CHAPTER II Application of Algorithms for Instrument Classification

II.1 Creazione degli algoritmi in linguaggio Python

Come anticipato nell'introduzione del capitolo, la fase di implementazione si è concentrata sulla costruzione di tre modelli di classificazione supervisionata: Random Forest, XGBoost e una rete neurale convoluzionale (CNN), utilizzando il linguaggio di programmazione *Python* e alcune delle sue librerie più consolidate nel campo del machine learning, come scikit-learn, XGBoost e PyTorch.

Questa triplice strategia ha permesso di confrontare approcci differenti: modelli basati su alberi decisionali da un lato, e tecniche di deep learning ispirate alla computer vision dall'altro.

II.1.1 Caricamento e Preprocessing dei dati

Il dataset iniziale era composto da registrazioni audio di diversi strumenti musicali. In una prima fase di preprocessing, i file sono stati puliti per rimuovere rumori di fondo o silenzi non significativi e successivamente suddivisi in sequenze di *10 secondi* ciascuna, ottenendo circa 120 tracce per ogni strumento. Gli strumenti considerati sono stati: chitarra, violino, pianoforte, viola e flauto.

A partire da questi file audio sono stati generati due differenti tipi di dati, in funzione del modello da addestrare:

- Per Random Forest e XGBoost: sono state estratte feature numeriche sintetiche.
- Per la rete neurale convoluzionale: sono stati creati spettrogrammi salvati come immagini.

Estrazione delle feature numeriche

Per i modelli Random Forest e XGBoost, l'obiettivo è stato quello di estrarre da ogni traccia un insieme di feature che rappresentassero in modo compatto e informativo le caratteristiche principali del suono, sia dal punto di vista timbrico, armonico che ritmico. Le feature calcolate sono state:

- MFCC (Mel-Frequency Cepstral Coefficients): catturano la forma dello spettro sonoro su una scala di frequenze che rispecchia la percezione umana. Sono fondamentali per descrivere il timbro degli strumenti.
- Chroma Features: rappresentano l'energia associata alle 12 classi di pitch all'interno di un'ottava, permettendo di caratterizzare la componente armonica del suono.
- Spectral Contrast: misura il contrasto tra picchi e valli dello spettro in diverse bande di frequenza, utile per distinguere suoni armonici da quelli più rumorosi o percussivi.
- Tonnetz (Tonal Centroid Features): mappa le relazioni armoniche tra note, aiutando a cogliere la tonalità o il "colore" armonico del segnale.

- Zero Crossing Rate (ZCR): conta il numero di volte in cui il segnale audio attraversa lo zero. È indicativo della rumorosità o della presenza di suoni percussivi.
- Root Mean Square Energy (RMS): misura l'intensità media del segnale audio.
- Tempo: stima la velocità ritmica del brano, espressa in BPM (battiti per minuto).

Questa selezione di feature è stata pensata per offrire una rappresentazione completa e robusta del contenuto musicale, aumentando le possibilità di distinguere correttamente strumenti anche molto simili tra loro.

Per garantire l'efficacia dei modelli:

- Le etichette di classe sono state codificate numericamente usando LabelEncoder.
- Le feature numeriche sono state standardizzate con StandardScaler, portandole ad avere media 0 e deviazione standard 1, operazione fondamentale per modelli che fanno uso di metriche di distanza o somiglianza tra feature.

Inoltre, una prima analisi esplorativa è stata condotta mediante una PCA (Principal Component Analysis), riducendo il dataset a 2 componenti principali per visualizzare graficamente la distribuzione e separabilità delle classi.

Creazione degli spettrogrammi per la CNN

Per addestrare la rete neurale convoluzionale, invece, si è scelto di rappresentare ogni file audio come un'immagine bidimensionale, trasformandolo in uno *spettrogramma*. Questa scelta è motivata dal fatto che un file audio, quando analizzato nel dominio *tempo-frequenza*, rivela pattern spaziali molto simili a quelli delle immagini, rendendo naturale l'applicazione di tecniche di deep learning sviluppate per la computer vision.

