UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE
CORSO DI LAUREA IN INGEGNERIA INFORMATICA

"IL MONITORAGGIO ACUSTICO PASSIVO DI STIMOLI ECOLOGICI NEI SISTEMI DI RETI ARTIFICIALI"

Relatore: Prof. Loris Nanni

Laureando: Leonardo Faggion
Matricola numero: 1216373

ANNO ACCADEMICO 2022 – 2023
Data di laurea 16/03/2023

INDICE

ABSTRACT

Nel presente studio, si analizza l'efficacia e la messa in pratica del monitoraggio acustico passivo nell'ambito dell'identificazione di lupi tramite i loro ululati. Il monitoraggio acustico passivo è uno strumento molto utile per la salvaguardia della fauna selvatica, poiché permette di monitorare e censire gli animali, senza dover ricorrere a tecniche invasive, o estremamente costose e quindi poco applicabili.

Sono stati utilizzati gli ululati appartenenti a 9 lupi differenti (sia selvatici, che in cattività) originari della penisola indiana. Da questi 70 ululati, rappresentati da 13 features, sono stati estratti un set di testing e uno di addestramento, su cui è poi stata applicata una *PCA (principal component analysis)*. È quindi stata addestrata una rete neurale sul training set, che è poi stata utilizzata per assegnare a ciascun vettore di testing uno score in base alla sua somiglianza con ogni vettore del set di addestramento. Su questo score è stato poi eseguito un clustering gerarchico che ha permesso di creare dei cluster, che sono stati poi valutati in base all'accuratezza rispetto ai lupi di appartenenza.

Infine, considerando i dati ottenuti, questa tecnica può essere considerata efficace, seppure ancora in fase di sviluppo, e con ampi margini di miglioramento, definiti in gran parte dai dati a disposizione per l'addestramento della rete.

CAPITOLO 1
INTRODUZIONE

Questa tesi si basa sui dati della ricerca effettuata da Sadhukhan, Root-Gutteridge, and Habib, *Identifying unknown Indian wolves by their distinctive howls: its potential as a non-invasive survey method*.

1.1. Le Reti Neurali

Le reti neurali artificiali vengono studiate fin dagli anni '80, tuttavia solo nell'ultimo decennio ne si sono scoperte le numerose potenzialità e applicazioni. Il deep learning, come illustrato dalla School of Management del Politecnico di Milano ('Alla Scoperta Del Deep Learning: Significato, Esempi e Applicazioni' 2021), è composto da numerose reti neurali organizzate in strati, ognuno connesso al precedente, ogni strato elabora i dati che riceve tramite un sistema di pesi, ovvero punteggi basati sul valore del dato ricevuto, e passa i risultati prodotti allo strato successivo. I singoli pesi degli strati vengono tarati facendo analizzare al sistema una vasta quantità di dati, affinché impari da solo a risolvere problemi. Il deep learning è parte della famiglia del machine learning, sempre più diffusa in ambito informatico e non, le sue applicazioni spaziano dall'apprendere le abitudini di un utente su un sito web, al rendere un'auto in grado di guidare autonomamente.

Il recente successo del deep learning è dovuto principalmente alla sempre maggiore disponibilità dei due ingredienti principali per le intelligenze artificiali: dati e capacità computazionale.

L'aumento delle quantità dei dati a disposizione per l'addestramento delle reti e delle prestazioni delle schede grafiche (*graphics processing unit* - *GPU*), come analizzato da N. Evanson (Evanson 2020), in grado di processare numerosi dati in parallelo, ha reso le reti di deep learning negli ambiti applicativi come classificazione di immagini, diagnosi medica e riconoscimento ed elaborazione del linguaggio.

1.2. Il Monitoraggio Acustico Passivo - PAM

Il riconoscimento passivo dei suoni, anche detto passive acoustic monitoring - PAM, è una tecnologia che sta prendendo sempre più piede nel monitoraggio della fauna. È definito passivo proprio perché è una tecnica non invasiva e che quindi non va a turbare animali e piante, né a deturpare il paesaggio. Viene utilizzato installando sensori acustici nell'ambiente da monitorare, lasciandoli a registrare per un determinato periodo di tempo. Successivamente, tramite l'applicazione di algoritmi di deep learning le tracce audio vengono analizzate e selezionate secondo i criteri previsti.

1.3. Applicazioni del PAM

Il PAM è una risorsa molto utile per la supervisione della fauna, e viene maggiormente utilizzata per il monitoraggio di quella marina, poiché il monitoraggio visivo, seppur efficace, è limitato da diversi fattori. Come analizzato da Aguiar et al., nel caso dei mammiferi marini, infatti le immagini possono essere scattate solo quando emergono in superficie, quando c'è bel tempo, e quando le condizioni metereologiche lo permettono (Aguiar et al. 2021). Anche nel caso degli animali che vivono sulla terraferma, sussistono queste limitazioni nel monitoraggio visivo. Dalle ricerche di Wrege et al. si può evincere che il monitoraggio acustico passivo permette di ottenere dati sia di giorno che di notte, mentre secondo di monitorare specie difficilmente distinguibili visivamente o che vivono disperse in vasti territori.

