazione in un contesto caratterizzato da elevata variabilità e complessità dei dati.

Di seguito sono riportati alcune configurazioni ottimali registrate durante lo studio del progetto per i diversi target della classificazione.

Target	Kernel	Degree	C	Scaler	Root Transformer
Label	Polynomial	3	4.1	StandardScaler	None
YMin	Linear	1	3.4	StandardScaler	None
YMax	Linear	3	0.3	StandardScaler	None

SVC

Target	Metric	Neighbors	Weight	Scaler	Root Transformer
Label	Euclidean	10	distance	StandardScaler	None
YMin	Euclidean	4	distance	StandardScaler	None
YMax	Euclidean	6	distance	StandardScaler	None

KNC

Target	Criterion	Scaler	Root Transformer
Label	Entropy	QuantileTransformer	CBRT
YMin	Gini	RobustScaler	CBRT
YMax	Gini	RobustScaler	None

Random Forest

Target	N.estimators	Max Depth Decision Tree	Scaler	Root Transformer
Label	100	2	RobustScaler	None
YMin	100	2	RobustScaler	None
YMax	100	2	RobustScaler	None

AdaBoost

Target	L2 reg.	Scaler	Root Transformer
Label	0.3	StandardScaler	SQRT
YMin	0.2	RobustScaler	SQRT
YMax	0.2	StandardScaler	SQRT

Hist Gradient Boosting

Di seguito viene mostrata l'accuratezza ottenuta dalle migliori configurazioni dei modelli utilizzati per ogni task di classificazione.

Target	SVC	KNC	Random Forest	AdaBoost	Hist Gradient Boosting
Label	0.90	0.85	0.94	0.91	0.93
YMin	0.87	0.77	0.89	0.55	0.88
YMax	0.83	0.77	0.86	0.64	0.86

Accuratezza per ogni modello

## Valutazione dei modelli

Nella fase di valutazione dei modelli di classificazione, è stato selezionato come modello di riferimento l'HistGradientBoosting, sulla base delle sue prestazioni complessive. Per una valutazione approfondita delle sue capacità di classificazione, sono state adottate tre metriche principali: recall, precision e F1 score.

Al fine di esplorare la relazione tra precision e recall, sono stati generati due grafici chiave. Il primo grafico pone in relazione diretta la precision con la recall, fornendo una rappresentazione visiva di come queste due metriche interagiscano tra loro per il modello selezionato (Figura 1).

Il secondo grafico affronta il trade-off tra precision e recall attraverso l'analisi della funzione di decisione fornita da Scikit-learn. Questa visualizzazione dimostra come l'aggiustamento della soglia di decisione influenzi direttamente il bilanciamento tra aumentare la precision e aumentare la recall. Tale analisi è fondamentale per comprendere le implicazioni pratiche della scelta di una soglia più alta o più bassa, in base alle specifiche esigenze del contesto applicativo, dove potrebbe essere preferito privilegiare l'una o l'altra metrica. (Figura 2)

In aggiunta alle analisi basate su metriche quali precision, recall e F1 score, è stata condotta un'ulteriore valutazione delle prestazioni del modello Hist Gradient Boosting mediante l'utilizzo di un ConfusionMatrixDisplay. Questo strumento grafico ha giocato un ruolo cruciale nell'analizzare la capacità del modello di classificare correttamente i dati di test, permettendoci di visualizzare in maniera immediata e intuitiva le accuratezze e le misclassificazioni per ciascuna classe. Il ConfusionMatrixDisplay presenta una matrice di confusione, dove ogni riga della matrice rappresenta le istanze della classe reale, mentre ogni colonna rappresenta le istanze della classe predetta dal modello. I valori lungo la diagonale principale della matrice indicano il numero di classificazioni corrette per ogni classe, mentre i valori fuori dalla diagonale mostrano le misclassificazioni, cioè i casi in cui il modello ha erroneamente assegnato una classe diversa da quella reale.