Il processo di creazione degli spettrogrammi ha seguito questi passaggi:

- Caricamento dell'audio: ogni file .wav è stato letto insieme alla sua frequenza di campionamento.
- Calcolo del Mel Spectrogram: utilizzando la scala Mel, si ottiene una rappresentazione in cui l'energia del segnale è distribuita in modo più fedele alla percezione umana.
- Conversione in scala logaritmica: lo spettrogramma è stato trasformato in decibel (dB), per enfatizzare meglio le variazioni di intensità anche nei segnali deboli.
- Visualizzazione e salvataggio: ogni spettrogramma è stato salvato come immagine in scala di grigi (grayscale), senza assi o etichette, per concentrarsi esclusivamente sulle informazioni visive del segnale.

Questa trasformazione ha permesso di "tradurre" il problema della classificazione audio in un problema di classificazione di immagini, sfruttando appieno le potenzialità delle CNN nel riconoscere pattern complessi, come armoniche, attacchi o variazioni ritmiche.

Tra i principali vantaggi di questa tecnica:

- Conservazione della struttura armonica e temporale del segnale.
- Evidenziazione di pattern specifici per ogni strumento.

- Possibilità di utilizzare le architetture più avanzate di deep learning nate per la computer vision.

I dati, sia quelli numerici sia quelli visivi, sono stati organizzati in tre classici set:

- Training set: per addestrare i modelli.
- Validation set: per la validazione incrociata e la selezione degli iperparametri.
- Test set: per la valutazione finale delle prestazioni.

La divisione dei dati ha mantenuto la distribuzione equilibrata delle classi, in modo da garantire modelli il più possibile generalizzabili.

II.1.2 Creazione dei modelli Random Forest e XGBoost

Modello XGBoost

La costruzione del modello XGBoost ha previsto una fase iniziale di *ottimizzazione degli iperparametri* mediante GridSearchCV. Le combinazioni esplorate hanno riguardato:

- Numero di alberi (n_estimators): 100, 200, 500
- Profondità massima degli alberi (max_depth): 3, 5, 7
- Tasso di apprendimento (learning_rate): 0.01, 0.05, 0.1

La ricerca è stata effettuata con una validazione incrociata a 3 fold, adottando come metrica di riferimento l'accuratezza. Il miglior set di parametri è stato quindi selezionato per l'addestramento finale del modello.

Per valutare la robustezza del classificatore, è stata successivamente applicata una cross-validation stratificata a 5 fold sull'insieme di training, permettendo di stimare l'affidabilità delle prestazioni su dati non visti.

Il modello ottimizzato è stato poi valutato sia sull'insieme di validation sia su quello di test, calcolando le principali metriche di classificazione e analizzando la matrice di confusione associata.

Modello Random Forest

In parallelo, è stato sviluppato anche un modello basato su Random Forest, seguendo un approccio analogo. La fase di ricerca degli iperparametri ha coinvolto:

- Numero di alberi (n_estimators): 100, 200, 500
- Profondità massima degli alberi (max_depth): None (illimitata), 10, 20

Anche in questo caso, l'ottimizzazione è stata effettuata tramite GridSearchCV con validazione a 3 fold. Una volta individuati i parametri ottimali, il modello finale è stato validato con una cross-validation stratificata a 5 fold.

I risultati sulle predizioni sugli insiemi di validation e test sono stati analizzati attraverso l'accuratezza, i report di classificazione e le matrici di confusione, consentendo un confronto diretto con il modello XGBoost.

Modello CNN

La rete neurale convoluzionale (CNN) sviluppata è stata progettata con l'obiettivo di ottenere una struttura compatta ed efficace nell'estrazione delle caratteristiche più significative delle immagini in input. L'architettura si articola in tre blocchi principali, ciascuno costituito da un'operazione di convoluzione bidimensionale, seguita da una Batch Normalization e dall'applicazione della funzione di attivazione ReLU. Procedendo nei livelli della rete, il numero di filtri viene incrementato progressivamente (32, 64 e 128 canali), consentendo così al modello di costruire rappresentazioni gerarchiche via via più complesse e astratte.

A valle di ogni blocco convoluzionale, viene inserito un livello di Max Pooling con finestra 2×2 e stride 2, con l'obiettivo di ridurre la dimensione spaziale delle mappe di attivazione e di favorire una migliore generalizzazione. Una volta completata la fase convoluzionale, l'output tridimensionale viene appiattito in un vettore monodimensionale, che costituisce l'input per la successiva fase di classificazione. In questa fase, un primo strato completamente connesso (fc1) riduce la dimensionalità a 512 unità e impiega una funzione di attivazione ReLU, accompagnata da un meccanismo di dropout al 50% per contrastare il rischio di overfitting. A conclusione del percorso, un secondo strato fully connected (fc2) produce l'output finale, costituito da cinque neuroni, uno per ciascuna classe di strumenti musicali da riconoscere.