Il PAM, come le fototrappole, permette inoltre di monitorare in modo continuativo un'area senza l'intervento umano e, grazie al recente sviluppo delle tecnologie, è possibile recuperare e analizzare i dati automaticamente come visto nello studio di Gibb et al. (Gibb et al. 2019; Wrege et al. 2017; Pérez-Granados et al. 2019). Per utilizzare il PAM vanno innanzitutto installati i microfoni in luoghi strategici per registrare i suoni ambientali nelle aree di interesse. I microfoni possono essere installati su supporti come pali, alberi o piattaforme galleggianti nel caso si volessero registrare suoni sottomarini. I microfoni, quindi, vanno lasciati a registrare continuamente i suoni ambientali in un periodo di tempo definito (ad esempio giorno o notte) per diversi giorni, settimane o addirittura mesi. Le registrazioni vengono quindi recuperate, catalogate e analizzate da specifici software che a seconda dei parametri scelti riconoscono il tipo di animali presenti nell'area di studio. Questa analisi può essere effettuata manualmente, tramite la consultazione di esperti, o in modo automatico per mezzo di algoritmi. I risultati ottenuti vanno quindi interpretati per determinare la presenza, distribuzione o densità della specie in analisi nell'area di studio.

Il crescente inquinamento ambientale, acustico e urbanistico, rende sempre più cruciale il monitoraggio della fauna selvatica per poterla proteggere e preservare. Tuttavia, non è un compito facile poiché spesso gli animali vivono in zone isolate o molto vaste, quindi difficili da monitorare. Per sapere quali e quanti individui sono presenti in una determinata area, spesso si ricorre all'osservazione umana o a tecniche di monitoraggio visivo. Queste ultime tuttavia, come illustrato precedentemente, sono spesso limitate nel campo d'azione o nell'affidabilità, soprattutto nelle ore notturne. Proprio in queste casistiche viene in aiuto il PAM, che permette di riconoscere gli animali in un ampio raggio e quindi di poter fare valutazioni sulla loro popolazione o su eventuali migrazioni. Inoltre, è stato dimostrato da Avots et al. che con un setup di 4 microfoni sincronizzati si può monitorare un'area di circa mezzo km^2, non solo tenendo traccia del numero di individui ma anche localizzandoli all'interno della zona controllata. (Avots et al. 2022)

Questo metodo innovativo offre una soluzione efficiente per il monitoraggio della fauna selvatica e può contribuire a proteggere e preservare le specie in pericolo.

CAPITOLO 2
METODO

2.1. Specie animale oggetto della ricerca

Nella ricerca, l'attenzione è stata posta principalmente sulla specie faunistica del lupo indiano (*Canis lupus pallipes*), sottospecie del lupo grigio, che viene considerato una specie di alta importanza nell'ecosistema del paesaggio indiano centrale. A differenza del lupo grigio (*Canis lupus*), il quale è una specie terrestre ampiamente diffusa e ben studiata in Nord America ed Europa, il lupo indiano è a rischio di estinzione e le informazioni sulla sua distribuzione attuale sono limitate (Hamid et al. 2019). Uno studio condotto da Jhala et al., ha dimostrato quanto affermato riguardo al ruolo chiave del lupo indiano nell'ecologia della regione indiana indagata, aiutando a mantenere l'equilibrio tra le popolazioni di ungulati in ambienti antropizzati. Tuttavia, la stima della popolazione di questi animali è difficile a causa della vasta area di distribuzione e della loro capacità di evitare le telecamere trappola, evitando i suoni e le luci emessi (Jhala et al. 2022). Sappiamo inoltre che il lupo indiano è a rischio poiché minacciato dall'attività umana, il suo habitat naturale viene sempre più inquinato dall'uomo, aumentando il conflitto uomo-fauna selvatica. Pertanto, la conservazione di questa specie è oltremodo una questione ecologica che richiede particolare attenzione.

In base agli studi di Sadhukhan et al., l'ululato del lupo indiano ha una frequenza fondamentale media di 422 Hz e una durata media di 5,21 secondi, caratteristiche che lo rendono udibile anche a distanze considerevoli. Grazie a queste proprietà acustiche, il lupo indiano rappresenta un candidato eccellente per l'utilizzo della tecnica del PAM (Sadhukhan, Hennelly, and Habib 2019).

