

Sex and age determination in European lobsters using AI-Enhanced bioacoustics

Feliciano Pedro Francisco Domingos^a, Isibor Kennedy Ihianle^a, Omprakash Kaiwartya^a, Ahmad Lotfi^a, Nicola Khan^c, Nicholas Beaudreau^b, Amaya Albalat^b, Pedro Machado^a

^aNottingham Trent University, Erasmus Darwin Building Clifton, NG11 8NS, Nottingham, United Kingdom

^bUniversity of Stirling, Pathfoot Building, Stirling, FK9 4LA, Scotland, Scotland, United Kingdom

^cAce Aquatec, 1 Water's Edge, Camperdown Street, Dundee, DD1 3HY, Scotland, Scotland, United Kingdom

Abstract

Monitoring aquatic species presents considerable challenges due to their elusive nature and complex habitats. Consequently, the development and application of innovative, non-invasive approaches, such as Passive Acoustic Monitoring (PAM), are paramount for effective ecological assessment and management. This study addresses these challenges by focusing on the European lobster (*Homarus Gammarus*), a key representative species of rocky benthic environments that underpins valuable local fisheries and aquaculture ventures. Comprehensive understanding of lobster habitats, welfare, reproduction, sex, and age is critical for robust aquaculture management, ecological research, conservation strategies and sustainable fisheries. While bioacoustic emissions have been successfully employed to classify various aquatic species using Artificial Intelligence (AI) models, such as fish, this research specifically leverages lobster bioacoustics to classify *Homarus Gammarus* by age group (juvenile and adult) and sex (male and female). Despite lacking vocal cords, different types of lobsters produce a variety of characteristic sounds, such as stridulation characterised by the European spiny lobster (*Panulirus elephas*) and Caribbean spiny lobster (*Panulirus argus*), buzzing or carapace vibrations expressed by the *Homarus Gammarus* and American lobster (*Homarus Americanus*), rattling sounds marked by the Tropical spiny lobster (*Panulirus ornatus*) and clicking or snapping sounds. These varied lobster sounds are amenable to classification using advanced computational AI models. The dataset collection was carried at Johnshaven in Scotland at the local Lobster Shop of Murray McBay and Company. Hydrophones were used and installed underwater in concrete tanks to record the lobster sounds. This investigation explores the efficacy of Deep Learning (DL) models (specifically One-Dimensional Convolutional Neural Networks (1D-CNN) and One-Dimensional Deep Convolutional Neural Networks (1D-DCNN) with varying hidden layers) and six commonly employed Machine Learning (ML) models (Support Vector Machine (SVM), k-Nearest Neighbors (KNN), Naive Bayes (NB), Random Forest (RF), Extreme Gradient Boosting (XGBOOST), and Multi Layer Perceptron (MLP) in classifying buzzing sounds or carapace vibrations of *Homarus Gammarus* for age and sex classification. The Mel Frequency Cepstral Coefficients (MFCC) served as features for all models across four distinct datasets. MFCC has been reported to extract reliable features from audio signals. The majority of models achieved classification accuracies exceeding 97% for adult versus juvenile differentiation, with the exception of NB, which attained 91.31%. For sex classification, all models except NB surpassed 93.23% accuracy. These compelling results underscore the substantial potential of supervised ML and DL models to extract age- and sex-related features from lobster sounds. Ultimately, this research offers a promising non-invasive approach for lobster conservation, detection, and management within aquaculture and fisheries, thereby enabling real-world edge computing applications for PAM of underwater species.

Keywords: Lobster bioacoustics, AI, ML, DL, age and sex classification, aquaculture management, PAM.

1. Introduction

Effective lobster aquaculture management, ecological research, and conservation strategies require a detailed understanding of habitat use, welfare, reproductive conditions, and the accurate identification of sex and age Barroso et al. (2023); Looby et al. (2023). Bioacoustics has emerged as a powerful tool to detect and monitor behavioural and ecological traits, with demonstrated success across avian and aquatic taxa, including fish and invertebrates Bardeli et al. (2010); Noda et al. (2016); Malfante et al. (2018). Lobsters produce diverse acoustic signals depending on species, including stridulation (*Panulirus elephas*), buzzing or carapace vibrations (*Homarus Americanus*, *Homarus Gammarus*), rattling (*Panulirus ornatus*), and clicking or snapping sounds Patek (2002). These emissions can be systematically classified using artificial intelligence (AI) models Briones-Fourzán et al. (2014). The present study focuses on classifying buzzing sounds (carapace vibrations) produced by *Homarus Gammarus* to distinguish individuals by age class (adult vs. juvenile) and sex (male vs. female).

Sound production in lobsters arises from mechanical actions: *Panulirus elephas* and *Panulirus ornatus* generate stridulation by rubbing or contracting body parts, whereas *Homarus Gammarus* emit vibrations of the carapace in the 80–250 Hz range Patek (2002); Staaterman et al. (2010); Kerkhofs (2024). Such bioacoustic emissions are vital for aquatic life, serving functions including predatory avoidance, defence, aggression, courtship, and mating Domingos et al. (2024); Looby et al. (2023); Barroso et al. (2023); Weilgart (2018). More than 900 species of fish and invertebrates are known to produce active sounds, reflecting the centrality of acoustic communication in marine ecosystems Jézéquel et al. (2020b). For lobster aquaculture and fisheries, reliable and efficient methods for sex and age determination are critical to optimise yields, welfare, and sustainability McGarvey et al. (2015). Traditional manual methods are time-consuming, labour-intensive, and prone to human error, motivating exploration of DL approaches such as 1D-CNN, 1D-DCNN, alongside established ML models, for classification of *Homarus Gammarus* acoustic signals.

To situate our study within broader management contexts, it is essential to note that effective lobster management extends beyond sex and age identification. Population dynamics are shaped by distribution patterns, life-history traits, spawning behaviours, and environmental pressures, including climate change (Araújo and Guilhaumon, 2018; Phillips et al., 2015; Radhakrishnan et al., 2020). Emerging threats, such as contamination by microplastics and heavy metals, further impact lobster health and seafood safety (Hwang et al., 2019; Yadav et al., 2024). Figure 1 illustrates how bioacoustics-based sex and age classification integrates within this wider ecological and policy framework.

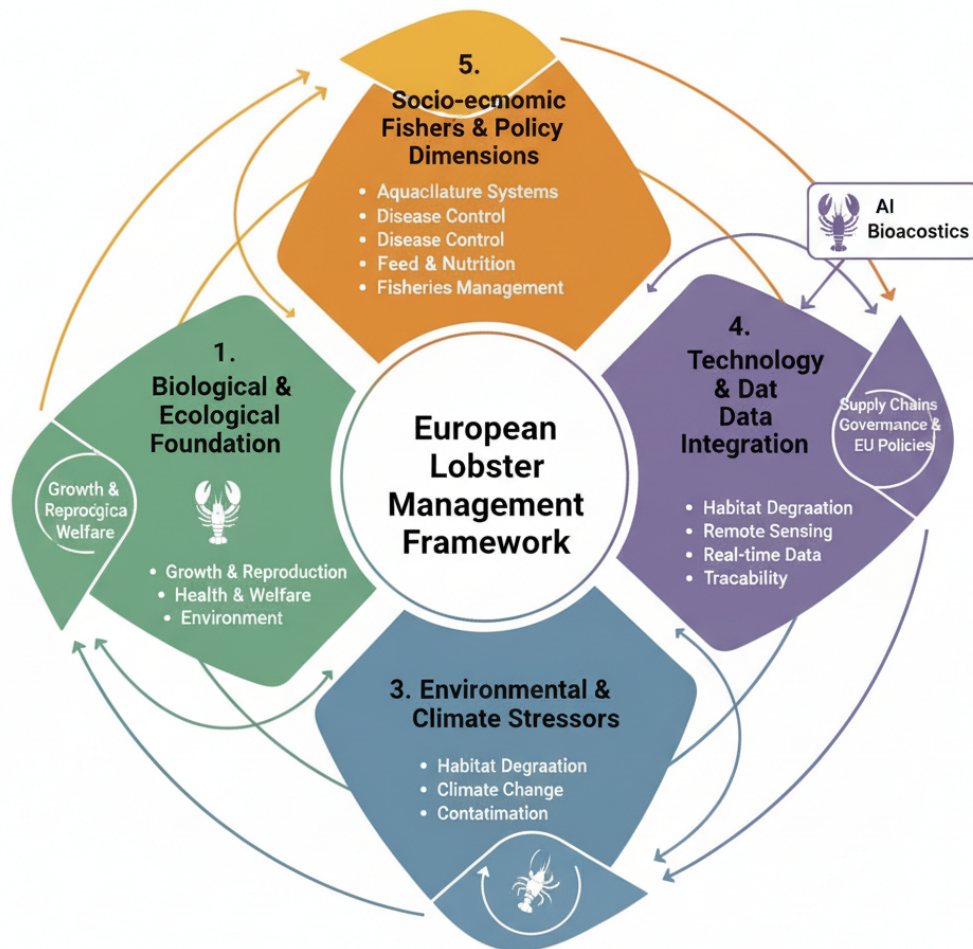


Figure 1: Conceptual diagram of lobster management context showing how AI-enhanced bioacoustics for sex and age determination fits within broader ecological and policy considerations.

In addition, regulatory frameworks such as European Union (EU) fisheries policies are central to stock sustainability, though they face ongoing scrutiny regarding ecological effectiveness and socio-economic trade-offs

(European Commission, Directorate-General for Maritime Affairs and Fisheries, 2025; Council of the European Union, 2025). Thus, management decisions are embedded in governance as much as in biology.

Our contribution addresses this interdisciplinary challenge by applying AI models to classify lobster bioacoustics. Recent advances in ML and DL have demonstrated strong potential for detecting and classifying acoustic signatures of fish and crustaceans, enabling improved resource monitoring (Niu et al., 2023; Bharath et al., 2025; Rountree et al., 2023). We argue that AI-enhanced bioacoustic monitoring offers a scalable, non-invasive method for lobster aquaculture and conservation, complementing ecological, socio-economic, and policy dimensions.

The main contributions of this article are as follows: (i) classification of lobster bioacoustics to determine age and sex; (ii) comparative evaluation of traditional ML and modern DL models, including four 1D-CNN and four 1D-DCNN architectures; (iii) analysis of model performance metrics to identify the most effective approaches; and (iv) a critical synthesis of related works on lobster acoustics and AI classification.

2. Related Works

2.1. Sound Production and Detection in Lobsters

The understanding of how marine invertebrates produce and perceive sound remains limited Taylor and Patek (2010); Jézéquel et al. (2020b). In both *Homarus Americanus* and *Homarus Gammarus*, vibrations and noise are generated by opposing remotor and promotor muscle contractions beneath the second antenna Kerkhofs (2024). The abdominal vibrations of *Homarus Americanus* typically exhibit a base frequency of approximately 180 Hz, consistent with earlier reports Staaterman et al. (2010). For sound detection, external cuticular hairs appear to play a key role, while evidence suggests that crustacean statocysts are not primary auditory organs Jézéquel et al. (2021). Spiny lobsters (*Panulirus* spp.) also produce sounds through stick-slip mechanisms, where antennal bases rub against the antennular plate Patek (2002).

2.2. Behavioural Context of Acoustic Emissions

Lobster acoustic emissions are closely tied to behavioural states Jézéquel et al. (2018); Looby et al. (2023); Domingos et al. (2024). Stressed *Homarus Gammarus* individuals vibrate their carapace, producing low-frequency “buzzing” sounds similar to *Homarus Americanus* Jézéquel et al. (2018, 2021, 2020b). During feeding, lobsters emit broadband, transient “rattles,” comparable to those of tropical spiny lobsters (*Palinurus longipes*, *P. argus*) Jézéquel et al. (2018). Spiny lobsters also stridulate in anti-predatory contexts Bouwma and Herrnkind (2009); Kikuchi et al. (2015). Male *Homarus Gammarus* produce low-frequency buzzing during agonistic encounters, though tank-based hydrophones capture only ~15% of such signals due to attenuation Jézéquel et al. (2020b). Accelerometer-based detection has been employed to overcome these challenges.

2.3. Propagation and Environmental Considerations

Under natural low-noise conditions, spiny lobster sounds propagate over several kilometers, with the largest individuals exceeding 3 km Jézéquel et al. (2020a). However, tank environments introduce reverberation and attenuation, which complicates analysis. Artificial sound source experiments demonstrated the importance of characterising tank acoustics before interpreting lobster behaviour from such recordings Jézéquel et al. (2018). These environmental factors highlight both the potential and limitations of laboratory-based bioacoustic studies.

2.4. Applications of AI in Marine Bioacoustics

Beyond ecological studies, machine learning (ML) and deep learning (DL) approaches are increasingly used for acoustic classification. Discriminative ML models, such as Random Forests (RF) and Support Vector Machines (SVM), have achieved high accuracy in fish sound classification (up to 96.9%) Malfante et al. (2018). ML techniques are also being explored for undersea communication and environmental monitoring, offering noise reduction, adaptive modulation, and robust data handling Domingos et al. (2024); Islam et al. (2025).

DL architectures, including Convolutional Neural Networks (CNNs) and Deep Belief Networks (DBNs), have shown superior performance in underwater acoustic target classification, achieving accuracies above 94% Yue et al. (2017). While MFCCs remain standard features for ML models Davis and Mermelstein (1980a); Slaney (1998), CNNs and DBNs can directly process raw signals or FFT-based features, reducing reliance on handcrafted features. For lobster acoustics specifically, 1D-CNN and dilated CNNs (1D-DCNN) offer the ability to capture temporal dynamics directly from raw or minimally processed signals Huang et al. (2019).

2.5. Towards Predictive Modelling in Lobster Bioacoustics

Recent advances emphasise predictive performance over inference in ecological modelling Breiman (2001b). Ensemble methods such as bagging and boosting improve robustness, with RF reducing variance through bootstrapped sampling Breiman (2001a, 1996a) and XGBoost offering efficient gradient boosting with regularisation Chen and Guestrin (2016). Stacking methods further enhance accuracy by combining diverse learners (e.g., SVM, RF, XGBoost, CNNs) and training a meta-learner on out-of-fold predictions Wolpert (1992); Breiman (1996b); Hastie et al. (2001). These approaches hold strong potential for automated lobster classification, with applications in fisheries management, aquaculture monitoring, and conservation planning.