## Conclusioni

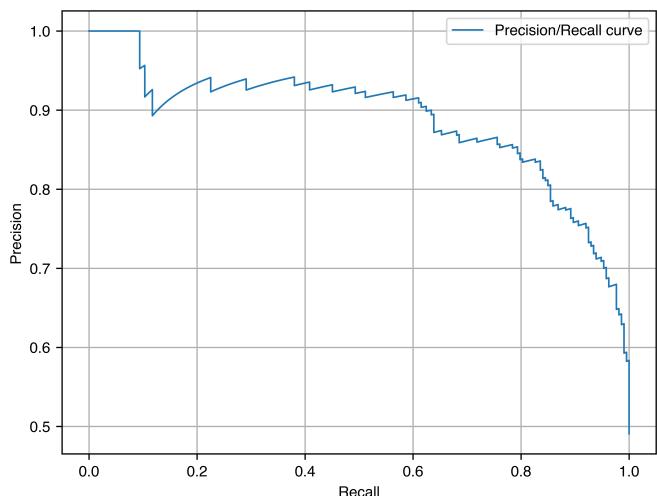


Figura 1

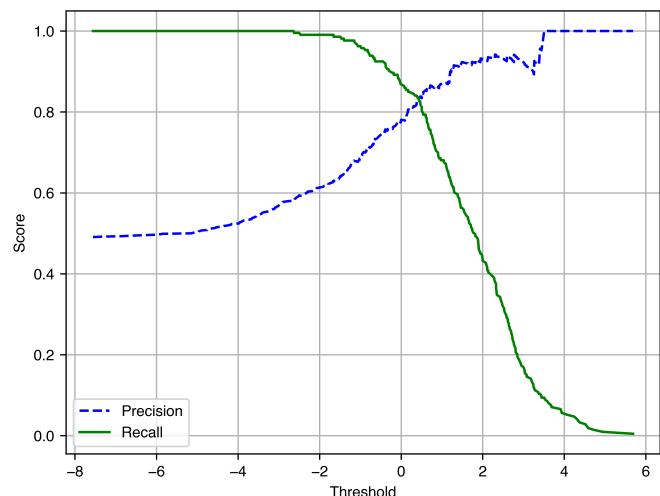
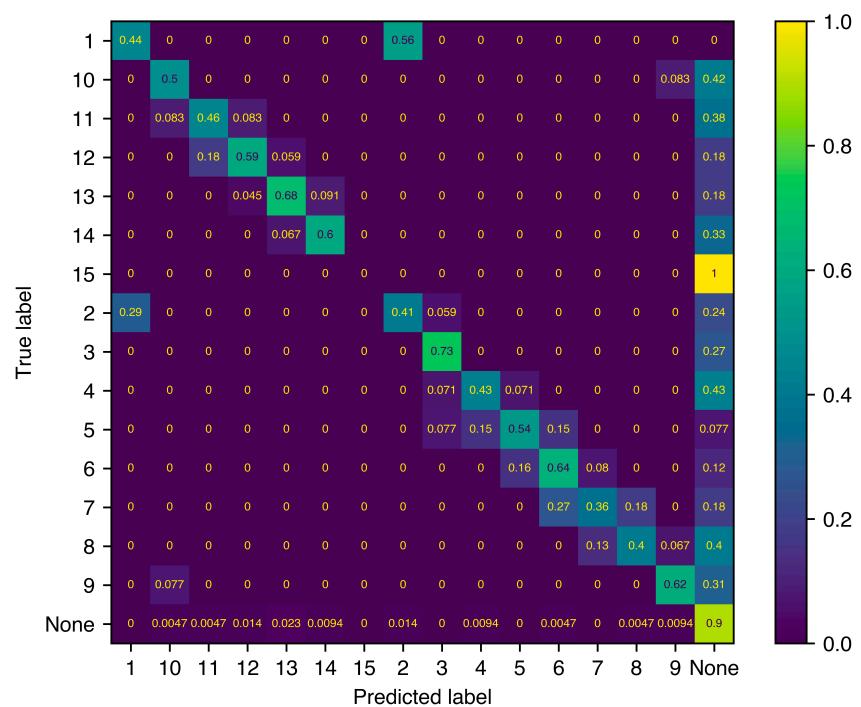
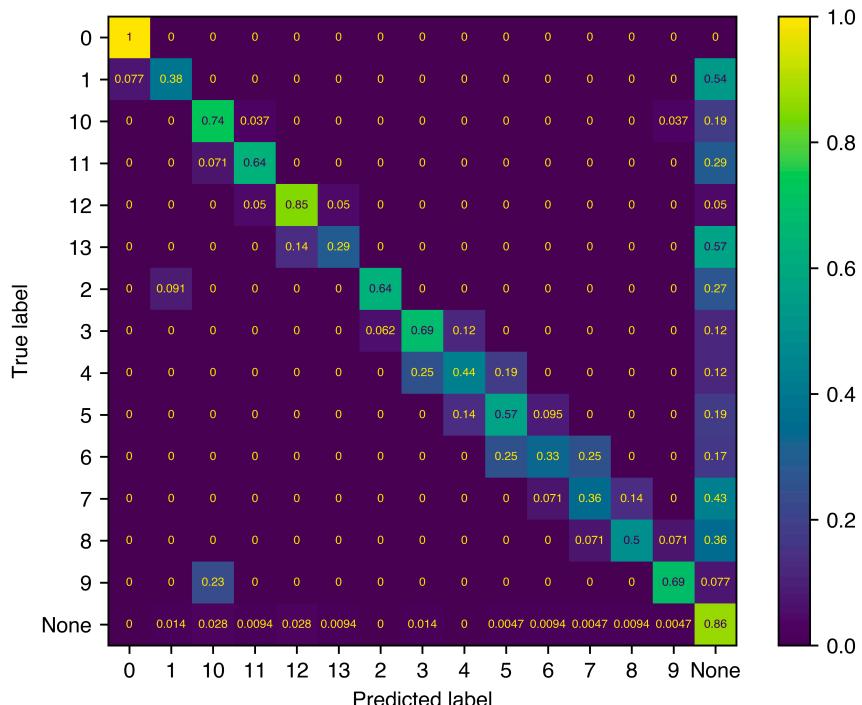
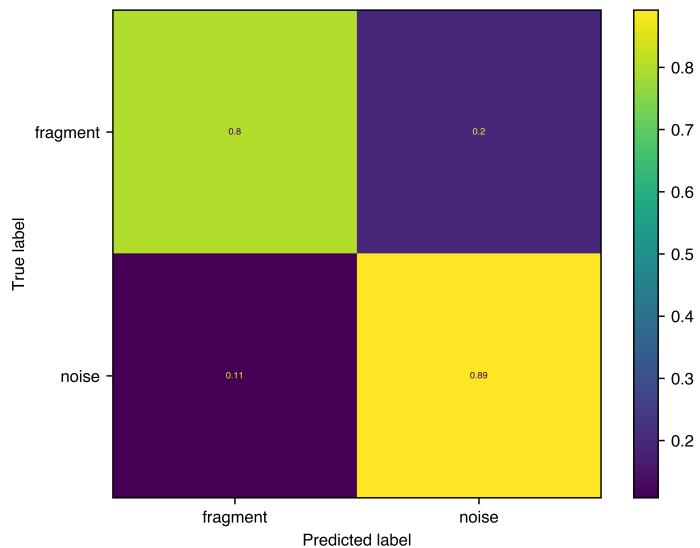