2.2. Raccolta dei dati

I dati utilizzati in questa analisi sono stati prelevati dallo studio condotto da Sadhukhan et al., (Sadhukhan, Root-Gutteridge, and Habib 2021) che ha registrato gli ululati dei lupi indiani tra novembre 2015 e luglio 2016, accertandosi di non nuocere nessun animale. Lo studio è stato condotto sia su individui in cattività dello zoo di Jaipur, che di lupi dello stato del Maharashtra, in India. Le registrazioni sono state effettuate durante sondaggi sistematici di ululato, i quali prevedono la riproduzione di ululati di prova e l'ascolto delle risposte dei lupi nella zona circostante, al fine di rilevare la presenza e il numero di lupi in una determinata area. Durante le sessioni di registrazione, che si sono svolte nelle prime ore del mattino e al calare del sole, sono state raccolte anche registrazioni spontanee di ululati di lupi in cattività e in libertà. Ogni sessione di registrazione prevede cinque tentativi con intervalli di 3 minuti, in caso di risposte di ululati venivano terminati e ripresi dopo 15/20 minuti. Tutti gli ululati di prova trasmessi nella ricerca appartenevano ad uno stesso individuo originario dello Zoo di Jaipur e venivano trasmessi per 50 secondi ciascuno con un'ampiezza sempre maggiore, seguiti poi dalla trasmissione di una registrazione corale, della stessa

durata, di tre individui appartenenti ancora una volta allo Zoo. Gli ululati di risposta, se presenti, sono stati registrati da un microfono singolo con una frequenza di campionamento di 44.1 kHz.

2.3. Estrazione delle caratteristiche principali (features)

Per analizzare gli ululati dei lupi indiani, il gruppo di ricerca di Sadhukhan (Sadhukhan, Root-Gutteridge e Habib, 2021) ha utilizzato un algoritmo di Trasformata di Fourier Discreta (DFT) per generare gli spettrogrammi dei segnali audio. In questo modo, solo gli ululati con bassi livelli di rumore di fondo e senza sovrapposizioni sono stati selezionati per le analisi, per poter assicurare una chiara identificazione degli ululati. Inoltre, è stata scelta una lunghezza minima di 5 secondi per escludere le chiamate sociali che presentano caratteristiche simili ma una durata più breve e per aumentare la quantità di dati analizzabili. Sono state estratte 13 *feature* per ogni ululato selezionato, come riportato nella Tabella 1, e solo 69 ululati su 133 che avevano superato la selezione sono stati mantenuti per le successive analisi. Questi 69 ululati sono stati associati a 9 lupi, come indicato nella Tabella 2, 3 in cattività e 6 selvatici.

Nome della variabile	**Definizione**
Min f	La frequenza minima f_0
Max f	La frequenza massima f_0
Range f	Intervallo di f_0. $Range = Max\ f - Min\ f$
Mean f	Frequenza media di f_0 a intervalli di 0.1s
Duration	Durata dell'ululato misurata *in* f_0. $Duration = t_{end} - t_{start}$
$Abrupt_{0.025}$	Numero di cambiamenti improvvisi di f_0 superiori a 25 Hz in un singolo intervallo di tempo (0.1s)
$Abrupt_{0.05}$	Numero di cambiamenti improvvisi di f_0 superiori a 50 Hz in un singolo intervallo di tempo (0.1s)
$Abrupt_{0.1}$	Numero di cambiamenti improvvisi di f_0 superiori a 100 Hz in un singolo intervallo di tempo (0.1s)
Stdv	Deviazione standard di f_0
Co fm	Coefficiente di modulazione di frequenza di f_0. $Co\ fm = \Sigma\|f(t) - f(t+1)\|/(n-1) \times 100/Mean\ f0$
Co fv	Coefficiente di variazione di frequenza di $f0$. $Co\ fv = (SD/mean) \times 100$

Pos Min	Istante temporale in cui si verifica la frequenza minima. *Pos Min = time of Minf/Duration*
Pos Max	Istante temporale in cui si verifica la frequenza massima. *Pos Max = time of Minf/Duration*

Tabella 1: Features estratte dagli ululati.

CAPITOLO 3
ANALISI

Il codice per l'analisi, scritto e implementato in MATLAB Programming Language (MPL), si divide in sei funzioni: *dataTable, training, testtable, dissimilaritySpace, hclustering, accuracy.*

Con *pattern* ci si riferisce a un insieme di caratteristiche che descrivono un oggetto o un fenomeno. Nel caso in analisi un *pattern* è l'insieme di features che compone un vettore-ululato. Un *cluster* invece è un insieme di oggetti o fenomeni che sono stati raggruppati in base a caratteristiche simili. Nel contesto corrente quindi, un *cluster*, composto da *pattern*, rappresenta un'entità lupo.

3.1. Analisi delle componenti principali – PCA

La principal component analysis (PCA), è una tecnica di analisi multivariata utilizzata per analizzare insiemi di dati che comprendono più d'una variabile (Abdi and Williams 2010). L'analisi multivariata è una tecnica statistica che permette di studiare simultaneamente molte variabili che possono essere correlate tra loro, fornendo così informazioni più complete rispetto all'analisi univariata (Abdi 2003). La PCA, in particolare, viene utilizzata per identificare eventuali pattern o cluster all'interno di un set di dati, estraendo le informazioni importanti ed esprimendole creando un nuovo set di variabili che sono combinazioni lineari di quelle originali (Ringnér 2008). Queste nuove variabili dette componenti principali, evidenziano le similarità e le differenze tra le variabili in modo più evidente rispetto al set originale, facilitando così l'interpretazione dei dati.