3. Materials and Methodology

This study employed a structured methodology encompassing lobster sound acquisition, preprocessing, feature extraction, dataset preparation, model training with hyperparameter optimisation, performance evaluation, and considerations for edge deployment and reproducibility. The workflow is summarised in Figure 2.

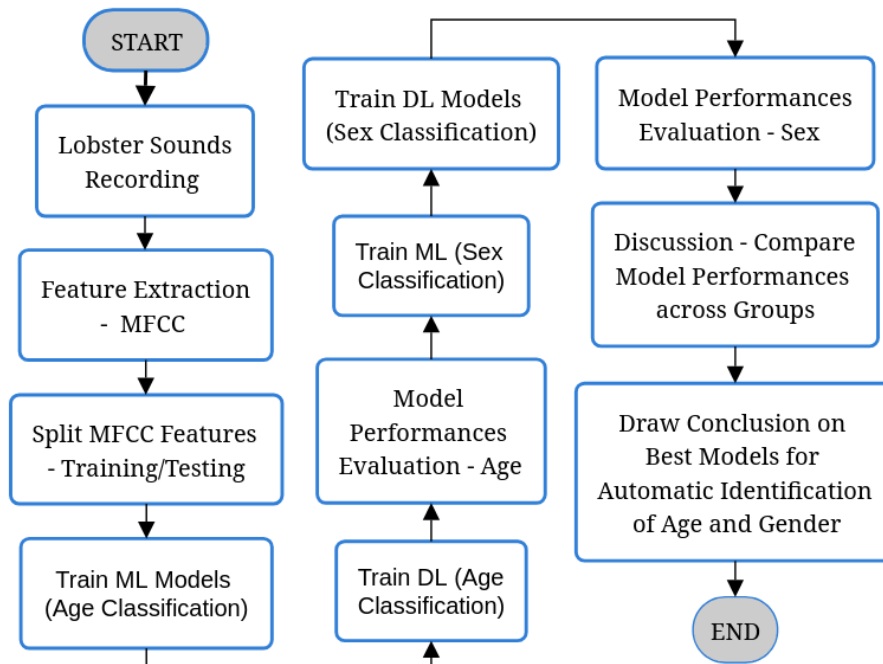


Figure 2: Research methodology flowchart depicting the stages undertaken in this study.

3.1. Materials Specifications

Underwater recordings were acquired using an Aquarian Audio H2d hydrophone connected to a Raspberry Pi 3, with audio captured in real time via the PyAudio library. Recordings were subsequently stored and processed on a Lenovo ThinkPad X1 Carbon Gen 13 Aura Edition (PF-2D95WH). Detailed specifications of the Raspberry Pi and hydrophone are provided in Appendix 8. This hardware configuration balanced portability, real-time acquisition, and computational capability for both data collection and analysis.

3.2. Study Species and Dataset

The dataset comprised acoustic recordings of *Homarus Gammarus* (European lobster). In total, 7307 s of recordings were collected from 24 individuals, stratified into four groups: 6 juvenile females, 6 juvenile males, 6 adult males, and 6 adult females. Furthermore, all juvenile were joined (male and female), all adults were joined (male and female); likewise, all male were joined (juvenile and adult) and all female were joined (juvenile and adult). Then, each new class, the long were split into 1 second segments, which were used for the MFCC feature extraction for the classification. Table 1 summarises the dataset by sex and age categories. All recordings were conducted at the Johnshaven Lobster Shop (Scotland) between 16–18 June 2024, with individual sex and age identified visually by experts based on gonopore position and pleopod morphology.

Table 1: Summary of lobster acoustic dataset by sex and age categories.

Category	Number of Individuals	Recording Duration (s)
Juvenile Female Lobsters	6	1200
Juvenile Male Lobsters	6	1200
Adult Male Lobsters	6	3207
Adult Female Lobsters	6	1700
Total	24	7307

3.3. Dataset Recording

Lobsters were housed in large concrete tanks supplied with continuously circulating seawater, low light, and aeration via pumps. Claws were restrained with rubber bands to prevent aggression Shabani et al. (2009). Concrete tanks were preferred over metal tanks due to reduced reverberation Jézéquel et al. (2018). Acoustic recordings were acquired at 22.05 kHz, segmented into 1 s files with buffered reads of 6,400 samples. A band-pass filter (50–8,000 Hz) was applied to reduce low-frequency flow noise and high-frequency artefacts.

Sessions ran between 08:00–16:00. Lobsters were not fed during recordings, and environmental conditions (temperature, salinity) were not strictly controlled, ensuring realistic holding conditions. Each lobster was recorded sequentially by age and sex category, then returned to its holding tank. Figure 3 illustrates the recording setup.

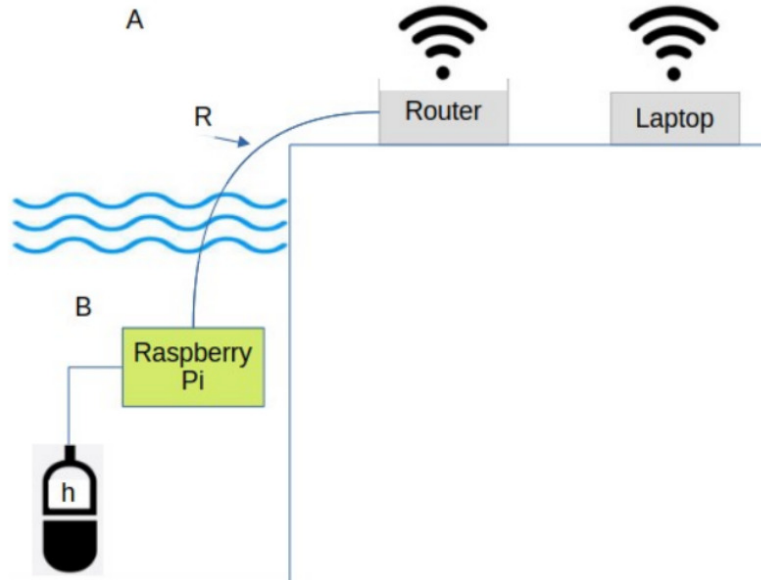


Figure 3: Schematic of the acoustic recording system. Area A: airborne environment; Area B: underwater environment where lobster sounds were recorded.

3.4. Data Preprocessing

Preprocessing ensured consistency and noise reduction prior to feature extraction:

1. **High-pass filtering:** DC offset removed (20–50 Hz).
2. **Band-pass filtering:** 50–8,000 Hz range retained to capture lobster sound energy Jézéquel et al. (2020b, 2021).
3. **Segmentation:** Fixed 1 s windows.
4. **SNR screening:** Low-energy segments discarded.
5. **Normalisation:** Z-score standardisation applied to all extracted features.

3.5. Feature Extraction

Feature extraction relied exclusively on Mel-Frequency Cepstral Coefficients (MFCCs). For each 1 s segment, 40, 50, and 60 MFCCs were computed using the `librosa` library and averaged over time, producing a 40-, 50-, and 60-dimensional feature vector. This representation balances robustness to noise with compactness, widely adopted in both speech and bioacoustic classification. Features were standardised prior to model training. No spectrograms or raw waveform inputs were used.

3.6. Class Balancing and Splitting

To prevent identity leakage, splitting was performed at the individual level. Data were partitioned into 80% for training and 20% for testing, stratified by both sex and age classes. Evaluation metrics were calculated on the held-out 20% test set. This split ensured sufficient training data while retaining robust generalisation evaluation.

3.7. Models Evaluated

Two families of models were implemented for classification:

Classical ML models: SVM, KNN, NB, RF, XGBOOST, and MLP. All were trained on the 40, 50 and 60-dimensional MFCC feature vectors, with hyperparameters tuned via cross-validation.

Deep learning model: A 1D Convolutional Neural Network (1D-CNN) implemented with Keras/TensorFlow. The architecture comprised:

1. Conv1D layer (64 filters, kernel size = 3, ReLU).
2. MaxPooling1D layer (pool size = 2).
3. Flatten layer.
4. Dense hidden layer (128 units, ReLU).
5. Dense output layer (1 unit, sigmoid) for binary classification.

Inputs were reshaped MFCC vectors of dimension (40, 1). Training employed the Adam optimiser, binary cross-entropy loss, batch size 32, and 10 epochs with early stopping.

3.8. Performance Metrics

Performance was assessed using accuracy, precision, recall, F1-score, AUC-ROC, and inference time. Confusion matrices provided error analysis. For rigour, statistical comparisons between models used bootstrap confidence intervals for AUC differences and McNemar tests for paired classification errors, with Benjamini–Hochberg correction applied for multiple comparisons.

3.9. Edge Deployment Considerations

Inference feasibility on edge devices was evaluated (e.g., Raspberry Pi, Jetson Nano). Model optimisation techniques such as quantisation, pruning, and knowledge distillation were considered to improve computational efficiency without significant loss of accuracy Sze et al. (2017); Han et al. (2015). This ensures practical transfer of the pipeline into real-world aquaculture and fisheries monitoring.

3.10. Reproducibility and Data Availability

In line with FAIR principles, all raw and processed datasets are publicly available via Mendeley Data Domingos (2025a), with metadata structured according to ISO 19115 standards. Preprocessing and training scripts are openly shared on GitHub Domingos (2025b) to enable full reproducibility.

4. Experiments and Results

This study evaluates whether biological attributes (age and sex) can be inferred from lobster bioacoustic recordings using a range of machine learning (ML) and deep learning (DL) methods. Two experiments were carried out:

1. **Experiment 1 — Age-based classification (Adult vs. Juvenile).** We compare six classical ML algorithms and eight DL variants (1D-CNN / 1D-DCNN families with varying depth).
2. **Experiment 2 — Sex-based classification (Male vs. Female).** A comparable set of models and metrics are used to assess discriminability by sex.

For both experiments we report standard discrimination metrics (Accuracy, Precision, Recall, F1-score, AUC-ROC) and per-sample inference time (IT). Inputs are MFCC representations (tested at 40, 50 and 60 coefficients) and, where indicated, features were decorrelated with PCA. All metrics reported are point estimates computed on the held-out test partitions.

4.1. Experiment 1: Age-based bioacoustics classification

We framed age classification as a binary problem (Adult vs. Juvenile) and evaluated six classical ML models (RF, XGBOOST, MLP, NB, SVM, KNN) and a family of DL models (1D-CNN and 1D-DCNN with 1–4 hidden layers). Tables 2 and 4 summarise results across MFCC dimensionalities (40, 50, 60). Figures 4 and 5 illustrate metric trends for KNN and SVM as MFCC number varies; confusion matrices for leading models are presented in Figures 6–11.

4.1.1. Key observations

- **High discrimination overall.** With the exception of NB, nearly all models achieve accuracy $\gtrsim 97\%$ on the Adult vs. Juvenile task.
- **Top performers.** SVM and MLP are among the top performers in several metrics; the highest single accuracy (per the provided point estimates) is **98.50%** (SVM, MFCC=50). MLP achieves the best Area Under the Receiver Operating Characteristic Curve (AUC-ROC) (**99.86%**, MFCC=60).
- **Inference time trade-offs.** Classical ML methods such as KNN and NB yield very low Inference Time (IT) (sub-millisecond to a few milliseconds), whereas DL models require seconds per sample — an important operational consideration.
- **Effect of MFCC dimensionality.** There is no universal, monotonic improvement with larger MFCC counts: model-dependent behaviour is observed (some models improve slightly, others remain unchanged or vary non-monotonically).

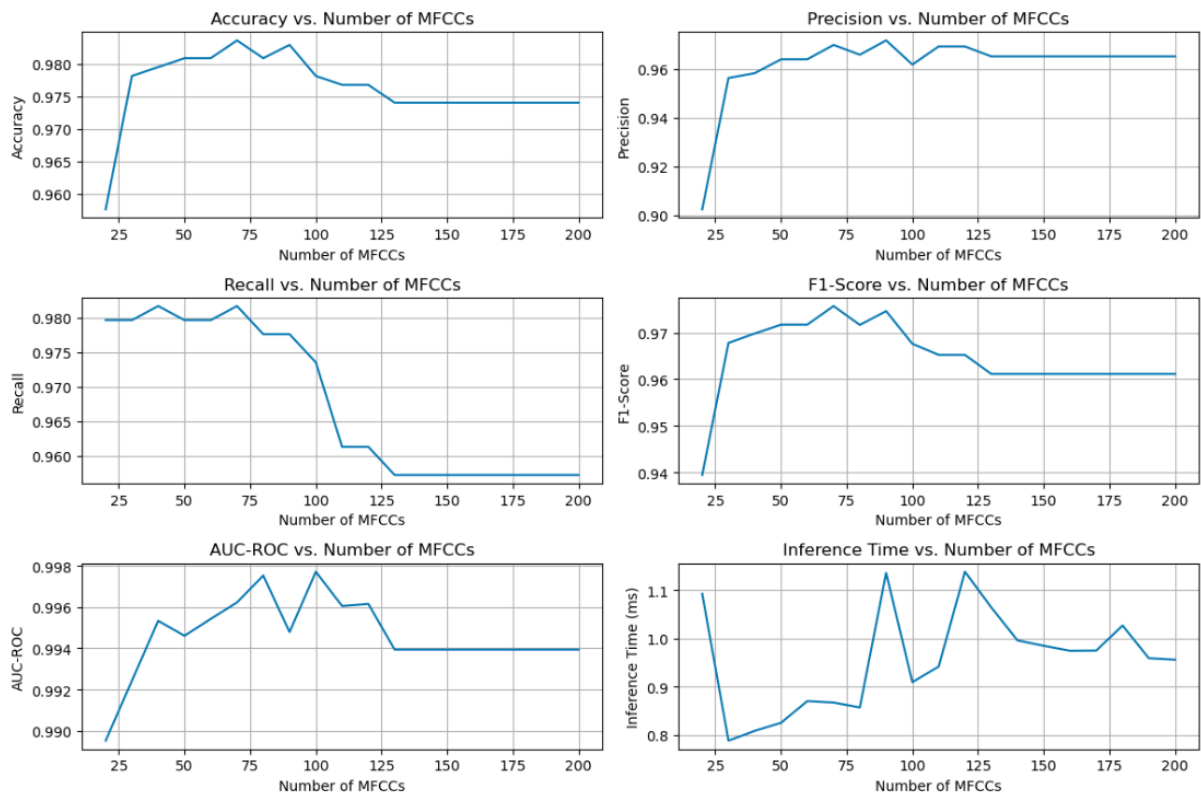


Figure 4: Performance metrics for KNN as a function of MFCC dimensionality (Adult vs. Juvenile).

