Iby and Aladar Fleischman
Faculty of Engineering
Tel Aviv University

הפקולטה להנדסה
ע"ש איבי ואלדר פליישמן
אוניברסיטת תל-אביב

Seismic signal analysis of Spalax

Project Number: **2977**

Student: Michael Zuck ID: 315780932

Student: Tomer Makmal ID: 206591976

Supervisor: Professor Yossi Yovel

Project Carried Out at: 3/3/2025

Table of Contents

1.Abstract

Seismic communication in Israeli blind mole-rats (genus *Spalax*), which inhabit underground burrow systems, has attracted considerable scientific attention in recent years. These Spalaxes generate vibrational signals by drumming their heads against tunnel walls or the ground surface, enabling them to exchange information despite living in sealed, solitary burrows. The present study focuses on analyzing, classifying, and interpreting these seismic signals by integrating advanced signal processing and machine learning techniques.

In this work, we utilized an existing dataset of seismic recordings obtained from multiple *Spalax* individuals. We applied a preprocessing pipeline that involved noise filtering, normalization, and the extraction of salient features such as root mean square (RMS) amplitude, zero-crossing rate, dominant frequency, and spectral entropy. We then employed various supervised machine learning algorithms, with a Random Forest classifier as the primary model, to address two core tasks: (1) identifying the sender of a given signal (i.e., *which* Spalax produced it) and (2) determining the recipient (i.e., the intended target of the signal). Additionally, we integrated a One-Class Support Vector Machine (SVM) for novelty detection, thereby allowing the system to recognize seismic signals produced by untrained individuals or novel environmental conditions.

Our empirical results indicate that the methodology achieves an average classification accuracy of approximately 65% for sender and recipient identification, contingent on the specific train-test splits and hyperparameter configurations. Furthermore, when combined with the novelty detection module, the system can detect previously unobserved class of an accuracy of approximately 50%. These findings substantiate the viability of automated seismic signal classification in *Spalax*, while underscoring the potential for further enhancements through refined feature engineering and more advanced model architectures.

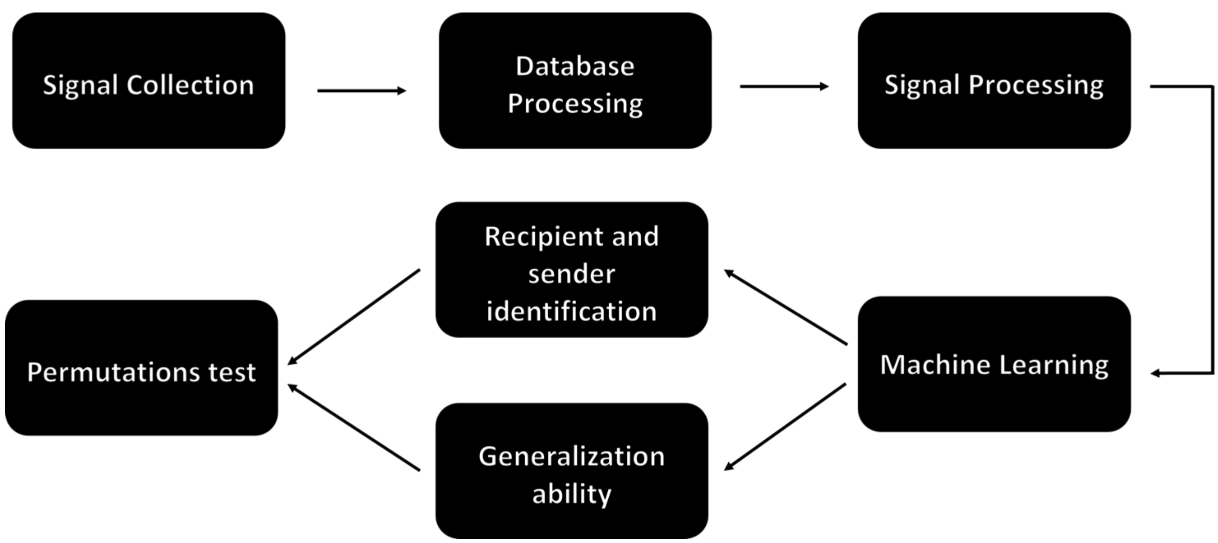

Figure 1- Block Diagram

2. Introduction and Theoretical Background

2.1 Seismic Communication in Subterranean Mammals

Subterranean mammals such as spalax produce and detect low-frequency seismic signals for intraspecific communication. In particular, the Middle Eastern blind mole-rat (commonly referred to by the Hebrew acronym "חולד ארצישראלי") has evolved specialized head-tapping or drumming behaviors on the tunnel walls, generating seismic waves that propagate through the soil. These vibrations convey crucial information about territory, reproductive status, or alarm signals [1]. Early studies by Rado et al. (1998) observed that blind Spalaxes can recognize conspecifics through their unique seismic signatures, indicating that the temporal and spectral features of these signals carry individual-specific cues [2].