Figura 2



Dall'analisi delle matrici di confusione relative ai diversi modelli di machine learning testati, è emerso un pattern di errore significativo, particolarmente nelle previsioni riguardanti la definizione della frequenza minima e massima. Un numero notevole di campioni, che avrebbero dovuto essere classificati in specifiche categorie di frequenza, sono stati erroneamente identificati come rumore.

Dalle analisi condotte invece sui vari modelli di machine learning, è emerso che K-Nearest Neighbors (KNC) e Support Vector Machines (SVC) si distinguono per la loro velocità di addestramento, consentendo l'esplorazione di uno spazio di configurazione degli iperparametri più ampio in tempi relativamente brevi. Tuttavia, sebbene vantaggiosi in termini di velocità, questi modelli tendono a offrire una precisione inferiore rispetto agli algoritmi ensemble come Random Forest, AdaBoost e Gradient Boosting, i quali, nonostante richiedano tempi di addestramento maggiori, hanno dimostrato di garantire livelli di accuratezza superiori.

In particolare, AdaBoost ha mostrato miglioramenti significativi nelle prestazioni solo incrementando il numero di estimatori e la profondità degli alberi di decisione utilizzati come base. Questo approccio, però, ha come conseguenza un aumento esponenziale dei tempi di addestramento, sottolineando un trade-off critico tra accuratezza e efficienza computazionale.

Nonostante queste considerazioni, il Gradient Boosting si è rivelato essere il modello con la maggiore accuratezza complessiva, evidenziando come l'ottimizzazione sequenziale degli estimatori possa portare a prestazioni notevolmente elevate, sebbene a costo di una maggiore complessità computazionale e temporale.

Ai fini di questo studio, l'analisi si è concentrata esclusivamente su tecniche di Machine Learning convenzionali, lasciando spazio alla possibilità di esplorare approcci alternativi che potrebbero offrire un equilibrio ottimale tra accuratezza e velocità di esecuzione. In particolare, l'impiego di librerie più efficienti come XGBoost o CatBoost, noti per la loro capacità di eseguire algoritmi di boosting in maniera più rapida ed efficace, potrebbe rappresentare un'evoluzione naturale di questa ricerca. Allo stesso modo si potrebbero adottare delle reti neurali convolutive, particolarmente adatte al trattamento di dati strutturati in forma di immagini o sequenze