From Table 3 it is conclusive that MLP and SVM tie as best by average rank (MLP best by small margin), KNN and XGBOOST are competitive mid-pack, RF lags mainly due to inference time, and NB ranks last due to poor discrimination despite fast inference.

The DL family results are shown next. As in the ML table above, best values per column have been highlighted (ties bolded).

Based on the results in Table 5, 1D-CNN (1 layer) and 1D-DCNN (1–2 layers) yield the best average rank: good discrimination with comparatively lower inference costs among DLs. Very deep variants (3–4 layers) rarely yield substantial gains but increase inference latency.

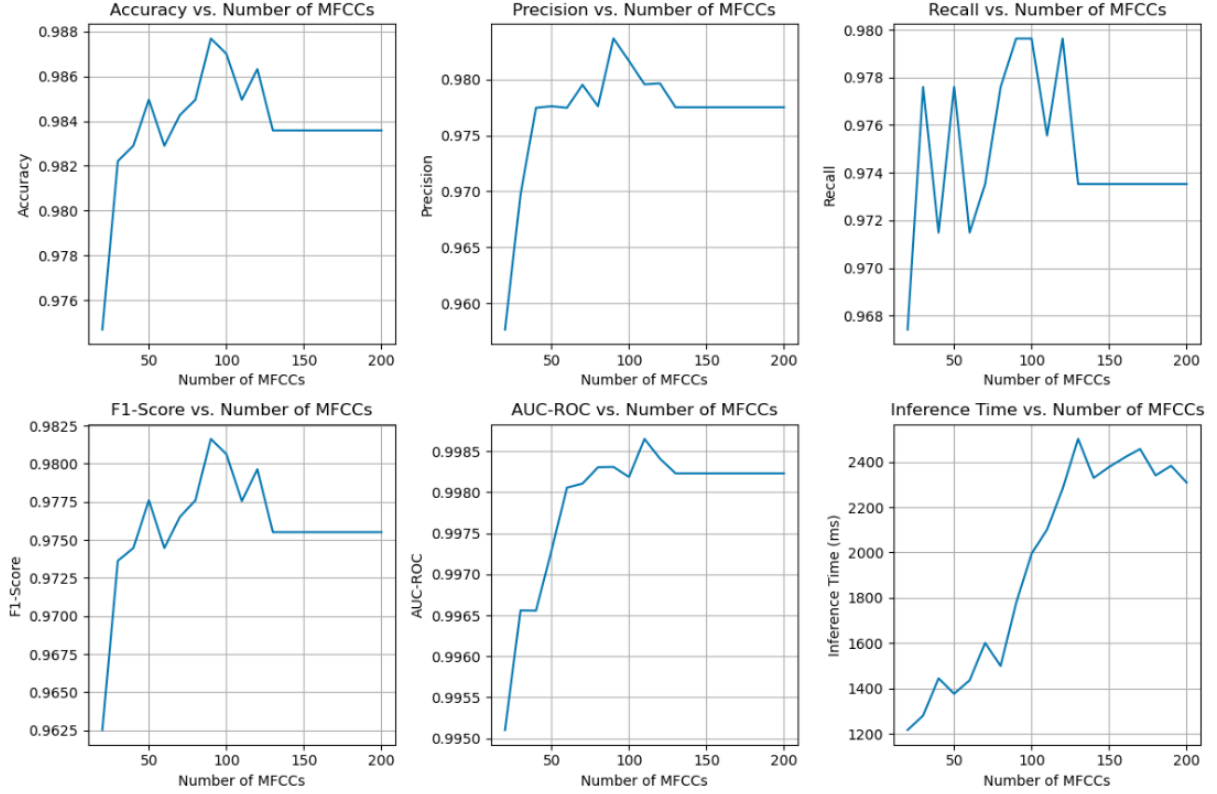


Figure 5: Performance metrics for SVM as a function of MFCC dimensionality (Adult vs. Juvenile).

4.2. Experiment 1: Comparative analysis

Overall, classical ML and shallow DL models both achieve strong performance on the adult vs. juvenile task. Key takeaways:

- **SVM and MLP are among the most consistent classifiers** with top-tier accuracy and F1; SVM reaches the top accuracy point while MLP yields the top AUC.
- **KNN has excellent recall and the fastest inference time** (suitable for low-latency systems), while Naive Bayes is computationally cheap but substantially weaker in accuracy.
- **DL models can match or slightly exceed ML performance** (notably some 1D-DCNN variants), but generally at a significantly higher inference cost.
- **MFCC dimensionality matters but is model dependent.** We cannot recommend a single MFCC count for all models; nested CV and broader sweeps (e.g., 20–80) are advised.

4.3. Experiment 2: Sex-based bioacoustics classification

We evaluated the capacity of machine-learning (ML) and deep-learning (DL) models to discriminate sex (male vs. female) from lobster bioacoustic recordings. Models tested include Random Forest (RF), XGBoost, multilayer perceptron (MLP), Naive Bayes (NB), support vector machine (SVM), K-nearest neighbours (KNN) and a family of 1D convolutional networks (1D-CNN and 1D-DCNN) with varying depths. All inputs were derived from MFCC representations (40, 50 and 60 coefficients) and preprocessed with PCA where indicated. Performance is summarised in Table 6 (ML classifiers) and Table 8 (DL classifiers). Figure 13 illustrates metric trends for the 1D-CNN (1 hidden layer) across MFCC dimensions.

The ranking summary in Table 7 indicates that for the classification of Male vs Female SVM and MLP are the strongest performers by average rank (SVM marginally best on accuracy and multiple metrics; MLP achieves top AUC), while KNN is notable for best precision and best (lowest) inference time. XGBOOST and RF occupy the mid-range; NB ranks lowest due to substantially weaker discrimination despite very low IT.

Table 2: Performance metrics of ML models for Adult vs. Juvenile classification (MFCC = 40, 50, 60). Best value per column is **bolded** (for IT lower is better).

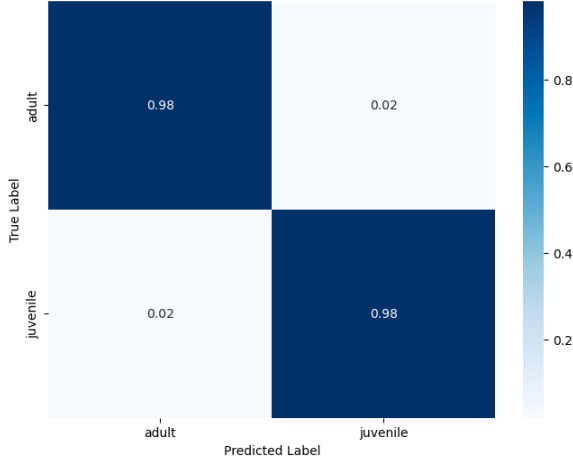
Model	MFCC	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	AUC-ROC (%)	Inference time (ms)
Random Forest	40	97.20	97.27	94.30	95.76	99.50	2300.04
	50	97.26	97.47	94.30	95.86	99.60	2511.00
	60	97.20	97.67	93.89	95.74	99.62	2535.04
XGBoost	40	97.47	96.71	95.72	96.21	99.65	120.47
	50	97.61	96.53	96.33	96.43	99.74	154.90
	60	97.67	96.54	96.54	96.54	99.75	148.32
MLP	40	98.43	98.35	96.95	97.64	99.76	1271.72
	50	97.88	96.75	96.95	96.85	99.82	1452.11
	60	98.43	97.37	97.96	97.66	99.86	2015.05
Naive Bayes	40	89.12	80.29	89.61	84.70	97.09	2.10
	50	91.31	82.04	94.91	88.01	97.73	2.31
	60	90.83	80.94	95.11	87.45	97.80	2.38
SVM	40	98.29	97.75	97.15	97.45	99.66	1381.98
	50	98.50	97.76	97.76	97.76	99.73	1437.78
	60	98.29	97.75	97.15	97.45	99.81	1507.90
KNN	40	97.95	95.83	98.17	96.98	99.53	0.87
	50	98.08	96.39	97.96	97.17	99.46	0.91
	60	98.08	96.39	97.96	97.17	99.54	1.81

Notes: Best values per column are bolded.

Table 3: Compact ranking summary (Adult vs Juvenile) — ML models. Lower average rank = better. Ranks computed from point estimates (Accuracy, Precision, Recall, F1, AUC) and Inference Time (IT, lower is better).

Model	MFCC	Acc_rank	Prec_rank	Rec_rank	F1_rank	AUC_rank	IT_rank	AvgRank
MLP	40	2	1	3	2	1	4	2.17
SVM	50	1	2	2	1	3	5	2.33
KNN	60	3	5	1	3	5	1	3.00
XGBoost	60	4	4	4	4	2	3	3.50
RF	50	5	3	6	5	4	6	4.83
Naive Bayes	50	6	6	5	6	6	2	5.17

Normalized Confusion Matrix (KNN on Adult vs Juvenile)



Normalized Confusion Matrix (SVM on Adult vs Juvenile)

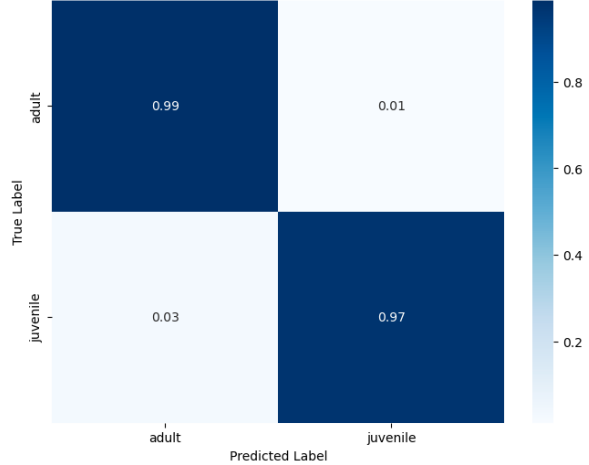


Figure 6: Normalised confusion matrix for KNN (Adult vs. Juvenile). Figure 7: Normalised confusion matrix for SVM (Adult vs. Juvenile).

The results in Table 9 summarises that among DLs, shallow architectures (1D-CNN 1-layer and 1D-DCNN 1-layer) obtain the best average ranks, balancing strong discrimination and comparatively lower inference costs. Some deeper architectures have high precision but suffer from larger inference time; deeper does not uniformly improve the overall rank.

4.3.1. Summary of main findings

Across the classical ML classifiers, SVM achieved the highest point accuracy (96.17%, MFCC=50) and high values on other discrimination metrics (precision, recall, F1) (see Table 6). XGBOOST and MLP are close competitors: XGBOOST attains accuracies above 95% for MFCC=50 and MFCC=60, while the MLP likewise attains its best accuracy at MFCC=60 (95.90%). NB performs substantially worse than the other ML methods ($\approx 82\%$ – 83% accuracy), illustrating that strong parametric assumptions are less appropriate for this task.

Table 4: Performance metrics of DL models (1D-CNN and 1D-DCNN families) for Adult vs. Juvenile classification. Best value per column is **bolded** (for IT lower is better).

Model (layers)	MFCC	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	AUC-ROC (%)	Inference time (ms)
1D-CNN (1 L)	40	97.88	96.00	97.76	96.87	99.80	7537.79
	50	98.15	97.15	97.35	97.25	99.83	16 249.91
	60	97.67	98.11	94.91	96.48	99.79	6463.39
1D-CNN (2 L)	40	97.26	94.65	97.35	95.98	99.72	10 580.45
	50	97.67	96.35	96.74	96.54	99.72	27 973.32
	60	97.40	95.03	97.35	96.18	99.72	9178.96
1D-CNN (3 L)	40	97.47	95.95	96.54	96.24	99.66	13 254.90
	50	97.40	98.09	94.09	96.05	99.59	29 017.33
	60	97.95	98.53	95.32	96.89	99.73	12 854.35
1D-CNN (4 L)	50	95.90	98.00	89.61	93.62	99.58	3318.51
	60	97.67	95.07	98.17	96.59	99.56	13 406.18
1D-DCNN (1 L)	40	98.02	96.39	97.79	97.07	99.81	7478.64
	50	98.02	96.57	97.56	97.06	99.82	8040.94
	60	97.74	96.54	96.74	96.64	99.83	6583.42
1D-DCNN (2 L)	40	98.08	96.77	97.56	97.16	99.70	10 615.49
	50	97.74	95.44	97.96	96.68	99.68	12 299.25
	60	97.47	98.30	94.09	96.15	99.58	9735.64
1D-DCNN (3 L)	40	96.24	98.23	90.43	94.17	99.54	12 310.96
	50	97.67	96.73	96.33	96.53	99.64	14 671.59
	60	96.99	93.40	97.96	95.63	99.63	12 111.80
1D-DCNN (4 L)	50	95.42	91.73	94.91	93.29	99.14	15 311.84
	60	97.13	97.06	94.30	95.66	99.69	12 630.36

Notes: Best values per column are bolded; ties are bolded for all tied entries. Inference time is per-sample wall-clock time measured on the tuning hardware.

Table 5: Compact ranking summary (Adult vs Juvenile) — DL models (1D-CNN / 1D-DCNN families). Lower average rank = better.

Model (layers)	MFCC	Acc_rank	Prec_rank	Rec_rank	F1_rank	AUC_rank	IT_rank	AvgRank
1D-CNN (1 L)	50	1	2	4	1	1	7	2.67
1D-DCNN (1 L)	40	3	6	2	3	2	1	2.83
1D-DCNN (2 L)	40	2	4	3	2	5	2	3.00
1D-CNN (3 L)	60	4	1	7	4	3	4	3.83
1D-CNN (4 L)	60	6	8	1	5	8	5	5.50
1D-CNN (2 L)	50	6	7	5	6	4	8	6.00
1D-DCNN (4 L)	60	8	3	8	8	6	3	6.00
1D-DCNN (3 L)	50	6	5	6	7	7	6	6.17

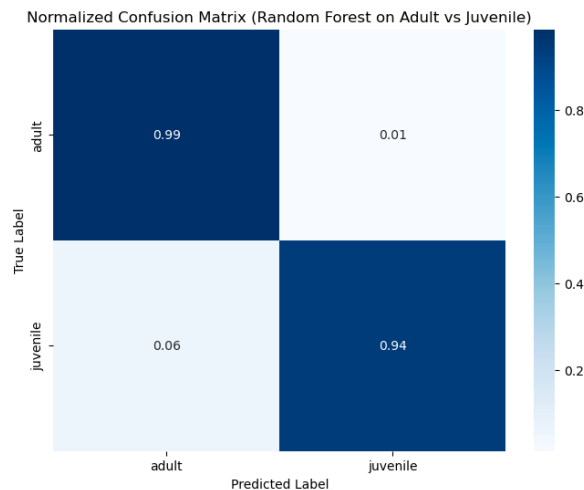


Figure 8: Normalised confusion matrix for RF (Adult vs. Juvenile).