Il cambio di variabili, inoltre consente di eliminare le variabili meno significative, le informazioni ridondanti e il rumore eccessivo, e ridurre quindi la dimensionalità del set di dati, pur mantenendo la pertinenza e la rappresentatività dei dati originali (Kurita 2020).

La PCA è ampiamente utilizzata in diversi campi, tra cui la biologia, la chimica, l'informatica e la finanza, per analizzare dati complessi e ottenere informazioni utili per prendere decisioni informate.

3.2. Formattazione dati

Utilizzando la funzione *dataTable*, i dati delle feature vengono prelevati dal file originale, in cui i nomi dei lupi a cui appartengono gli ululati sono stati sostituiti con le etichette *[111, 222, 333, 444, 555, 666, 777, 888, 999],* al fine di agevolarne il riconoscimento nel codice MATLAB. Successivamente, i dati vengono importati nel workspace e viene applicata una PCA tramite la funzione *pca* di MATLAB. Viene quindi generata la tabella "trainconc", ottenuta concatenando ogni elemento del training con ogni altro elemento del training. Tale tabella presenta *n* righe e *m* colonne, dove *n* rappresenta il numero di possibili combinazioni fra due vettori nel training set, ovvero $n = x*(x-1)/2$ (dove *x* è il numero di vettori nel training set), e *m* è la somma tra il numero di features del primo vettore e quelle del secondo. Ad esempio, la concatenazione tra i vettori *v1 = [12.2, 0.14]* e *v2*

= [9.3, 0.45] restituirebbe il vettore *v3 = [12.2, 0.14, 9.3, 0.45],* avente una riga e quattro colonne. Inoltre, viene generato il vettore obiettivo "responsesTrain", il quale per ogni coppia indica se i due vettori concatenati appartengono allo stesso lupo o meno. Questo vettore rappresenta l'output ottimale che ci si aspetta dalla rete neurale, quando viene addestrata utilizzando il training set come input. In altre parole, il vettore rappresenta la previsione ideale della rete per ogni input, in base a quanto appreso durante l'addestramento.

Successivamente viene applicato lo stesso processo ai dati del test set, con la differenza che non vi viene applicata direttamente la PCA. La tabella di testing, infatti, viene proiettata nello spazio delle componenti principali calcolate nel set di training, utilizzando i coefficienti calcolati durante l'analisi del set di training. Questo approccio garantisce una coerenza tra i due set di dati e consente di effettuare una classificazione utilizzando la stessa base di componenti principali, senza dover calcolare nuove componenti per il set di testing. Inoltre, questo permette di mantenere la stessa interpretazione dei dati e di poter confrontare direttamente i risultati della classificazione tra i due set di dati.

3.3. Addestramento rete

L'addestramento di una rete neurale è un processo fondamentale nell'ambito dell'apprendimento automatico, che consente di insegnare alla rete a riconoscere pattern e a prendere decisioni basate sui dati di input. In sostanza, l'addestramento consiste nel fornire alla rete neurale un insieme di dati di input e di output corrispondente (il cosiddetto training set), in modo che la rete possa apprendere la relazione tra gli input e gli output desiderati. Durante l'addestramento, la rete neurale viene regolata gradualmente fino a quando non raggiunge un livello di accuratezza accettabile per il problema specifico che si vuole risolvere. In pratica, l'addestramento di una rete neurale prevede la definizione dell'architettura della rete, dei suoi parametri e delle funzioni di attivazione dei suoi strati, nonché la scelta di un algoritmo di ottimizzazione per la regolazione dei pesi dei neuroni (Larochelle et al. 2009).

La funzione *training* si occupa dell'addestramento della rete neurale per l'analisi dei dati, utilizzando la funzione *trainNetwork* fornita da MATLAB.

Nello specifico, la rete viene addestrata per riconoscere se la coppia di vettori audio in ingresso appartiene allo stesso lupo o meno, restituendo uno score in merito. Per addestrare la rete, vengono forniti a *trainNetwork* il training set "trainconc" e il vettore di target "responsesTrain", come descritto in precedenza. Inoltre, vengono definiti i parametri della rete e dei suoi strati.

La rete neurale avrà infatti un layer di input che accetta "numFeatures" features (nel nostro caso 26, ovvero le 13 features del primo ululato combinate alle 13 di quello con cui è concatenato), normalizzate con lo z-score. Lo z-score è una misura di quanto un valore sia lontano dalla media di

un insieme di dati, espresso in unità di deviazione standard. Indica quanti scarti standard il valore si discosta dalla media. Successivamente, vengono aggiunti due layer fully connected, un layer di batch normalization, una funzione di attivazione *ReLU*, uno strato di output con due neuroni e una funzione di attivazione *softmax* per la classificazione. La rete viene addestrata utilizzando l'algoritmo di ottimizzazione Adam con una dimensione di mini-batch di 32, scambiando gli esempi di addestramento in ogni epoca e mostrando i progressi dell'addestramento tramite grafici. Infine, la rete addestrata viene restituita come uscita della funzione *training*. In Figura 1 è rappresentato il progresso del training della rete.