Il modello è stato progettato per elaborare immagini in scala di grigi, caratterizzate dalla presenza di un solo canale, in linea con le specifiche del dataset considerato.

Il processo di addestramento della rete è stato impostato seguendo un paradigma di apprendimento supervisionato. Per l'ottimizzazione dei pesi è stato scelto l'algoritmo Adam, noto per la sua capacità di adattare dinamicamente il learning rate durante l'addestramento, mentre come funzione obiettivo è stata adottata la Cross Entropy Loss, particolarmente adatta ai problemi di classificazione multiclasse.

Durante il training, al termine di ogni epoca, il modello è stato valutato sull'insieme di validazione, monitorando costantemente sia la funzione di perdita sia l'accuratezza. Tali metriche sono state registrate in modo sistematico, consentendo di tracciare nel tempo l'andamento delle prestazioni sia in fase di training che di validation.

Per limitare il rischio di overfitting, è stata implementata una strategia di early stopping: qualora l'accuratezza di validazione non migliorasse per cinque epoche consecutive, l'addestramento veniva interrotto anticipatamente. Il modello che ha ottenuto la migliore accuratezza in fase di validazione è stato salvato e successivamente utilizzato per la fase di test.

Infine, tutte le metriche relative all'addestramento sono state archiviate in un file CSV per consentire un'analisi successiva più dettagliata. È stato inoltre prodotto un grafico riepilogativo che illustra l'andamento della loss e dell'accuratezza nel corso delle epoche, permettendo una valutazione visiva dell'evoluzione del processo di apprendimento.

II.2 Risultati ottenuti

Dopo aver completato la fase di progettazione e addestramento dei tre modelli di classificazione, Random Forest, XGBoost e rete neurale convoluzionale (CNN), si è proceduto alla valutazione sistematica delle loro prestazioni sui rispettivi set di test. L'obiettivo è stato quello di confrontare l'efficacia dei diversi approcci, evidenziandone i punti di forza e le eventuali criticità nel compito di classificazione degli strumenti musicali.

Per ciascun modello sono state calcolate le principali metriche di valutazione, accuratezza, precisione, richiamo (recall) e misura F1 (F1-score), affiancate dall'analisi delle rispettive matrici di confusione.

Inoltre, per la CNN, è stato monitorato anche l'andamento della loss e dell'accuratezza durante l'addestramento, al fine di valutare il comportamento dinamico del modello.

Nelle sezioni seguenti vengono presentati i risultati ottenuti, seguiti da un confronto critico delle differenze emerse.

Per il modello XGBoost, l'ottimizzazione degli iperparametri ha portato alla selezione dei valori: learning_rate = 0.1, max_depth = 5, e n_estimators = 100.

I risultati di classificazione, riassunti nella tabella seguente, mostrano un comportamento eterogeneo a seconda delle classi:

label	precision	recall	F1-score
chitarra	0.83	0.56	0.67
fluato	0.71	0.67	0.69
pianoforte	0.80	1.00	0.89
viola	0.65	0.55	0.59
violino	0.57	0.72	0.63