For the DL family, shallow architectures (1–2 hidden layers) generally match or slightly outperform the best ML models on overall accuracy and AUC-ROC, with the 1D-CNN and 1D-DCNN achieving peak accuracies of 95.35% under different MFCC settings (Table 8). AUC-ROC values are uniformly high across DL models (>98%), indicating strong separability between male and female classes.

IT differs substantially between paradigms: ML classifiers (except RF which uses many trees) generally yield low inference latency relative to DL models; DL ITs are several seconds per sample (Tables 6, 8), which may

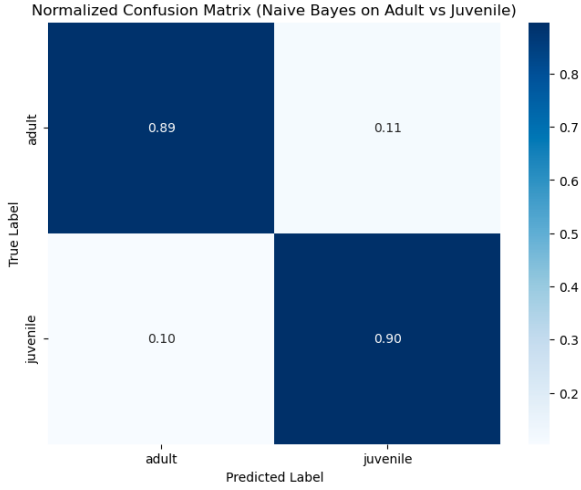


Figure 9: Normalised confusion matrix for NB (Adult vs. Juvenile).

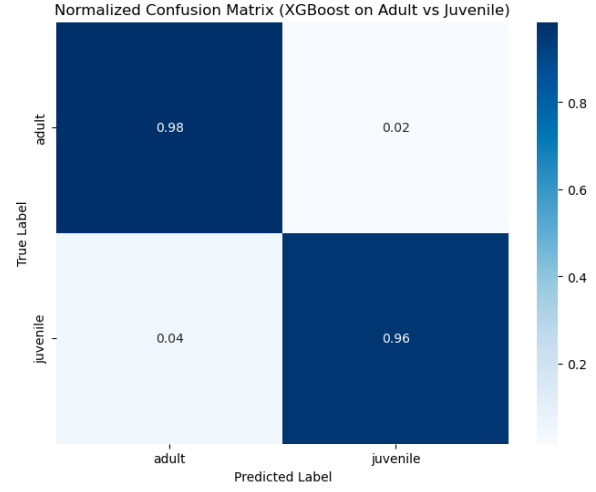


Figure 10: Normalised confusion matrix for XGBoost (Adult vs. Juvenile).

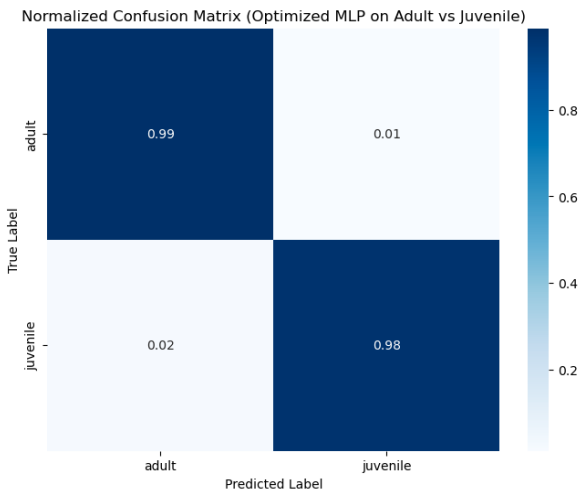


Figure 11: Normalised confusion matrix for MLP (Adult vs. Juvenile).

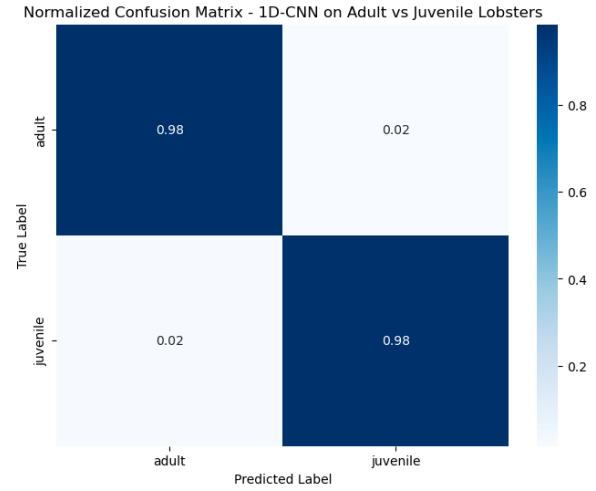


Figure 12: Normalised confusion matrix for 1D-CNN (1 layer) (Adult vs. Juvenile).

constrain real-time deployments.

4.3.2. Interpretation and practical implications

The results indicate three practical points:

1. **Top accuracy vs. operational cost.** SVM and XGBoost produce the best discrimination with modest inference cost (SVM is competitive albeit with higher IT than lightweight ML methods). Deep models yield comparable discrimination but incur substantially higher inference times; this trade-off should weigh in deployment decisions where latency matters.
2. **Model-dependent sensitivity to input dimensionality.** Increasing MFCC count does not universally improve performance. For some models (XGBoost, MLP) modest gains occur when moving from 40 to 50/60 MFCCs; for RF and some CNN variants the effect is neutral or inconsistent. Thus, optimal MFCC dimensionality appears model-dependent and should be identified per algorithm (preferably by nested CV).
3. **Shallow architectures can suffice.** For both 1D-CNN and 1D-DCNN families the best performance is often achieved with one or two hidden layers; deeper architectures yield diminishing returns and substantially larger inference times, suggesting overparameterisation for this dataset.

4.3.3. On inference time and deployment

DL models exhibit inference times in the multi-second range per sample (Tables 8), whereas some ML models (e.g. KNN, NB) have very low latency. When deployment latency is critical (e.g., in-situ monitoring, edge devices), we recommend profiling models under representative hardware and considering:

Table 6: Performance metrics of ML models for Male vs. Female classification (MFCC = 40, 50, 60). Best value per column is **bolded**.

Model	MFCC	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	AUC-ROC (%)	Inference time (ms)
Random Forest	40	93.23	94.58	94.26	94.42	98.08	2559.0391
	50	92.54	93.23	94.59	93.91	98.16	2986.6513
	60	92.54	93.33	94.48	93.90	98.18	2968.7502
XGBoost	40	94.12	94.75	95.61	95.18	98.73	334.6043
	50	95.21	96.16	95.95	96.05	98.81	231.1075
	60	95.28	96.17	96.06	96.11	98.97	221.2673
MLP	40	95.35	96.80	95.50	96.15	99.03	3324.9233
	50	95.14	96.58	95.38	95.98	99.19	2794.5994
	60	95.90	96.83	96.40	96.61	99.16	2260.6717
Naive Bayes	40	81.60	87.52	81.31	84.30	91.04	2.3317
	50	82.56	88.36	82.09	85.11	90.85	2.0922
	60	82.01	88.82	80.52	84.47	89.83	2.3787
SVM	40	95.42	96.07	96.40	96.23	98.86	2586.4148
	50	96.17	96.95	96.73	96.84	99.03	2609.8358
	60	95.96	96.94	96.40	96.67	99.11	2811.5256
KNN	40	94.87	95.93	95.61	95.77	98.85	0.8528
	50	95.49	97.03	95.50	96.25	98.97	0.6002
	60	95.62	97.14	95.61	96.37	99.00	0.6658

Notes: Best per-column values are bolded (for inference time lower is better).

Table 7: Compact ranking summary (Male vs Female) — ML models. Lower average rank = better. Cells shaded indicate rank ≤ 2 (top-2).

Model	MFCC	Acc_rank	Prec_rank	Rec_rank	F1_rank	AUC_rank	IT_rank	AvgRank
KNN	60	3	1	4	3	3	1	2.50
MLP	60	2	3	2	2	1	4	2.33
Naive Bayes	50	6	6	6	6	6	2	5.33
Random Forest	40	5	5	5	5	5	5	5.00
SVM	50	1	2	1	1	2	6	2.17
XGBoost	60	4	4	3	4	4	3	3.67

- Model compression / quantisation for CNNs.
- Pruning or smaller ensembles for tree/boosting methods.
- Lightweight feature sets selected via feature-importance (SHAP) or sequential feature selection.

4.3.4. Limitations

- Only three MFCC dimensionalities were evaluated (40, 50, 60). A broader sweep (e.g., 20–80 in 10-step increments) would better characterise the relationship between feature dimensionality and performance.
- Reported metrics appear as point estimates in Tables 6 and 8.
- Inference time measurements can be confounded by background processes and hardware variability; report hardware specs (CPU/GPU, RAM) and repeat timing with controlled conditions (fixed CPU affinity, disabled turbo boost) or present median times over multiple runs.

4.3.5. Summary and recommendations

SVM and XGBoost are the best performing classical approaches on this dataset; shallow DL architectures are competitive but more computationally expensive at inference. We therefore recommend:

- Reporting uncertainty (CIs) and formal statistical comparisons (McNemar / bootstrap / DeLong) in the final manuscript.
- Prioritising models for deployment based on the accuracy–latency trade-off, and applying calibration corrections when probabilistic outputs are needed.
- Performing additional MFCC dimensionality sweeps and nested CV to robustly identify model-dependent optima.

4.4. Stacking ensemble: proposed pipeline and rationale

Ensemble learning synthesises multiple base learners to produce a single, often more accurate and robust predictor than any individual model Sill (2009); Zhou (2012); Breiman (1996b); Wolpert (1992); Dietterich (2000); Hastie et al. (2001). Following this principle, we propose a stacking ensemble that explicitly leverages complementary model inductive biases to improve discrimination and calibration on bioacoustic classification tasks.

Table 8: Performance metrics of DL models (1D-CNN and 1D-DCNN families) for Male vs. Female classification. Best value per column is **bolded**.

Model (layers)	MFCC	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	AUC-ROC (%)	Inference time (ms)
1D-CNN (1 L)	40	94.60	94.50	96.73	95.60	98.75	8253.8011
	50	94.75	95.40	96.03	95.69	99.04	8069.7830
	60	95.35	95.61	96.82	96.20	99.13	6433.0564
1D-CNN (2 L)	40	94.25	95.17	95.38	95.28	98.67	10445.2045
	50	95.04	96.09	97.78	95.92	98.95	12180.0492
	60	94.94	95.83	95.86	95.83	99.00	8758.4734
1D-CNN (3 L)	40	92.13	90.64	97.07	93.75	98.10	13290.2761
	50	93.61	94.98	94.51	94.72	98.64	15455.8352
	60	94.63	96.05	95.08	95.56	98.78	12078.9611
1D-CNN (4 L)	50	93.66	93.45	96.31	94.86	98.49	15676.7719
	60	93.93	94.69	95.39	95.02	98.62	13249.9224
1D-DCNN (1 L)	40	94.19	94.66	95.83	95.24	98.79	7332.1474
	50	95.35	95.44	96.99	96.20	99.19	7940.4736
	60	95.11	95.70	96.37	96.00	99.21	6515.3026
1D-DCNN (2 L)	40	93.91	96.26	93.58	94.92	98.47	10611.2289
	50	94.87	95.75	95.83	95.76	98.97	12129.8255
	60	95.03	96.01	95.83	95.85	99.05	8682.1848
1D-DCNN (3 L)	40	92.95	93.47	95.05	94.25	98.27	12144.5677
	50	93.98	94.45	95.72	95.08	98.58	14186.0330
	60	94.27	95.35	95.22	95.28	98.63	11781.5477
1D-DCNN (4 L)	50	93.23	98.80	93.92	94.40	98.46	15143.1859
	60	92.89	96.34	91.78	94.00	98.37	12650.9302

Notes: Best per-column values are bolded (for inference time lower is better). Ties are bolded for all tied entries.

Table 9: Compact ranking summary (Male vs Female) — DL models (1D-CNN / 1D-DCNN). Lower average rank = better. Cells shaded indicate rank ≤ 2 (top-2).

Model (layers)	MFCC	Acc_rank	Prec_rank	Rec_rank	F1_rank	AUC_rank	IT_rank	AvgRank
1D-CNN (1 L)	60	1	5	3	1	2	1	2.17
1D-CNN (2 L)	50	3	2	1	3	4	6	3.17
1D-CNN (3 L)	60	5	3	7	5	5	5	5.00
1D-CNN (4 L)	60	7	8	5	7	7	7	6.83
1D-DCNN (1 L)	50	1	6	2	1	1	2	2.17
1D-DCNN (2 L)	60	4	4	4	4	3	3	3.67
1D-DCNN (3 L)	60	6	7	6	6	6	4	5.83
1D-DCNN (4 L)	50	8	1	8	8	8	8	6.83

Design rationale.. We assemble a diverse set of base learners—a Random Forest (RF), an XGBoost classifier (XGBOOST), a support vector machine (SVM) and a 1D convolutional neural network (1D-CNN)—because (i) tree ensembles (RF, XGBoost) provide complementary variance- and bias-reduction mechanisms Breiman (2001a); Chen and Guestrin (2016), (ii) SVMs often perform well on moderate-dimensional engineered feature spaces, and (iii) 1D-CNNs can extract local temporal/spectral patterns directly from short waveform or spectrogram segments. Combining these heterogeneous learners increases the likelihood that different aspects of the signal are captured, which stacking can exploit Wolpert (1992); Breiman (1996b).

Data streams and feature engineering.. The pipeline maintains two parallel feature streams: (1) engineered tabular features (spectral summaries, statistical descriptors and other domain-specific features), used by RF, XGBoost and SVM; and (2) raw or minimally preprocessed short waveform / spectrogram segments used to train the 1D-CNN. Prior to modelling, tabular features are standardised and, where appropriate, decorrelated with PCA; PCA is only applied within training folds to avoid leakage.