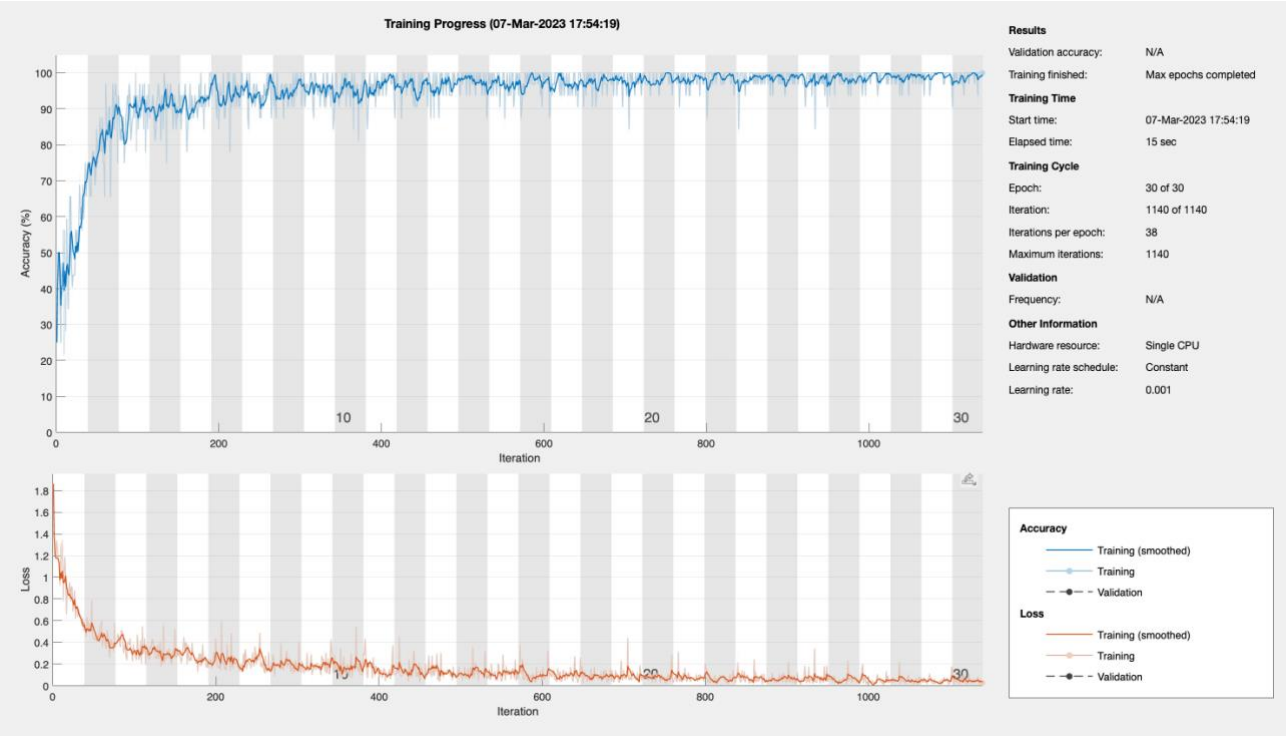

Figura 1: Progresso del training della rete.

3.4. Spazio di dissimilarità (dissimilarity space)

Il dissimilarity space è uno spazio vettoriale in cui le dimensioni sono definite da vettori di dissimilarità, che misurano le dissimilarità tra esempi e oggetti individuali dal cosiddetto set di presentazione. In questo modo, i singoli pattern non sono più definiti dalle variabili originali, ma sono rappresentati tramite *n* distanze rispetto agli *n* vettori dell'insieme di riferimento. L'utilizzo di un dissimilarity space può essere utile per effettuare analisi quantitative su dati complessi, ridurre la dimensionalità di un set di dati, classificare gli oggetti in base alle loro relazioni di somiglianza e visualizzare i dati in uno spazio bidimensionale o tridimensionale (Duin and Pękalska 2012).

Per implementare il dissimilarity space, in questo studio viene prima eseguita la funzione *testtable,* prende in input due tabelle, il test set e il training set, e crea una matrice concatenando orizzontalmente ogni vettore del test set con ogni vettore del training set, in modo da formare tutte le possibili coppie

di vettori. La matrice restituita avrà quindi *n*m* righe e *k* colonne, dove *n* e *m* sono rispettivamente il numero di vettori nel test set e nel training set, mentre k è la somma del numero di features in entrambi i vettori.