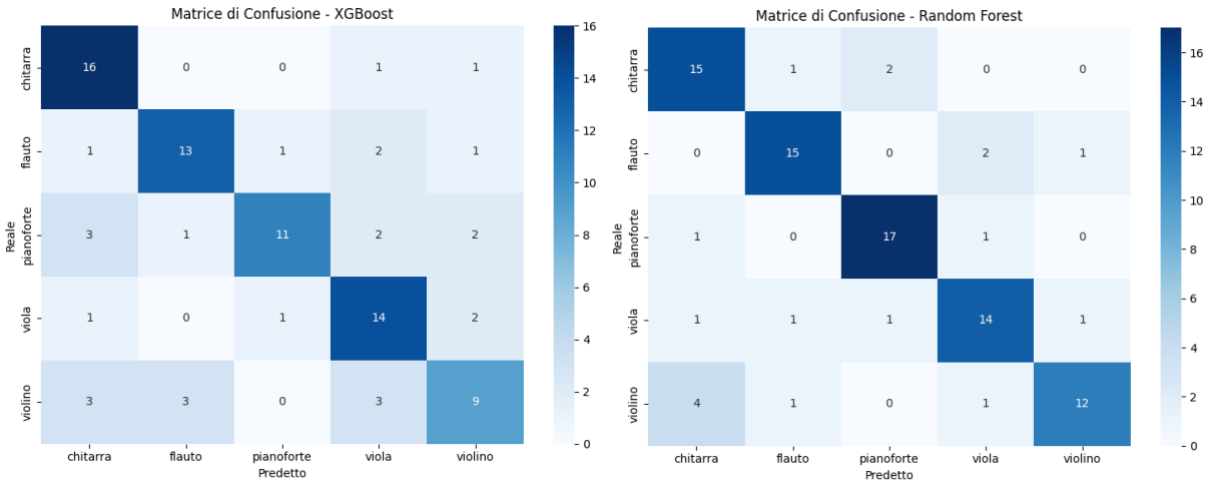
L'accuratezza media del modello si attesta intorno a 0.71, un risultato soddisfacente considerando la difficoltà del compito su cinque classi differenti. Anche sul set di addestramento, il modello ha confermato prestazioni simili, con un'accuratezza media di circa 0.70.

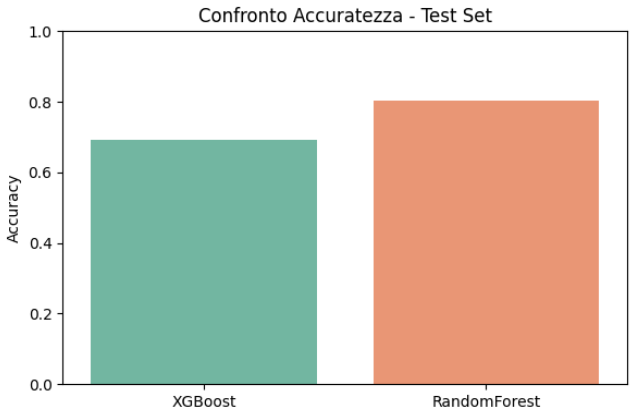
Diversamente da XGBoost, il modello Random Forest ha ottenuto le migliori prestazioni con n_estimators = 200 e senza limitazioni sulla profondità (max_depth = None). I risultati della classificazione sono i seguenti:

label	precision	recall	F1-score
chitarra	0.88	0.83	0.86
fluato	0.82	0.78	0.80
pianoforte	0.87	1.00	0.93
viola	0.89	0.85	0.87
violino	0.83	0.83	0.83

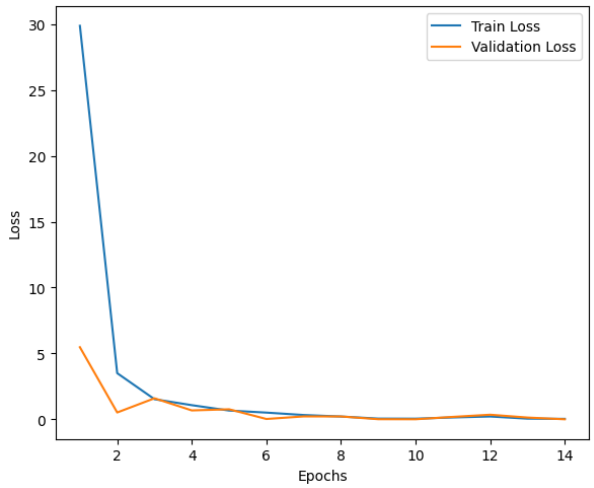
In media, il modello raggiunge un'accuratezza di circa 0.87, superiore a quella ottenuta con XGBoost. Tuttavia, sul set di test si osserva un lieve calo nelle prestazioni, con un'accuratezza che si stabilizza intorno a 0.81.