Training protocol and leakage prevention.. To avoid target leakage and to produce out-of-sample meta-training data, we use stratified K -fold cross-validation on the training set (here $K = 5$ unless otherwise stated) to generate out-of-fold (OOF) predicted probabilities for each base learner. For each fold:

1. Train each base learner on the $K - 1$ folds (using the same preprocessing pipeline fit only on those folds).
2. Predict class probabilities on the held-out fold and store these OOF probabilities.

After completing K folds we obtain an OOF prediction matrix of shape $(N_{\text{train}}, B \times C)$ where B is the number of base learners and C the number of classes. These OOF features are then used to train a low-capacity meta-learner (logistic regression with L_2 regularisation by default) that maps the base learners’ probability outputs to final predictions. Finally, for deployment each base learner is retrained on the full training set (with preprocessing pipelines refit on full training data) and the meta-learner is used to combine their outputs at test time. This procedure prevents

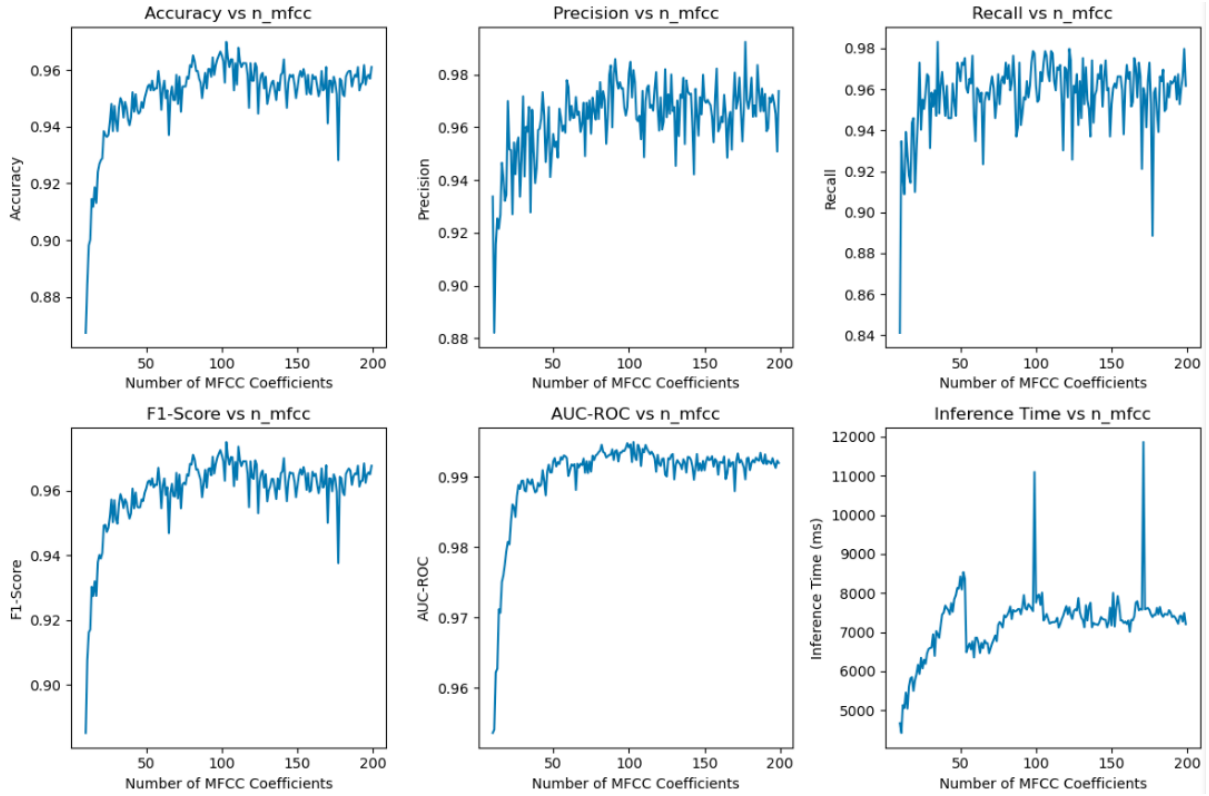


Figure 13: Performance of 1D-CNN (1 hidden layer) as a function of MFCC dimensionality for male vs. female classification. Points show mean performance per metric (accuracy, precision, recall, F1, AUC-ROC, inference time) computed across cross-validation folds.

information flow from test folds into the meta-learner during training and is standard in stacking implementations Wolpert (1992); Breiman (1996b).

Meta-learner and calibration.. We recommend a low-capacity, regularised meta-learner (logistic regression or a small gradient-boosting machine) to avoid overfitting the OOF matrix. Because stacked outputs can be miscalibrated, we emphasise post-hoc calibration diagnostics and correction (e.g., reliability diagrams, Brier score, Platt scaling or isotonic regression) as part of the validation pipeline Breiman (2001a). Where probabilistic interpretation is required, calibration should be reported alongside discrimination metrics.

Practical considerations.. We recommend:

- Hyperparameter optimisation for base learners inside the inner cross-validation loop (or via nested CV) to avoid optimistic bias.
- Parallelised training of base learners and careful resource reporting (CPU/GPU, wall-clock time) to document computational cost.
- Storing OOF predictions, fitted preprocessing objects, random seeds, and software environment (package versions or containers) to ensure reproducibility according to FAIR principles.
- Interpreting ensemble behaviour using feature-importance measures (e.g. SHAP for tree-based learners), and inspecting meta-learner coefficients to understand how base learners are weighted.

Implementation note.. A schematic of the proposed stacking pipeline is shown in Figure 14. The implementation (data preprocessing pipelines, model definitions, and OOF-generation scripts) is provided in the code repository cited in the manuscript Domingos (2025c). We also provide recommended default meta-learner settings (logistic regression with L_2 penalty, regularisation strength selected by CV) and suggest that authors include a brief ablation table in supplementary materials comparing averaging, majority-vote, and stacked meta-learner performance.

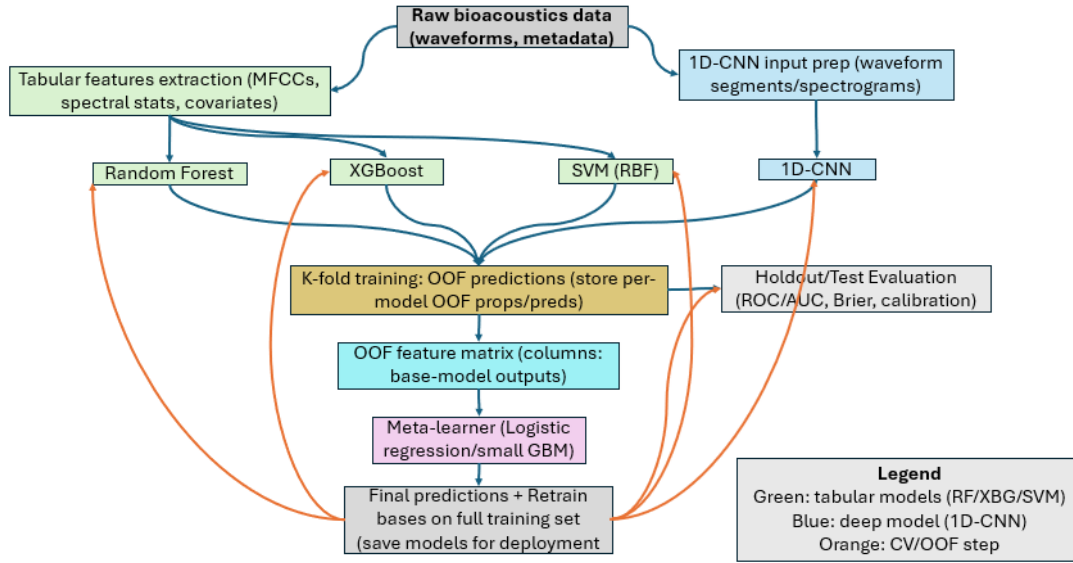


Figure 14: Schematic of the proposed stacking ensemble pipeline. Two parallel feature streams (engineered tabular features and 1D-CNN-ready waveform/spectrogram segments) are processed by diverse base learners. Out-of-fold (OOF) probability outputs from each base learner form the meta-feature matrix used to train a low-capacity meta-learner; final predictions are produced by retraining base learners on the full training set and combining their outputs through the fitted meta-learner.

5. Model architectures and parametrisation

A clear, concise description of model architectures and their parametrisation is essential for scientific reproducibility and interpretability. For each method studied we document the architecture (where applicable), the search space explored during hyperparameter tuning, and the final selected settings. We also report computational costs for hyperparameter optimisation to provide practical guidance for reproducibility and scale-up. The following subsections summarise the optimisation protocol and the best configurations found for each algorithm applied to MFCC-derived features (preprocessed with principal component analysis where indicated).

5.1. KNN parameter optimisation

We optimised the k -nearest neighbours (KNN) classifier via grid search across plausible neighbourhood sizes and distance metrics. Input features were derived from MFCC coefficients; principal component analysis (Principal Component Analysis (PCA)) was applied primarily for decorrelation and to stabilise optimisation — the retained component count and cumulative total explained variance (TEV) are reported below.

Table 10 summarises the results for three initial MFCC configurations. For each case we report TEV after PCA, the neighbour-search algorithm, the selected number of neighbours (k), the Minkowski exponent (p), the weight function used for voting, the mean cross-validated test accuracy (expressed as %), and the wall-clock time required for grid search. The results indicate a modest improvement when increasing the input MFCCs from 40 to 50 (97.72% $\rightarrow$ 97.79%), with negligible additional computational cost; further increasing to 60 MFCCs did not yield further gains. These findings support the use of PCA for compact, decorrelated representations and suggest that modest increases in MFCC dimensionality can improve KNN performance for this dataset.

Table 10: KNN grid-search and PCA settings. PCA was applied to decorrelate MFCC features; TEV is the cumulative total explained variance by the retained components.

Model	# MFCC	Original shape	Shape after PCA	TEV	Algorithm	# Neighbors	p	Weight	Best CV Acc. (%)	Grid time (s)
KNN	40	(7307, 40)	(7307, 40)	1.00	auto	5	1	uniform	97.72	10.93
KNN	50	(7307, 50)	(7307, 40)	0.98	auto	7	2	uniform	97.79	12.10
KNN	60	(7307, 60)	(7307, 40)	0.96	auto	9	1	uniform	97.72	12.21

Notes: ‘Algorithm’ is the neighbour-search method passed to `scikit-learn`’s `KNeighborsClassifier`; ‘ p ’ is the Minkowski exponent (1 = Manhattan, 2 = Euclidean). Best CV accuracy is the mean over cross-validation folds (reported as percent). Grid time is wall-clock tuning time on the reported machine.

5.2. SVM parameter optimisation

We performed a grid search to identify optimal support vector machine (SVM) hyperparameters for classification using MFCC-derived features preprocessed with principal component analysis (PCA). PCA was applied to decorrelate features and reduced each input set to 30 components while preserving a high cumulative total explained variance (TEV). Table 11 summarises the best validation configurations for each initial number of MFCC coefficients. For each case we report TEV after PCA and the hyperparameters that yielded the best cross-validated performance (regularisation parameter C , kernel scale ‘gamma’, and kernel type). Reported grid-search wall-clock times provide an empirical indication of tuning cost. These results indicate the SVM is robust across input dimensionalities, with only modest adjustments to regularisation and kernel scaling required.

Table 11: SVM grid-search and PCA settings. PCA reduced feature dimensionality to 30 components while retaining high TEV.

Model	# MFCC	Original shape	Shape after PCA	TEV	C	gamma	kernel	Grid time (s)
SVM	40	(5845, 40)	(5845, 30)	0.97	10	auto	rbf	24.58
SVM	50	(5845, 50)	(5845, 30)	0.94	10	scale	rbf	23.60
SVM	60	(5845, 60)	(5845, 30)	0.91	1	auto	rbf	26.63

Notes: TEV = total explained variance after PCA. ‘gamma’ values such as ‘auto’/‘scale’ are shown verbatim; numeric values (if used) will align via ‘siunitx’. Grid times are wall-clock seconds measured on the tuning workstation.

5.3. Random Forest parameter optimisation

We performed a grid search to identify robust Random Forest (RF) hyperparameters using MFCC-derived inputs preprocessed by PCA. PCA truncated each input set to 30 components while retaining the majority of variance (high TEV). Table 12 summarises the best hyperparameters discovered for each initial MFCC count.

For each configuration we report TEV after PCA and the RF hyperparameters that maximised validation performance: tree depth (‘max_depth’), minimum samples per leaf and split, number of estimators, and the wall-clock grid-search time. These results reveal the degree of tree complexity and minimal leaf-size regularisation required under the different input dimensionalities.

Table 12: Random Forest grid-search and PCA settings. PCA reduced features to 30 components while preserving most variance.

Model	# MFCC	Original shape	Shape after PCA	TEV	max_depth	min_samples_leaf	min_samples_split	n_estimators	Grid time (s)
RF	40	(5845, 40)	(5845, 30)	0.97	None	1	2	200	61.36
RF	50	(5845, 50)	(5845, 30)	0.94	20	3	2	200	61.78
RF	60	(5845, 60)	(5845, 30)	0.91	None	1	10	200	62.56

Notes: TEV = total explained variance after PCA. ‘None’ indicates unrestricted tree depth. ‘n_estimators’ denotes the number of trees in the forest. Grid times are wall-clock seconds on the tuning machine.

5.4. Naive Bayes parameter optimisation

We performed a focused grid search to determine the Naive Bayes (NB) configuration that yields robust classification performance on MFCC-derived feature sets preprocessed with principal component analysis (PCA). For these experiments PCA was applied to decorrelate the features and then truncated to a fixed 30 components—this choice reflects consistently high total explained variance (TEV) across the tested initial feature counts.

Table 13 summarises the optimisation results for the different initial MFCC dimensions. For each configuration we report TEV after PCA, the number of PCA components retained, mean and standard deviation of the test score (cross-validation), the ranking of the best configuration, and wall-clock grid-search time. The results indicate stable and high predictive performance across input dimensionalities, with minimal tuning cost and low variance in validation performance, supporting the choice of 30 PCA components as a parsimonious and effective representation for NB in this task.

Table 13: NB grid-search and PCA settings. PCA was used to decorrelate features and a fixed 30 components were retained for subsequent classification.