La matrice creata viene quindi passata come input alla funzione *dissimilaritySpace*. La funzione *dissimilaritySpace* crea uno spazio di dissimilarità utilizzando la funzione "classify" di MATLAB, la quale utilizza la rete neurale precedentemente addestrata per assegnare un punteggio di similarità ad ogni riga della matrice. Ciò significa che ogni possibile coppia di vettori del set di addestramento e del set di testing viene confrontata per valutare quanto siano simili. Il vettore di score risultante, avente *n*m* elementi in colonna, viene quindi riorganizzato per formare una matrice di dissimilarità con n righe, ognuna corrispondente a un vettore del test set, ed *m* colonne, una per ogni vettore del training set. L'elemento alla n-esima riga e alla m-esima colonna rappresenta quindi il punteggio di dissimilarità tra l'n-esimo vettore del set di testing e l'm-esimo vettore del set di addestramento. In questo modo, è possibile valutare la similarità tra i vettori del set di testing e quelli del set di addestramento.

3.5. Clustering gerarchico

Il clustering gerarchico è un algoritmo di machine learning che si basa sulla distanza o la similarità tra le osservazioni per costruire una gerarchia di cluster. Questo algoritmo può essere utilizzato quando non si sa a priori il numero di cluster in cui suddividere i dati, come nel caso in analisi, poiché non si sa in anticipo a quanti esemplari di lupo appartengono gli ululati registrati. Il clustering gerarchico è un algoritmo "bottom up", il che significa che parte dall'ipotesi che ogni elemento è un cluster a sé stante e combina iterativamente i cluster sulla base della loro somiglianza, finché tutti gli elementi non appartengono a un singolo cluster o a un numero prefissato di cluster (Murtagh and Contreras 2012).

Ad ogni iterazione, il clustering gerarchico genera una gerarchia di cluster, detta dendrogramma (Figura 2), che rappresenta graficamente l'ordine di unione dei cluster. Il dendrogramma è un diagramma a forma di albero che viene generato dal clustering gerarchico. Ogni nodo dell'albero rappresenta un cluster, e le foglie dell'albero rappresentano le osservazioni o i singoli elementi del dataset. Le foglie sono unite iterativamente fino a formare dei cluster sempre più grandi, fino ad arrivare a un unico cluster che contiene tutti gli elementi. Il dendrogramma fornisce una visualizzazione grafica della gerarchia di cluster generata dal clustering gerarchico, in cui l'ordine di unione dei cluster è rappresentato dalla lunghezza delle linee che connettono i nodi. Le linee più lunghe rappresentano la fusione di cluster più distanti, mentre le linee più corte rappresentano la fusione di cluster più simili.

Nel presente studio, viene utilizzato il metodo di linkage “single”, nel quale la somiglianza tra due cluster viene valutata sulla base della minima distanza tra due elementi appartenenti a cluster differenti. Altri metodi di linkage, come il “complete” o l'”average”, utilizzano diverse funzioni di dissimilarità per valutare la somiglianza tra i cluster (Murtagh and Contreras 2011).

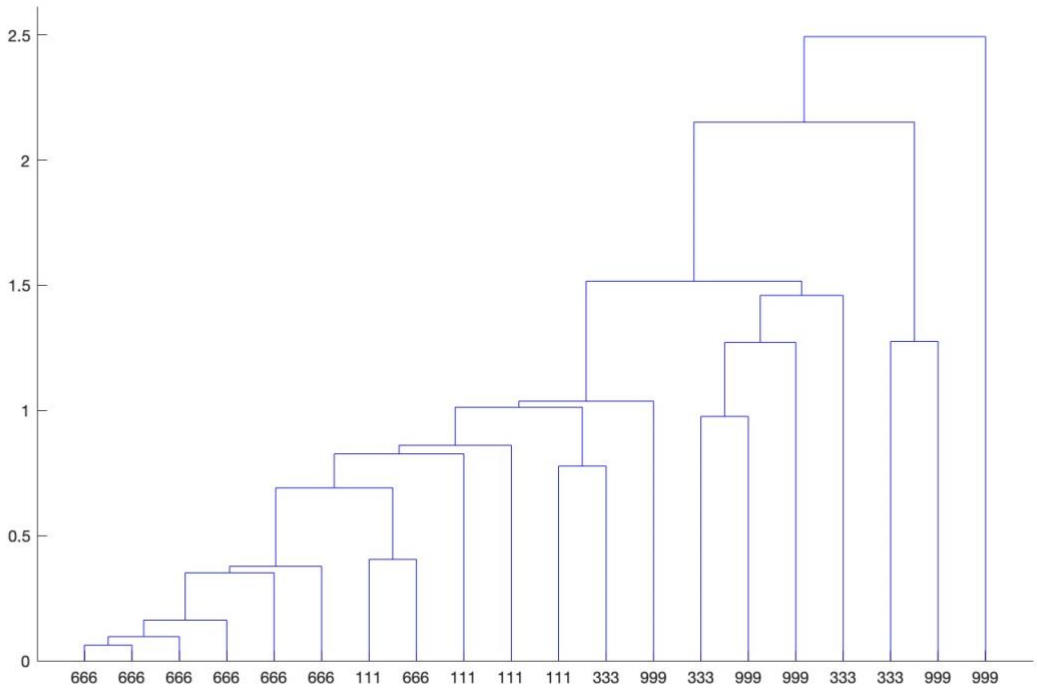

Figura 2: Dendrogramma pattern del training set.