Spalaxes live in isolated, underground burrow systems with minimal light and limited air-borne acoustic transmission. Under these conditions, low-frequency seismic waves can travel efficiently—thus providing a reliable communication channel [3]. Understanding how these animals generate, transmit, and decode seismic signals not only adds insight into subterranean biology but also inspires technological applications (e.g., soil-based sensors, underground communication devices) [4].

2.2 Digital Signal Processing

Because we are working with vibrations recorded by geophones or similar sensors, the raw data often contain noise from environmental sources (e.g., footsteps, wind, mechanical vibrations). Moreover, individual signal segments may differ significantly in amplitude, duration, and frequency content. Digital signal processing (DSP) provides systematic methods to denoise signals and extract relevant temporal and spectral features.

2.2.1 Filtering and Noise Reduction

A fundamental step in analyzing seismic signals is filtering out unwanted frequency components (noise) while retaining the core part of the Spalax's head-tap or drumming waveform. We use a digital low-pass filter to remove high-frequency noise typically not produced by Spalaxes:

- **Butterworth Low-Pass Filter.** The Butterworth filter is widely favored for its maximally flat passband [5]. Given a desired cutoff frequency f_c and sampling rate f_s, the normalized cutoff is:

$$1. w_c = \frac{f_c}{f_s/2}$$

where $f_s/2$ is the Nyquist frequency. For an Nth-order analog Butterworth filter, the magnitude-squared frequency response $|H(\Omega)|^2$ is often expressed as:

$$2. |H(\Omega)|^2 = \frac{1}{1 + (\frac{\Omega}{\Omega_c})^{2N}}$$

where Ω is the angular frequency and Ω_c is the cutoff frequency in radians per second. In digital implementations, a bilinear transform or similar method is used to convert this analog prototype into a discrete-time filter.

A low-order Butterworth design (e.g., 3rd to 5th order) typically provides smooth attenuation without sharp ripples in the frequency response [6]. If x[n] is the discrete-time signal, then the low-pass filtering operation can be viewed mathematically as a convolution of x[n] with the filter's impulse response. In practice, forward-backward filtering is commonly applied to minimize phase distortion.

By attenuating frequencies above f_c , we isolate the principal frequency range used by Spalaxes to communicate. This range typically lies below a few hundred hertz 111, although precise cutoff settings may vary depending on soil conditions and species-specific head-tapping behavior.

2.2.2 Fourier Transform and Spectral Analysis

After basic noise reduction, spectral features are computed via the Discrete Fourier Transform (DFT) [7]. For a sampled time-domain signal x(n) of length N, the DFT is defined as:

$$3.\ X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi \frac{k}{N} n} \quad ,k = 0,1,\ldots,N\text{-}1.$$

In practice, we use the Fast Fourier Transform (FFT) for efficiency [8]. From the FFT, we compute:

- **Power**
  The total signal power in the frequency domain can be approximated by the sum of squared magnitudes of the FFT, often normalized by NNN. In code,

$$4.\ \text{power} = \frac{\sum_{k=0}^{N-1} |X(k)|^2}{K}$$

- **Dominant Frequency**: the frequency bin *k* at which $|X(k)|$ is maximized.

$$f_{dom} = \left| f_{req}[argmax \mid X(k) \mid] \right|$$

  where f_{req} is the array of frequency values corresponding to each FFT bin.

- **Spectral Entropy**:

$$5.\ H = -\sum_{k=0}^{N-1} P(k) \log_2 \lfloor P(k) \rfloor$$

  Where,

$$6.\ P(k) = \frac{|X(k)|^2}{\sum_{m=0}^{N-1} |X(m)|^2}$$

  is a normalized power spectrum. High entropy corresponds to a more uniform spectral distribution, whereas low entropy indicates concentrated energy in fewer frequencies [9].