Model	# MFCC	Original shape	Shape after PCA	TEV	# PCA comps	Mean test score	Std test score	Rank	Grid time (s)
Naive Bayes	40	(5845, 40)	(5845, 30)	0.97	30	0.929 600	0.011 571	1	0.28
Naive Bayes	50	(5845, 50)	(5845, 30)	0.94	30	0.935 672	0.010 552	1	0.81
Naive Bayes	60	(5845, 60)	(5845, 30)	0.91	30	0.933 105	0.005 837	1	0.56

Notes: TEV = total explained variance after PCA. Mean and standard deviation refer to the cross-validation test score (reported here to four decimal places). Grid-search times are wall-clock seconds measured on the tuning workstation.

5.5. XGBoost parameter optimisation

We conducted a grid search to identify the optimal XGBOOST hyperparameters for classification using MFCC features preprocessed with principal component analysis (PCA). PCA was used primarily to decorrelate features while preserving the original dimensionality when total explained variance (TEV) remained high.

Table 14 summarises the best-performing hyperparameter configurations for different initial MFCC counts. For each configuration we report TEV after PCA and the hyperparameters that produced the best validation performance: column subsampling ('colsample_bytree') and row subsampling ('subsample') (regularisation), learning rate, maximum tree depth ('max_depth') (model complexity), and the number of estimators. We also report wall-clock grid-search time to give an empirical sense of tuning cost. These results show modest shifts in regularisation and learning-rate settings as the input dimensionality changes, while the number of estimators remained consistent across experiments.

Table 14: XGBOOST grid-search and PCA settings. PCA was applied to decorrelate features while preserving original dimensionality when TEV remained high.

Model	# MFCC	Original shape	Shape after PCA	TEV	colsample_bytree	learning_rate	max_depth	n_estimators	subsample	Grid time (s)
XGBoost	40	(5845, 40)	(5845, 30)	0.97	0.70	0.10	5	300	0.70	28.16
XGBoost	50	(5845, 50)	(5845, 30)	0.94	0.90	0.20	4	300	0.80	27.79
XGBoost	60	(5845, 60)	(5845, 30)	0.91	0.90	0.10	4	300	0.90	24.73

Notes: TEV = total explained variance after PCA. 'colsample_bytree' and 'subsample' control feature- and row-level regularisation respectively; 'n_estimators' denotes the number of boosted trees. Grid times are wall-clock seconds measured on the tuning machine.

5.6. MLP parameter optimisation

We performed a systematic grid search to optimise the multilayer perceptron (MLP) architecture for classification using MFCC features. Principal component analysis (PCA) was applied primarily to decorrelate features; in this study we retained the original feature dimensionality (i.e., PCA served as a rotation rather than a dimensionality reduction) because the cumulative total explained variance (Total Explained Variance (TEV)) remained high.

Table 15 summarises the grid-search results for different initial MFCC feature counts. For each configuration we report the total explained variance after PCA, the hyperparameters that produced the best validation performance (activation, hidden-layer size, learning-rate policy, solver, and regularisation strength α), and the wall-clock time required for the grid search. The selected hyperparameters reflect the trade-offs between representational capacity and regularisation required for our dataset; runtimes provide an indication of the computational cost of tuning.

Table 15: MLP grid-search and PCA settings (PCA used for decorrelation while retaining original dimensionality).

Model	# MFCC	Original shape	Shape after PCA	TEV	Activation	Hidden	LR policy	Solver	alpha	Grid time (s)
MLP	40	(5845,40)	(5845,40)	0.97	tanh	128	constant	adam	0.0010	110.14
MLP	50	(5845,50)	(5845,50)	0.98	relu	64	constant	adam	0.0100	90.02
MLP	60	(5845,60)	(5845,60)	0.99	relu	64	constant	adam	0.0001	80.06

Notes: Shapes shown as (samples, features). TEV = total explained variance after PCA.

5.7. 1D-CNN parameter optimisation with one hidden layer

Table 16 reports the best hyperparameter configurations found for a 1D-CNN with a single hidden layer, when the original MFCC feature sets (40, 50, 60 coefficients) are reduced to 30 components by PCA. For each setting we report the post-PCA feature shape, the total explained variance (TEV), the best-performing hyperparameters discovered by grid search (batch size, epochs, dense units, filters, kernel size, pooling) and elapsed grid-search time.

Across the explored configurations the optimisation consistently selected the same batch size, epoch count and dense-layer width, indicating a robust, compact architecture for this task. Kernel size and optimiser choices exhibited minor variation between MFCC settings, reflecting subtle dependencies of receptive-field design and optimisation dynamics on input dimensionality. Grid-search times (≈ 30 – 36 s) were short, indicating that single-hidden-layer configurations can be tuned efficiently; this makes them attractive for rapid prototyping and for deployment scenarios where compute budgets are limited.

Recommendations. For reproducible and efficient hyperparameter search we recommend:

- using randomized or Bayesian optimisers (e.g., Optuna, Hyperopt) to explore the space more efficiently than exhaustive grid search;

Table 16: 1D-CNN (one hidden layer): PCA + grid-search optimisation results (vertical-rule style).

Model	#MFCCs	Orig. shape	PCA shape	TEV	Batch	Epochs	Dense	Filters	Kernel	Pool	Optimiser	Grid time (s)
1D-CNN (1 HL)	40	(5845, 40)	(5845, 30)	0.97	32	10	128	64	5	2	adam	35.51
1D-CNN (1 HL)	50	(5845, 50)	(5845, 30)	0.94	32	10	128	64	3	2	rmsprop	35.65
1D-CNN (1 HL)	60	(5845, 60)	(5845, 30)	0.91	32	10	128	64	5	2	rmsprop	30.12

- adopting early stopping and short pilot runs to prune poor configurations quickly;
- employing multi-fidelity tuning (subsampling data, reduced epochs) prior to full training;
- logging experiments (MLflow, Weights & Biases or DVC) to ensure reproducibility and to enable warm-starting of subsequent searches.

These practices retain or improve performance while substantially cutting wall-clock time and compute cost, which is particularly important for lifecycle management and operational deployment.

5.8. Parameter optimisation for 1D-CNN with two hidden layers

Table 17 summarises the optimal architectures and training hyperparameters found for the 1D-CNN with two hidden layers when varying the number of MFCC features and applying PCA for dimensionality reduction. The table reports the post-PCA feature shape, total explained variance, best-performing hyperparameters discovered by grid search (batch size, epochs, dense units, filters, kernel sizes, pooling sizes and optimiser), and the elapsed grid-search time in seconds.

The results indicate consistent high performance across settings but also highlight that exhaustive grid search for DL architectures is computationally expensive: the reported grid-search times ($\approx 11,000$ – $11,200$ s) reflect substantial compute. For practical model development we recommend complementary strategies to reduce computation while preserving search effectiveness, such as randomized or Bayesian hyperparameter search, early stopping, progressive resizing, lower-fidelity estimates (reduced epochs or subset of data), and transfer learning with fine-tuning. These approaches can dramatically reduce wall-clock time and cost while finding comparably good hyperparameters.

Table 17: 1D-CNN (two hidden layers): PCA + grid-search optimisation results.

Model	Feature shape (post-PCA)	Total Expl. Var.	Batch	Epochs	Dense	Filters (L1,L2)	Kernel (L1,L2)	Pool (L1,L2)	Grid time (s)
1D-CNN (2 HL)	(5845, 30)	0.97	32	20	64	64, 128	5, 5	2, 2	11114.43
1D-CNN (2 HL)	(5845, 30)	0.94	64	20	128	128, 256	3, 5	2, 2	11112.93
1D-CNN (2 HL)	(5845, 30)	0.91	64	20	256	64, 128	5, 5	2, 2	11223.71

Given the heavy computational burden of exhaustive grid searches for 1D-CNN hyperparameters, recommended approaches entail (1) using randomized or Bayesian optimization, such as Optuna, Hyperopt, instead of full-grid search to explore hyperparameter space more efficiently, (2) employing early stopping and reduced-epoch pilot runs to prune poor configurations quickly, (3) adopting multi-fidelity optimization (shorter runs or smaller data subsets) before committing to full training, (4) leveraging transfer learning or pre-trained backbones where applicable, followed by light fine-tuning, (5) logging and versioning experiments like MLflow, Weights & Biases, or DVC, to ensure reproducibility and enable warm-starting of searches. These actions typically reduce search time substantially while retaining or improving model performance and reproducibility.

5.9. 1D-CNN Architecture

A 1D-CNN is a neural network architecture tailored to one-dimensional sequential data such as time series or audio. A canonical design begins with a 1D input layer and proceeds through a sequence of convolutional blocks: small-kernel 1D convolutional layers (typically with ReLU activation) interleaved with pooling layers for progressive feature aggregation and dimensionality reduction. After the convolutional stack, a Flatten (or global-pooling) layer converts feature maps to a vector that feeds one or more fully connected (Dense) layers with optional Dropout for regularization, finishing with a task-dependent output layer (e.g., sigmoid for binary classification or softmax for multi-class).

In practice, design choices such as kernel size, number of filters per layer, pooling strategy, and the depth of the network determine the trade-off between representational capacity and inference cost. Small kernels (e.g., 3–7 samples) stacked across multiple layers allow the network to learn hierarchical temporal features, while pooling (max or average) controls temporal resolution and reduces compute. For resource-constrained deployments, shallow architectures with fewer filters and modest pooling often provide a favourable balance of accuracy and latency; for research experiments, modest increases in depth and filter count can improve performance but incur higher training and inference cost. Hyperparameters should be selected via cross-validation and evaluated in terms of both predictive metrics and operational constraints (latency, memory, power).

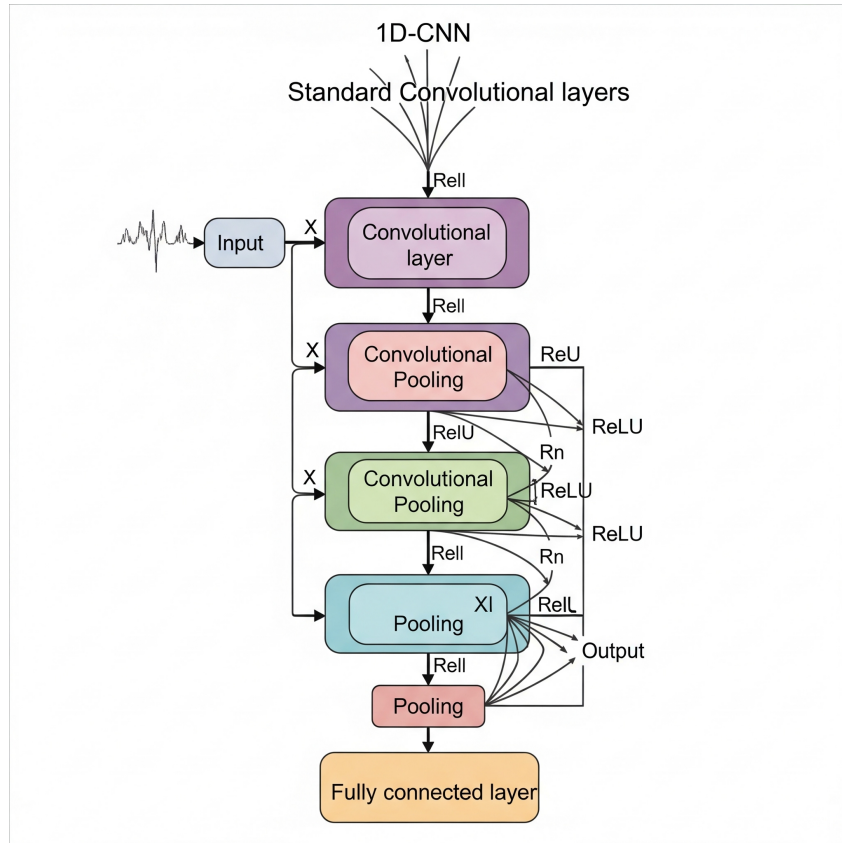


Figure 15: Generic schematic of the 1D-CNN architecture used in this study.

5.10. 1D-DCNN Architecture

A 1D-DCNN extends the 1D-CNN paradigm by using dilated convolutions to enlarge the receptive field without substantially increasing parameter count or reducing temporal resolution. The typical architecture begins with a 1D input layer for sequential data, followed by multiple blocks of dilated 1D convolutional layers (usually with ReLU activation), optionally interleaved with pooling. Dilated convolutions apply filters with a specified *dilation rate*, effectively inserting gaps between filter taps and enabling the network to model long-range dependencies while keeping networks relatively shallow. After the convolutional blocks, a Flatten (or global pooling) layer prepares features for fully connected (Dense) layers with optional Dropout for regularization, concluding with a task-specific output layer (e.g., sigmoid for binary, softmax for multi-class).

Compared with a standard 1D-CNN, the principal distinction is the convolution operator. Standard 1D convolutions process contiguous input segments, while dilated convolutions insert defined gaps (dilation) between kernel elements, permitting a much larger effective receptive field for the same kernel size and depth. Practically, this yields three main advantages for long sequential signals such as audio: (1) the ability to capture long-range context without deep stacking of layers, (2) improved parameter efficiency when modeling extended dependencies, and (3) preservation of temporal resolution because dilated convolutions can reduce the need for aggressive downsampling. These properties make 1D-DCNN particularly suitable for tasks where distant temporal structure is informative but excessive pooling would lose fine-grained cues.

There are trade-offs: dilation patterns must be chosen carefully (e.g., exponential vs linear schedules) to avoid blind spots in coverage, and dilated layers may be more sensitive to hyperparameter settings (kernel size, dilation rates, and regularization). In practice, shallow dilated stacks (one or two dilated blocks) often provide a good balance of performance and inference cost for our lobster audio tasks; further gains are attainable through systematic tuning, judicious use of global pooling, and model-compression techniques for deployment on resource-constrained hardware.