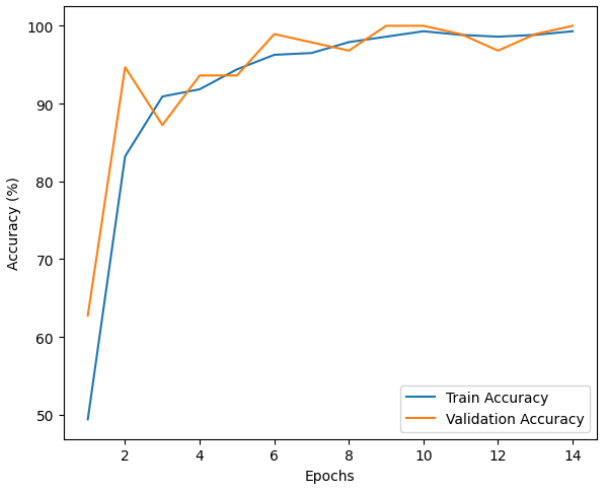
Una rappresentazione grafica delle matrici di confusione per entrambi i modelli è riportata nella seguente figura, così come un confronto diretto delle accuratezze, da cui emerge chiaramente il vantaggio del modello Random Forest su XGBoost.





La rete neurale convoluzionale (CNN) ha completato il processo di addestramento in 14 epoche, raggiungendo valori ottimali di train_loss pari a 0.0332 e val_loss pari a 0.00. Sia l'accuratezza sul training set che quella sul validation set hanno raggiunto il 100%. I grafici riportanti l'andamento della loss e dell'accuratezza durante l'addestramento sono presentati nelle figure seguenti.

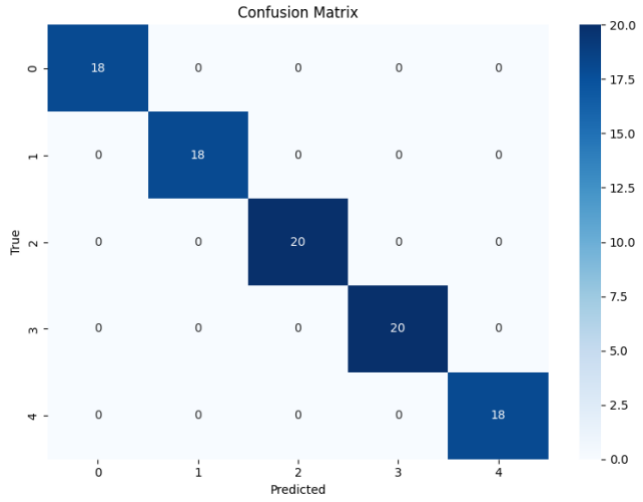




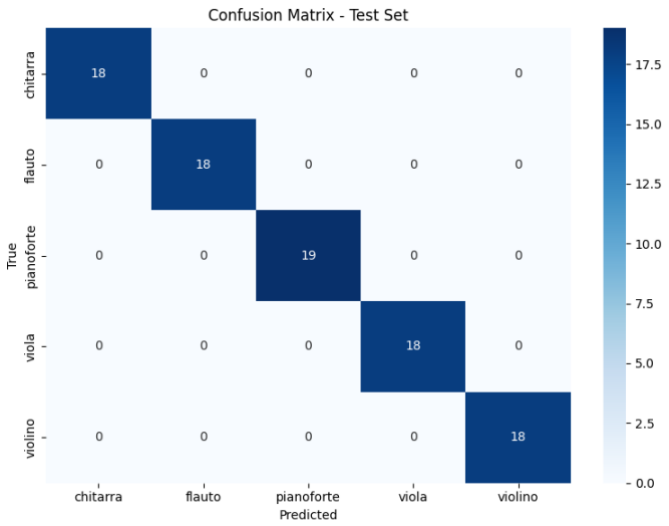
Il report di classificazione conferma la perfetta distinzione tra tutte le classi:

label	precision	recall	F1-score
chitarra	1.00	1.00	1.00
fluato	1.00	1.00	1.00
pianoforte	1.00	1.00	1.00
viola	1.00	1.00	1.00
violino	1.00	1.00	1.00

La matrice di confusione conferma ulteriormente la capacità del modello di classificare correttamente ogni strumento musicale, senza errori:



Anche durante la fase di test finale, utilizzando il miglior modello salvato, tutte le metriche hanno confermato un'accuratezza del 100%:



Da questa notiamo che l'algoritmo riesce a classificare sempre correttamente gli strumenti musicali.

Dal confronto complessivo emerge chiaramente come la rete neurale convoluzionale si distingua nettamente in termini di prestazioni, ottenendo una classificazione perfetta.