3.6. Verifica accuratezza

L’ultimo step dello script prevede la verifica e il calcolo dell’accuratezza del clustering appena eseguito. Per fare ciò, ad ogni cluster viene assegnato un pattern, scelto come il pattern più rappresentativo di ogni cluster. Per ogni cluster, viene calcolato il vettore mediana, che rappresenta la mediana di ogni feature tra i pattern presenti nel cluster. Il pattern più simile al vettore mediana viene poi selezionato come rappresentate del cluster.

Successivamente, viene conteggiato il numero di pattern assegnati al cluster corretto, ovvero quanti di essi appartengono allo stesso lupo del pattern rappresentativo del cluster. Per calcolare l’accuratezza del clustering si divide semplicemente il numero di pattern assegnati correttamente per il numero totale di pattern.

CAPITOLO 4
RISULTATI

Lo script è stato eseguito più volte, utilizzando diversi set di training e testing. Per ogni set sono state effettuate 50 iterazioni per minimizzare la variabilità e minimizzare l'impatto di eventuali valori anomali. Nella Tabella 2 sono riportati i nomi dei lupi, con le relative etichette. La Tabella 3 e la Tabella 4, invece, riportano per ogni set utilizzato l'accuratezza ottenuta, espressa in forma di media aritmetica, valore massimo e minimo ottenuto nelle 50 iterazioni con il set di dati in analisi. Sono inoltre riportati il numero di cluster e di ululati utilizzati in fase di testing e training, e il coefficiente di Cophenet e di Silhouette. Questi ultimi sono due indicatori di qualità di un modello di clustering, il primo valuta la correlazione tra le distanze originali dei dati e le distanze tra i cluster creati, mentre il secondo misura quanto ogni elemento del cluster sia simile agli altri elementi del cluster

Come si può notare dai dati ottenuti nella Tabella 3, l'accuratezza media dell'addestramento è generalmente più alta di quella del training, come ci si aspettava. Si evince inoltre che i casi in cui si ha l'accuratezza peggiore nel training è quando si ha uno scarso numero di cluster , o di ululati disponibili per l'addestramento.

Nome del lupo	Etichetta utilizzata nel codice	Status (selvaggio o in cattività)	Numero di ululati
BMT.A	111	Selvatico	4
BMT.SA1	222	Selvatico	5
BMT.SA2	333	Selvatico	4
CG1.A1	444	In cattività	9
CG2.A1	555	In cattività	29
CG2.A2	666	In cattività	7
GWD.A	777	Selvatico	4
NNJ.A	888	Selvatico	3
NU.A	999	Selvatico	5

Tabella 2: Nomi dei lupi.

Descrizione test set	Media test	Min test	Max test	Cophenet test	Silhouette test	Num cluster test	Num elementi test
BMT.SA1, CG1.A1, CG2.A1, GWD.A, NNJ.A	49,10%	30,00%	65,00%	0,0870	0,7792	4	20
BMT.A, BMT.SA1, BMT.SA2, GWD.A, NNJ.A, NU.A	42,58%	17,78%	66,67%	0,0458	0,6201	3	45
CG1.A1, CG2.A1, CG2.A2	35,36%	28,00%	44,00%	0,2016	0,6159	6	25
BMT.SA1, CG1.A1, CG2.A1, GWD.A, NU.A	67,78%	55,56%	83,33%	0,0526	0,7247	4	18
BMT.SA1, CG1.A1, CG2.A1, CG2.A2, GWD.A, NU.A	61,45%	36,36%	81,82%	-0,1960	0,7052	3	11
BMT.A, CG1.A1, CG2.A1, GWD.A, NU.A	56,00%	42,11%	68,42%	0,1049	0,6447	4	19
BMT.SA1, BMT.SA2, CG1.A1, CG2.A2, GWD.A, NU.A	85,61%	80,56%	94,44%	0,0669	0,1120	3	36
BMT.A, BMT.SA1, CG1.A1, CG2.A1, CG2.A2, GWD.A, NU.A	61,14%	42,86%	100,00%	-0,0686	0,4715	2	7
BMT.SA1, CG1.A1, CG2.A2, GWD.A, NU.A	48,00%	36,36%	63,64%	-0,0534	0,6699	3	11

Tabella 3: Risultati set di testing.