6. Discussion

ML and DL approaches exhibited complementary strengths on the sex and age bioacoustic tasks. Binary classification of adult versus juvenile achieved the strongest results overall (see Tables 2 and 4): the weakest

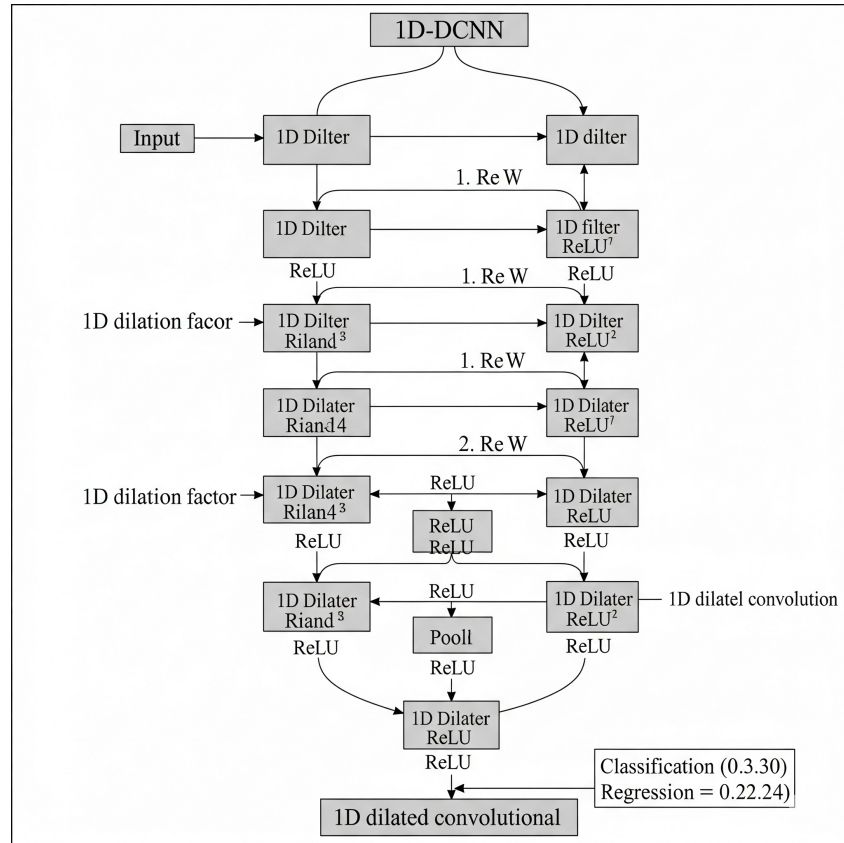


Figure 16: Schematic of the 1D-DCNN architecture used in this study.

performer was NB (89.12%), while the best single model was SVM (98.50%). For male versus female classification the lowest accuracy was also from NB (81.60%), and SVM again attained the highest score (96.17%). A 1D-DCNN with two hidden layers achieved the second-highest age-class accuracy (98.08%), but the difference between 98.50% and 98.08% is not large enough to conclude a clear superiority given the experimental scope and sample counts. Notably, SVM showed greater stability as the number of MFCC coefficients varied (98.29%, 98.50%, 98.29% for 40/50/60 coeffs), whereas the 1D-DCNN (2 hidden layers) followed 98.08%, 97.74%, 97.47% for the same settings. Because only three MFCC settings were explored, these trends are suggestive but not definitive; a denser sweep and graphical analysis of accuracy versus MFCC dimensionality would be required to establish a functional relationship.

Across models, SVM delivered the best balance of accuracy and consistency for the Adult vs Juvenile task (peak accuracy 98.50% and F1 score 99.76%). KNN provided the fastest inference time (0.87 ms) with competitive accuracy (98.08%). XGBOOST and MLP produced strong results ($\approx 97\text{--}98\%$ accuracy; $F1 \approx 97\text{--}98\%$) with inference times typically lower than SVM in our tests. RF performed well (97.26% accuracy; $F1 \approx 95.79\%$), while NB consistently performed worst across metrics despite a low inference time (2.10 ms). DL variants (1D-CNN and the 1D-DCNN family) attained comparable classification accuracy (generally $>97\%$) and very high AUC-ROC values ($>99.5\%$), but at substantially higher inference times than traditional ML models. Increasing the number of hidden layers in DL models tended to increase inference time without reliably improving accuracy; shallow architectures (one or two hidden layers) therefore appear preferable given our data and computational constraints.

For practical, real-time deployments (e.g., PAM or underwater Underwater Internet of Things (UIOT) edge systems Mackridge and Rowe (2018)), model selection must trade predictive performance against latency, power and memory budgets. In this context, compact ML pipelines built on efficient feature extractors (such as MFCC) and lightweight classifiers (e.g., SVM, KNN, or modest tree ensembles) often provide the best operational compromise; DL models can match or slightly exceed accuracy but require model-compression and efficient-inference engineering to be viable on resource-constrained hardware Han et al. (2015); Sze et al. (2017). The optimal architecture will therefore depend on the specific deployment constraints (sampling cadence, available compute, acceptable latency) and on whether probabilistic calibration or marginal improvements in accuracy are prioritized.

Observed performance differences between age and sex classifications likely reflect dataset factors (size, class imbalance, recording quality) and the relative strength of acoustic cues: ontogenetic or age-related signals appear

more salient than sex-specific cues in our tank recordings. Addressing these limitations—by increasing sample sizes, improving annotation quality, and expanding in-situ (natural habitat) recordings—should improve model generalization and clarify the roles of feature dimensionality and architecture depth. Future work should also evaluate sequence models and transformer-style architectures, carry out systematic hyperparameter and feature-selection studies, and perform rigorous cross-region validation to assess transferability Gong et al. (2021); Verma and Berger (2021).

Finally, it is important to situate these technical advances within the broader socio-ecological context. EU fisheries management has a mixed record with respect to overfishing and enforcement, and lobster stocks face region-specific depletion risks, size-structure concerns and climate-driven shifts in distribution and productivity The Pew Charitable Trusts (2021); European Court of Auditors (2022); Centre for Environment, Fisheries and Aquaculture Science (Cefas) (2023); Matić-Skoko et al. (2022). AI-enhanced monitoring can provide high-resolution observational inputs to support management, but technological solutions are complementary tools rather than substitutes for robust governance, compliance and ecosystem-based management. Integrating AI-derived indicators into policy requires transparent methods, reproducible data and explicit attention to the social and institutional dimensions of fisheries governance.

7. Conclusions

In conclusion, while DL models can deliver very high accuracy for lobster sound classification, their substantial training and inference costs — and the engineering required for edge deployment — often make traditional ML approaches (e.g., SVM, KNN, XGBOOST) the more pragmatic choice for many real-world bioacoustic applications Strubell et al. (2019); Sze et al. (2017); Chen and Guestrin (2016). Efficient feature extractors such as MFCC remain powerful front-ends; when paired with compact classifiers they frequently achieve an attractive trade-off between predictive performance and computational cost Davis and Mermelstein (1980b); Han et al. (2015); Strubell et al. (2019).

Our results demonstrate that *Homarus gammarus* rasping and clicking are classifiable with AI models and that both conventional ML and DL approaches are viable for age and sex discrimination under controlled (tank) conditions. For operational use within PAM and underwater UIOT systems, model selection must balance discrimination against latency, power and memory constraints; model-compression and efficient-inference techniques can mitigate but not eliminate the overhead of large DL architectures Han et al. (2015); Sze et al. (2017). Observed performance differences between adults vs juveniles and males vs females likely reflect dataset limitations (size, imbalance, recording quality) and weaker sex-specific acoustic cues; expanding representative sampling and improving in-situ annotation will be important next steps. Future work will explore sequence models and transformer-style architectures and perform rigorous cross-region validation to assess transferability Gong et al. (2021); Verma and Berger (2021).

Implementing the proposed stacking-ensemble pipeline would bring several concrete benefits. By combining diverse base learners (trees, kernel methods, and neural networks) into a stacked meta-learner, the pipeline typically improves discrimination and reduces variance relative to single models Wolpert (1992); Breiman (1996b); Zhou (2012). Boosting and bagging-based components (e.g., XGBOOST, RF) capture complementary signal structure in noisy bioacoustic data and tend to generalize more robustly across heterogeneous conditions Chen and Guestrin (2016); Dietterich (2000). The OOF-based stacking design prevents information leakage and enhances reproducibility, while a regularized meta-learner often yields better-calibrated probabilistic outputs — a critical asset when model outputs feed management thresholds. These gains come with practical trade-offs: fold-wise training and CNN retraining increase compute and require disciplined evaluation, model registry and monitoring (drift detection, scheduled re-training). In our judgment, the modest operational overhead is justified by the improvements in accuracy, robustness and calibrated uncertainty that support management decisions.

Finally, to enhance transparency and reproducibility we have published the data and code with ISO-compliant metadata and included ensemble baselines to strengthen predictive comparisons. Together, these measures advance the practical applicability of AI-enhanced bioacoustics for lobster management and conservation.

8. Acknowledgements

We are profoundly grateful to Murray McBay and Co., particularly their Lobster Shop in Scotland, for their generous support in allowing us to record lobster sounds within their facility. Their invaluable assistance in sorting female, male, juvenile, and adult lobsters was crucial to this research. We extend our sincere thanks to Dr. Nicola Khan for her pivotal role in establishing the connection with Murray McBay and Co.

The work reported in this article contributes to the EnviroBrain Impact Case Study.

References

- Araújo, M.B., Guilhaumon, F., 2018. Climate change impacts on the distribution of coastal lobsters. *Marine Biology* 165, 1–14. URL: <https://link.springer.com/article/10.1007/s00227-018-3441-9>, doi:doi:10.1007/s00227-018-3441-9.
- Bardeli, R., Wolff, D., Kurth, F., Koch, M., Tauchert, K.H., Frommolt, K.H., 2010. Detecting bird sounds in a complex acoustic environment and application to bioacoustic monitoring. *Pattern Recognition Letters* 31, 1524–1534. URL: <https://www.sciencedirect.com/science/article/pii/S0167865509002487>, doi:doi:https://doi.org/10.1016/j.patrec.2009.09.014. pattern Recognition of Non-Speech Audio.
- Barroso, V., Xavier, F., Ferreira, C., 2023. Applications of machine learning to identify and characterize the sounds produced by fish. *ICES Journal of Marine Science* 80, 1854–1867. doi:doi:https://doi.org/10.1093/icesjms/fsad126.
- Bharath, S., et al., 2025. Underwater sound classification using learning-based methods: A review. *Expert Systems with Applications* 240, 122236. URL: <https://www.sciencedirect.com/science/article/pii/S0957417424013654>, doi:doi:10.1016/j.eswa.2024.122236.
- Bouwma, P.E., Herrnkind, W.F., 2009. Sound production in caribbean spiny lobster *panulirus argus* and its role in escape during predatory attack by octopus *briareus*. *New Zealand Journal of Marine and Freshwater Research* 43, 3–13. URL: <https://doi.org/10.1080/00288330909509977>, doi:doi:10.1080/00288330909509977, arXiv:https://doi.org/10.1080/00288330909509977.
- Breiman, L., 1996a. Bagging predictors. *Machine Learning* 24, 123–140. doi:doi:https://doi.org/10.1023/A:1018054314350.
- Breiman, L., 1996b. Stacked regressions. *Machine Learning* 24, 49–64. doi:doi:https://doi.org/10.1007/BF00117832.
- Breiman, L., 2001a. Random forests. *Machine Learning* 45, 5–32. doi:doi:https://doi.org/10.1023/A:1010933404324.
- Breiman, L., 2001b. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science* 16, 199–231. doi:doi:10.1214/ss/1009213726.
- Briones-Fourzán, P., Domínguez-Gallegos, R., Lozano-Álvarez, E., 2014. Aggressive behaviour of spotted spiny lobsters (*panulirus guttatus*) in different social contexts: the influence of sex, size, and missing limbs. *ICES Journal of Marine Science* 72, i155–i163. URL: <https://doi.org/10.1093/icesjms/fsu219>, doi:doi:10.1093/icesjms/fsu219, arXiv:https://academic.oup.com/icesjms/article-pdf/72/suppl_1/i155/31227438/fsu219.pdf.
- Centre for Environment, Fisheries and Aquaculture Science (Cefas), 2023. Lobster (*Homarus gammarus*) — Stock Status Report 2023. Technical Report. Cefas. URL: <https://www.cefaz.co.uk/data-and-publications/does/international-ices-and-national-uk-fish-stock-and-shellfish-stock-data-from-2020-assessment-year/>. accessed: 2025-09-17.
- Chen, T., Guestrin, C., 2016. Xgboost: A scalable tree boosting system, in: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA. pp. 785–794. doi:doi:10.1145/2939672.2939785.
- Council of the European Union, 2025. The eu’s common fisheries policy and its goals. <https://www.consilium.europa.eu/en/policies/common-fisheries-policy-and-its-goals/>. Accessed 19 September 2025.
- Davis, S., Mermelstein, P., 1980a. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28, 357–366. doi:doi:10.1109/TASSP.1980.1163420.
- Davis, S., Mermelstein, P., 1980b. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28, 357–366. doi:doi:10.1109/TASSP.1980.1163396.
- Dietterich, T.G., 2000. Ensemble methods in machine learning, in: *Multiple Classifier Systems*, Springer. pp. 1–15.