Tuttavia, è importante sottolineare che modelli più tradizionali come Random Forest e XGBoost, pur mostrando prestazioni inferiori, mantengono comunque un'elevata accuratezza e possono rappresentare valide alternative in contesti dove risorse computazionali limitate o esigenze di interpretabilità rendano preferibile l'utilizzo di modelli più semplici.

Il Random Forest, in particolare, ha mostrato una robustezza maggiore rispetto a XGBoost, risultando più stabile sia sul training che sul test set.

La CNN, grazie alla ricchezza informativa contenuta negli spettrogrammi e alla capacità delle architetture convoluzionali di cogliere pattern complessi, si è rivelata estremamente efficace per questo tipo di problema.

II.3 Considerazione finali

Il lavoro svolto ha permesso di esplorare e confrontare tre approcci differenti alla classificazione di strumenti musicali a partire da rappresentazioni spettrali del suono. La progettazione e la valutazione dei modelli Random Forest, XGBoost e rete neurale convoluzionale hanno fornito una panoramica ampia sulle potenzialità e sui limiti delle diverse tecniche di machine learning applicate a dati audio.

I modelli basati su algoritmi tradizionali di apprendimento supervisionato, come Random Forest e XGBoost, si sono dimostrati validi, in particolare il primo, che ha raggiunto prestazioni notevoli con un'accuratezza test di circa il 81%. Questi approcci risultano particolarmente vantaggiosi per la loro interpretabilità e per il minor costo computazionale rispetto a modelli più complessi. Tuttavia, si è osservato che la capacità di generalizzazione su dati nuovi è limitata rispetto a quanto ottenuto con la rete neurale convoluzionale.

La CNN ha mostrato prestazioni straordinarie, riuscendo a classificare perfettamente tutte le classi di strumenti musicali sia in fase di training che di test. Questo risultato evidenzia l'efficacia delle architetture profonde, soprattutto quando si opera su dati che contengono pattern strutturali ricchi come quelli degli spettrogrammi audio. Le reti convoluzionali, infatti, sono particolarmente adatte a catturare correlazioni spaziali e frequenziali che risultano fondamentali per discriminare tra le diverse timbriche degli strumenti.

Tuttavia, se da un lato l'eccellente accuratezza della CNN testimonia la validità dell'approccio, dall'altro solleva interrogativi legati alla possibilità di overfitting. Nonostante la validazione e i test abbiano restituito risultati perfetti, in applicazioni reali, dove i dati possono presentare rumore, variazioni ambientali o registrazioni di qualità inferiore, potrebbe essere necessario introdurre tecniche di regolarizzazione più spinte o strategie di data augmentation per migliorare ulteriormente la robustezza del modello.

Un'altra osservazione importante riguarda il trade-off tra complessità del modello e prestazioni. Sebbene la CNN offra la massima accuratezza, richiede tempi di addestramento significativamente più lunghi e risorse computazionali elevate, che potrebbero non essere sempre disponibili in contesti applicativi pratici. In scenari in cui efficienza e rapidità siano prioritari rispetto alla precisione assoluta, modelli come Random Forest potrebbero rappresentare una scelta più appropriata.

Dal punto di vista metodologico, il lavoro ha evidenziato l'importanza della fase di preprocessing dei dati. La corretta generazione e normalizzazione degli spettrogrammi è risultata essere una componente fondamentale per il successo dell'approccio CNN, così come l'adeguata preparazione dei dataset di addestramento per i modelli tradizionali.

Infine, un possibile sviluppo futuro del progetto potrebbe riguardare l'ampliamento del numero di classi strumentali, l'inclusione di condizioni acustiche più variabili, o l'esplorazione di tecniche di transfer learning, che consentirebbero di sfruttare modelli pre-addestrati su grandi corpus audio per migliorare ulteriormente le prestazioni e ridurre i tempi di training.

In conclusione, il confronto tra i tre modelli ha permesso di evidenziare i diversi punti di forza e di debolezza di ciascun approccio, offrendo una visione articolata delle strategie possibili per la classificazione automatica di strumenti musicali. Il percorso di sperimentazione

ha confermato l’efficacia delle reti neurali convoluzionali, ma ha anche mostrato come, in funzione delle esigenze applicative, modelli più semplici possano ancora rappresentare una scelta valida ed efficiente.