Descrizione training set	Media train	Min train	Max train	Cophenet train	Silhouette train	Num cluster train	Num elementi train
BMT.A, BMT.SA2, CG2.A2, NU.A	86,64%	82,00%	94,00%	0,2405	0,7845	5	50
CG1.A1, CG2.A1, CG2.A2	40,08%	24,00%	56,00%	-0,0437	0,4148	6	25
BMT.A, BMT.SA1, BMT.SA2, GWD.A, NNJ.A, NU.A	87,20%	84,44%	97,78%	0,5188	0,8988	3	45
BMT.A, BMT.SA2, CG2.A2, NNJ.A	80,38%	78,85%	86,54%	0,2237	0,7734	5	52
BMT.A, BMT.SA2, NNJ.A	72,64%	71,19%	76,27%	0,2029	0,7267	6	59
BMT.SA1, BMT.SA2, CG2.A2, NNJ.A	80,63%	70,59%	84,31%	0,1746	0,7439	5	51
BMT.A, CG2.A1, NNJ.A	36,76%	17,65%	55,88%	0,0031	0,1172	6	34
BMT.SA2, NNJ.A	61,08%	55,56%	68,25%	0,1144	0,4647	7	63
BMT.A, BMT.SA2, NNJ.A	47,60%	20,00%	73,33%	0,0221	0,0732	5	30

Tabella 4: Risultati set di training.

CAPITOLO 5
CONCLUSIONI

I risultati ottenuti sembrano incoraggianti per la ricerca futura. L'accuratezza di clustering dei set di addestramento, infatti, è quasi sempre rimasta tra il 70% e l'80%, e quella di testing ben sopra il 50%. L'analisi eseguita potrebbe inoltre essere stata limitata dalla conformazione dei dati a disposizione. Gli ululati, infatti, non sono equamente distribuiti tra i lupi: i lupi in cattività compongono infatti il 64% degli elementi totali e in particolare il 41% degli ululati appartengono al lupo *CG2.A1*. Questo potrebbe aver portato a una scarsa diversificazione del set di addestramento e quindi ad un over fitting. Ciononostante, i risultati positivi ottenuti rendono credibile l'ipotesi di un miglioramento delle prestazioni del PAM, avendo a disposizione set di dati più ampi e quindi più diversificati.

Da questo studio si può infatti evincere che il PAM, anche con le prestazioni attuali, è un ottimo ed effettivo strumento per il monitoraggio della fauna. Esso nonostante abbia prestazioni inferiori a quelle dell'identificazione visiva o tramite DNA, è un'ottima alternativa economica, non invasivo nei confronti degli animali e facilmente implementabile.

RIFERIMENTI BIBLIOGRAFICI

Aguiar, Rafael, Gianluca Maguolo, Loris Nanni, Yandre Costa, and Carlos Silla. 2021. 'On the Importance of Passive Acoustic Monitoring Filters'. *Journal of Marine Science and Engineering* 9 (7): 685. https://doi.org/10.3390/jmse9070685.

'Alla Scoperta Del Deep Learning: Significato, Esempi e Applicazioni'. 2021. *Blog.Osservatori.Net* (blog). 2 February 2021. https://blog.osservatori.net/it_it/deep-learning-significato-esempiapplicazioni.

Avots, Egils, Alekss Vecvanags, Jevgenijs Filipovs, Agris Brauns, Gundars Skudrins, Gundega Done, Janis Ozolins, Gholamreza Anbarjafari, and Dainis Jakovels. 2022. 'Towards Automated Detection and Localization of Red Deer Cervus Elaphus Using Passive Acoustic Sensors during the Rut'. *Remote Sensing* 14 (10): 2464. https://doi.org/10.3390/rs14102464.

Evanson, Nick. 2020. '25 Years Later: A Brief Analysis of GPU Processing Efficiency'. *Techspot* (blog). 14 April 2020. https://www.techspot.com/article/2008-gpu-efficiency-historical-analysis/.

Gibb, Rory, Ella Browning, Paul Glover-Kapfer, and Kate E. Jones. 2019. 'Emerging Opportunities and Challenges for Passive Acoustics in Ecological Assessment and Monitoring'. Edited by Luca Börger. *Methods in Ecology and Evolution* 10 (2): 169–85. https://doi.org/10.1111/2041-210X.13101.

Pérez-Granados, Cristian, David Giralt, Adrián Barrero, Julia Gomez-Catasús, Daniel Bustillo-De La Rosa, Juan Traba, and Gerard Bota. 2019. 'Vocal Activity Rate Index: A Useful Method to Infer Terrestrial Bird Abundance with Acoustic Monitoring'.

Sadhukhan, Sougata, Lauren Hennelly, and Bilal Habib. 2019. 'Characterising the Harmonic Vocal Repertoire of the Indian Wolf (Canis Lupus Pallipes)'. Edited by Govindhaswamy Umapathy. *PLOS ONE* 14 (10): e0216186. https://doi.org/10.1371/journal.pone.0216186.

Sadhukhan, Sougata, Holly Root-Gutteridge, and Bilal Habib. 2021. 'Identifying Unknown Indian Wolves by Their Distinctive Howls: Its Potential as a Non-Invasive Survey Method'. *Scientific Reports* 11 (1): 7309. https://doi.org/10.1038/s41598-021-86718-w.

Wrege, Peter H., Elizabeth D. Rowland, Sara Keen, and Yu Shiu. 2017. ‘Acoustic Monitoring for Conservation in Tropical Forests: Examples from Forest Elephants’. Edited by Jason Matthiopoulos. *Methods in Ecology and Evolution* 8 (10): 1292–1301. https://doi.org/10.1111/2041-210X.12730.