- Domingos, F., 2025a. Lobster sound dataset. URL: <https://data.mendeley.com/datasets/wsm4vfxvsf>, doi:doi:10.17632/WSM4VFXVSF.
- Domingos, F., 2025b. lobster_soundscodes. URL: https://github.com/N0736086/lobster_soundscodes.
- Domingos, F.P.F., 2025c. stack_ensemble_pipeline – stack_ensemble_pipeline.py. URL: https://github.com/N0736086/stacking-ensemble-pipeline/blob/main/stack_ensemble_pipeline.py. gitHub repository file; accessed 2025-09-18.
- Domingos, F.P.F., Lotfi, A., Ihianle, I.K., Kaiwartya, O., Machado, P., 2024. Underwater communication systems and their impact on aquatic life—a survey. *Electronics* 14, 7. doi:doi:<https://doi.org/10.3390/electronics14010007>.
- European Commission, Directorate-General for Maritime Affairs and Fisheries, 2025. Common fisheries policy (cfp): Overview. https://oceans-and-fisheries.ec.europa.eu/policy/common-fisheries-policy-cfp_en. Accessed 19 September 2025.
- European Court of Auditors, 2022. Special report: Illegal, unreported and unregulated fishing — EU action and spending to prevent imports of illegal fishing products. Technical Report. European Union. URL: <https://op.europa.eu/webpub/eca/special-reports/illegal-fishing-20-2022/en/>. accessed: 2025-09-17.
- Gong, Y., Chung, Y.A., Glass, J., 2021. Ast: Audio spectrogram transformer. Interspeech 2021 Proceedings / arXiv preprint URL: <https://arxiv.org/abs/2104.01778>.
- Han, S., Mao, H., Dally, W.J., 2015. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint [abs/1510.00149](https://arxiv.org/abs/1510.00149). URL: <https://arxiv.org/abs/1510.00149>.
- Hastie, T., Tibshirani, R., Friedman, J., 2001. The elements of statistical learning: data mining, inference, and prediction. Springer, New York. doi:doi:10.1007/978-0-387-21606-5.
- Huang, S., Tang, J., Dai, J., Wang, Y., 2019. Signal status recognition based on 1dcnn and its feature extraction mechanism analysis. *Sensors* 19, 2018. doi:doi:<https://doi.org/10.3390/s19092018>.
- Hwang, J., et al., 2019. Unveiling microplastic contamination in seafood: Source, fate and biological effects. *Marine Pollution Bulletin* 146, 110–118. URL: <https://www.sciencedirect.com/science/article/pii/S0025326X25008896>, doi:doi:10.1016/j.marpolbul.2019.05.017.
- Islam, Y., Das, S.C., Chowdhury, M.J.U., 2025. The role, trends, and applications of machine learning in undersea communication: A bangladesh perspective. arXiv preprint [arXiv:2503.00669](https://arxiv.org/abs/2503.00669) doi:doi:<https://doi.org/10.48550/arXiv.2503.00669>.
- Jézéquel, Y., Bonnel, J., Coston-Guarini, J., Guarini, J.M., Chauvaud, L., 2018. Sound characterization of the european lobster homarus gammarus in tanks. *Aquatic Biology* 27, 13–23. doi:doi:<https://doi.org/10.3354/ab00692>.
- Jézéquel, Y., Chauvaud, L., Bonnel, J., 2020a. Spiny lobster sounds can be detectable over kilometres underwater. *Scientific Reports* 10, 7943. doi:doi:<https://doi.org/10.1038/s41598-020-64830-7>.
- Jézéquel, Y., Coston-Guarini, J., Chauvaud, L., Bonnel, J., 2020b. Acoustic behaviour of male european lobsters (homarus gammarus) during agonistic encounters. *Journal of Experimental Biology* 223, jeb211276. doi:doi:10.1242/jeb.211276.
- Jézéquel, Y., Jones, I.T., Bonnel, J., Chauvaud, L., Atema, J., Mooney, T.A., 2021. Sound detection by the american lobster (homarus americanus). *Journal of Experimental Biology* 224, jeb240747. doi:doi:<https://doi.org/10.1242/jeb.240747>.
- Kerkhofs, R., 2024. To hum or not to hum: analyzing and provoking sound production in the american lobster (homarus americanus). Bowdoin Library .
- Kikuchi, M., Akamatsu, T., Takase, T., 2015. Passive acoustic monitoring of japanese spiny lobster stridulating sounds. *Fisheries Science* 81, 229–234. doi:doi:<https://doi.org/10.1007/s12562-014-0835-6>.
- Looby, A., Vela, S., Cox, K., Riera, A., Bravo, S., Davies, H.L., Rountree, R., Reynolds, L.K., Martin, C.W., Matwin, S., et al., 2023. Fishsounds version 1.0: A website for the compilation of fish sound production information and recordings. *Ecological Informatics* 74, 101953. doi:doi:<https://doi.org/10.1016/j.ecoinf.2022.101953>.

- Mackridge, A., Rowe, P., 2018. A practical approach to using statistics in health research: From planning to reporting. John Wiley & Sons.
- Malfante, M., Mars, J.I., Dalla Mura, M., Gervaise, C., 2018. Automatic fish sounds classification. *The Journal of the Acoustical Society of America* 143, 2834–2846. doi:doi:https://doi.org/10.1121/1.5036628.
- Matić-Skoko, S., Pavičić, M., Šepić, J., Janeković, I., Vrdoljak, D., Vilibić, I., et al., 2022. Impacts of sea bottom temperature on cpue of european lobster (*Homarus gammarus*). *Frontiers in Marine Science* 9, 891197. URL: <https://www.frontiersin.org/articles/10.3389/fmars.2022.891197/full>, doi:doi:10.3389/fmars.2022.891197. accessed: 2025-09-17.
- McGarvey, R., Punt, A.E., Matthews, J.M., Feenstra, J.E., Gardner, C., Burch, P., Hartmann, K., Linnane, A., 2015. Comparing size-limit and quota policies to increase economic yield in a lobster fishery. *Canadian Journal of Fisheries and Aquatic Sciences* 72, 1292–1305. doi:doi:https://doi.org/10.1139/cjfas-2014-0405.
- Niu, H., Li, X., Zhang, Y., Xu, J., 2023. Advances and applications of machine learning in underwater acoustics. *Intelligent Marine Technology and Systems* 1, 1–28. URL: <https://link.springer.com/article/10.1007/s44295-023-00005-0>, doi:doi:10.1007/s44295-023-00005-0.
- Noda, J.J., Travieso, C.M., Sánchez-Rodríguez, D., 2016. Automatic taxonomic classification of fish based on their acoustic signals. *Applied Sciences* 6. URL: <https://www.mdpi.com/2076-3417/6/12/443>, doi:doi:10.3390/app6120443.
- Patek, S.N., 2002. Squeaking with a sliding joint: mechanics and motor control of sound production in palinurid lobsters. *Journal of Experimental Biology* 205, 2375–2385. doi:doi:https://doi.org/10.1242/jeb.205.16.2375.
- Phillips, B.F., Wahle, R.A., Ward, T.M., et al., 2015. Lobsters: ocean icons in changing times. *ICES Journal of Marine Science* 72, i1–i5. URL: https://academic.oup.com/icesjms/article/72/suppl_1/i1/622676, doi:doi:10.1093/icesjms/fsv002.
- Radhakrishnan, E., et al., 2020. Ecology and global distribution pattern of lobsters, in: *Lobsters: Biology, Fisheries and Aquaculture*. Springer, pp. 87–112. URL: https://link.springer.com/chapter/10.1007/978-981-32-9094-5_5, doi:doi:10.1007/978-981-32-9094-5_5.
- Rountree, R.A., et al., 2023. Applications of machine learning to identify and characterize the acoustic signatures of fishes: a review. *ICES Journal of Marine Science* 80, 1854–1875. URL: <https://academic.oup.com/icesjms/article/80/7/1854/7246570>, doi:doi:10.1093/icesjms/fsad085.
- Shabani, S., Kamio, M., Derby, C.D., 2009. Spiny lobsters use urine-borne olfactory signaling and physical aggressive behaviors to influence social status of conspecifics. *Journal of Experimental Biology* 212, 2464–2474. doi:doi:https://doi.org/10.1242/jeb.026492.
- Sill, J., 2009. Feature-weighted linear stacking. preprint arXiv:0911.0460. arXiv.
- Slaney, M., 1998. Auditory Toolbox: A MATLAB Toolbox for Auditory Modeling Work (Version 2). Technical Report. Interval Research Corporation / Apple Computer. Palo Alto, CA. URL: <https://engineering.purdue.edu/~malcolm/interval/1998-010/>.
- Staaterman, Claverie, Patek, 2010. Disentangling defense: the function of spiny lobster sounds. *Behaviour* 147, 235–258. doi:doi:https://doi.org/10.1163/000579509X12523919243428.
- Strubell, E., Ganesh, A., McCallum, A., 2019. Energy and policy considerations for deep learning in nlp, in: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3645–3650. doi:doi:10.18653/v1/P19-1355.
- Sze, V., Chen, Y.H., Yang, T.J., Emer, J., 2017. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE* 105, 2295–2329. doi:doi:10.1109/JPROC.2017.2761740.
- Taylor, J., Patek, S., 2010. Crustacean seismic communication: heard but not present. *The use of vibrations in communication: properties, mechanisms and function across taxa*. Research Signpost, Trivandrum, Kerala , 9–23.
- The Pew Charitable Trusts, 2021. Lessons from implementation of the eu’s common fisheries policy. <https://www.pewtrusts.org/en/research-and-analysis/reports/2021/03/lessons-from-implementation-of-the-eus-common-fisheries-policy>. Accessed: 2025-09-17.

- Verma, P., Berger, J., 2021. Audio transformers: Transformer architectures for large scale audio understanding. adieu convolutions. arXiv preprint arXiv:2105.00335. URL: <https://arxiv.org/abs/2105.00335>. accessed: 2025-09-17.
- Weilgart, L., 2018. The impact of ocean noise pollution on fish and invertebrates. Report for OceanCare, Switzerland <https://api.semanticscholar.org/CorpusID:49364618>.
- Wolpert, D.H., 1992. Stacked generalization. Neural Networks 5, 241–259.
- Yadav, R., et al., 2024. The unseen threat of the synergistic effects of microplastics and heavy metals in aquatic ecosystems. Current Pollution Reports 10, 1–20. URL: <https://link.springer.com/article/10.1007/s40726-024-00298-7>, doi:doi:10.1007/s40726-024-00298-7.
- Yue, H., Zhang, L., Wang, D., Wang, Y., Lu, Z., 2017. The classification of underwater acoustic targets based on deep learning methods, in: 2017 2nd International Conference on Control, Automation and Artificial Intelligence (CAAI 2017), Atlantis Press. pp. 526–529. doi:doi:10.2991/caai-17.2017.118.
- Zhou, Z.H., 2012. Ensemble Methods: Foundations and Algorithms. Chapman & Hall/CRC, Boca Raton, FL.

Appendix A. Specifications of the Raspberry Pi 3 Model B

Table A.18 provides a comprehensive technical overview of the Raspberry Pi 3 Model B. It details various hardware and software aspects, categorized into sections like CPU, RAM, Ports (covering video, audio, USB, Ethernet, wireless, Bluetooth, GPIO, camera, display, storage, and power), and Software (listing the primary OS, desktop environment, included software, and other compatible operating systems). Essentially, it serves as a detailed spec sheet for the Raspberry Pi 3 Model B.

Table A.18: Specifications of the Raspberry Pi 3 Model B

Category	Specification
CPU	
SoC	Broadcom BCM2837
Architecture	64-bit ARM Cortex-A53
Core Count	Quad-core (4 cores)
Clock Speed	1.2 GHz
RAM	
Size	1GB LPDDR2 SDRAM
Ports	
Video Output	Full-size HDMI (up to 1080p60), Composite (3.5mm TRRS)
Audio Output	Analog (3.5mm TRRS), Digital (HDMI)
USB Ports	4 x USB 2.0
Ethernet	10/100 BaseT RJ45
Wireless	802.11 b/g/n Wi-Fi (2.4 GHz)
Bluetooth	Bluetooth 4.1 Classic, Bluetooth Low Energy (BLE)
GPIO	40-pin header (26 GPIO, Power, Ground)
Camera Interface (CSI)	15-pin MIPI CSI
Display Interface (DSI)	15-pin MIPI DSI
Storage	MicroSD Card Slot
Power Input	Micro USB (5V/2.5A recommended)
Software (Commonly Equipped With)	
Primary OS	Raspberry Pi OS (Debian-based Linux)
Desktop Environment	LXDE
Included Software	Chromium Browser, LibreOffice (optional), Python, Scratch, etc.
Other Compatible OS	Ubuntu MATE/Server, Windows 10 IoT Core, LibreELEC, RetroPie, RISC OS, etc.

Appendix B. Hardware and Software resources

Table B.19 provides a concise overview of the key resources (both hardware and software) utilized in a project involving dataset collection and processing. It lists each resource, such as Raspberry Pi 3, Hydrophone, Python 3.12.3, PyAudio, Laptop (PC), ThinkPad, Keras, Scikit-learn, and TensorFlow, alongside a brief description of its role or function in the project. The table essentially acts as a quick reference guide to the technological stack employed. The details of the computer used is Table B.20

Table B.19: Hardware and Software resources for dataset collection and processing

Resources	Description
Raspberry Pi 3	Hardware and software equipped with CPU, Ports, RAM
Hydrophone	Hardware, underwater audio sensor
Python 3.12.3	Software, development environment
PyAudio	Software, library resource for recording sound.
Laptop (PC), ThinkPad	Housing Python, SSH, networking capabilities.
Keras	High-level neural networks API written in Python
Scikit-learn	Python library for ML, built on top of NumPy, SciPy, and Matplotlib, offer some basic tools for building and training deep neural networks
TensorFlow	An open-source ML framework developed by Google, designed to build and train various types of neural networks

Table B.20: Technical specifications of the Lenovo ThinkPad X1 Carbon Gen 13 Aura Edition (PF-2D95WH).

Category	Specification
Performance	Intel Core Ultra (Series 2) processor on Intel vPro, Evo Edition platform; integrated AI engine with up to 13 TOPS AI performance; optimized performance per watt for extended battery life.
AI functions	Local AI features including enhanced video conferencing, document automation, email management, scheduling, improved responsiveness, and enhanced data privacy.
Display	14-inch OLED, 2.8K (2880 × 1800) resolution, 120 Hz refresh rate, 400–500 nits brightness, 100% DCI-P3 color gamut, DisplayHDR True Black 500 certified.
Memory and storage	Up to 64 GB soldered LPDDR5x (8400 MT/s) RAM; up to 2 TB PCIe Gen5 SSD.
Design and construction	Lightweight design <1 kg; chassis built from recycled carbon fiber, magnesium, and plastic; MIL-STD-810H certified durability.
Security	ThinkShield suite: discrete TPM (dTPM), biometric authentication, AI-enhanced threat detection.
Usability	Haptic TouchPad; TrackPoint with double-tap menu for audio/video settings; Dictation Toolbar with speech-to-text support.
Connectivity	USB-C Thunderbolt 4, USB-A (full-size), and other modern I/O options.
Power	Rapid Charge technology; compact GaN charger for fast charging.

Appendix C. Hardware: Hydrophone Features

Table C.21 provides a detailed breakdown of the key characteristics and capabilities of a specific hydrophone device.

Table C.21: Hydrophone specification — Aquarian Audio Model H2d (3.5 mm cable termination)

Specification	Model H2d (Aquarian Audio)
Manufacturer	Aquarian Audio
Model	H2d
Termination	3.5 mm cable termination (note: termination only—electrical/acoustic specifications identical to other termination options)
Sensitivity	-172 dB re: 1 V/ μ Pa (± 4 dB, 20 Hz–4 kHz). Approx. sensitivity at 100 kHz: -210 dB re: 1 V/ μ Pa.
Useful range	<10 Hz to >100 kHz (not measured above 100 kHz).
Polar response	Omnidirectional (horizontal).
Operating depth	<80 m.
Output impedance	1.1 k Ω (typical).
Power (current)	0.7 mA (typical).
Minimum PIP requirement	2 V, 0.7 mA.
Dimensions	25 mm $\times$ 46 mm.
Mass	105 g.
Specific gravity	5.3.

Note: The “3.5 mm cable termination” refers only to the connector termination. All acoustic and electrical specifications in the table apply to the H2d hydrophone irrespective of connector type unless otherwise specified. Replace any placeholder numbers if you have manufacturer updates.