- **Spectral Centroid:** The "center of mass" of the spectrum:

$$7.\ \text{Centroid} = \frac{\sum_{k=0}^{N-1} f_K |X(k)|^2}{\sum_{k=0}^{N-1} |X(k)|^2}$$

- **Spectral Spread:** The second central moment around the spectral centroid:

$$8.\ \text{Spread} = \sqrt{\frac{\sum_{k=0}^{N-1} (f_K - centroid) \cdot |X(k)|^2}{\sum_{k=0}^{N-1} |X(k)|^2}}$$

- **Rolloff**: capturing the "center of mass" of the power spectrum, how widely the energy is distributed, and the frequency below which a certain percentage (e.g., 85%) of energy accumulates [10].

- **Spectral Flatness:** Compares the geometric mean to the arithmetic mean of the power spectrum, indicating how "tone-like" (peaky) or "noise-like" (flat) the distribution is:

$$9.\ \text{Flatness} = \frac{exp(\sum_{k=0}^{N-1} \log_{10}|X(k)|^2)}{\frac{1}{N}\sum_{k=0}^{N-1}|X(k)|}$$

- **Harmonic-to-Noise Ratio (HNR):** Estimates the ratio of harmonic components to broadband noise:

$$10.\ \text{HNR} = 10\log_{10}(\frac{\text{Harmonic}}{\text{Noise}})$$

  where "Harmonics" and "Noise" are approximated by splitting the FFT bins above and below a small frequency threshold (e.g., 0.1 Hz in the code). A higher HNR indicates a more periodic, tonal signal.

2.2.3 Time-Domain Feature Extraction

Time-domain measures also play a crucial role in identifying individual-specific or context-specific signatures. Common metrics include:

- **Root Mean Square (RMS) Amplitude**:

$$11.\ RMS = \sqrt{\frac{1}{N}\sum_{n=0}^{N-1} x^2(n)}$$

  which correlates with signal power.

- **Zero-Crossing Rate (ZCR)**: the rate at which the signal changes sign, often indicative of higher-frequency content or noise bursts.

$$12.\ ZCR = \frac{1}{N}\sum_{n=0}^{N-1}[\,x(n)\ \cdot x(n-1) < \ 0]$$

- **Energy**
  The total signal energy in the time domain:

$$13.\ \text{Energy} = \frac{1}{N}\sum_{n=0}^{N-1} x^2(n)$$

- **Skewness and Kurtosis**: higher-order statistical moments to quantify asymmetry and peakiness of the waveform distribution [11].

- **Peak-to-Peak Amplitude**: a measure of dynamic range, potentially useful if different individuals generate different head-tap intensities.

- **Crest Factor:** Defined as the ratio of peak amplitude to RMS:

$$14.\ \text{Crest Factor} = \frac{\text{Peak Amplitude}}{RMS}$$

  Signals with very sharp peaks relative to their overall energy will exhibit higher crest factors.

These extracted features (time-domain and frequency-domain) form a feature vector that we use as input to machine learning models.

2.3 Machine Learning Approaches

Once we have a set of features that describe each seismic signal, we employ supervised learning to classify the signals (e.g., "Which individual Spalax produced this vibration?" or "Which recipient is being signaled?"). We also employ novelty detection methods to identify classes or signals that were not observed during training.

2.3.1 Supervised Classification

Random Forest (RF).
Random Forest is an ensemble classifier introduced by Breiman [12]. It grows multiple decision trees on random subsets of the data and features, and then averages their predictions to reduce variance and mitigate overfitting. Key hyperparameters include:

1. **Number of Trees** ($n_{estimators}$). More trees generally improves stability but increases computation.
2. **Max Depth**. The maximum depth of each tree. A deeper tree can learn more complex patterns but risks overfitting if unregularized.
3. **Minimum Samples per Split/Leaf**. Constrains leaf-size to prevent the tree from fitting every noise fluctuation.

The final output in a classification scenario is a majority vote across all trees. Random Forests handle high-dimensional data well, are robust to outliers, and can provide feature-importance estimates [13].

Support Vector Machines (SVM).
Although not always shown in final results here, SVMs remain a classical approach to classification. They seek a decision boundary that maximizes the margin between classes in a high-dimensional feature space [14]. Typical hyperparameters include the choice of kernel (e.g. RBF, polynomial) and regularization constant C. The RBF kernel introduces a γ parameter, controlling the influence of individual training samples.

2.3.2 Novelty Detection and One-Class SVM

When a new signal is received that does not belong to any known class (i.e., an untrained or "novel" individual/recipient), a conventional supervised classifier often misclassifies it as the nearest known label [15]. To mitigate this, we integrate a One-Class SVM for novelty detection [16]. Trained only on "inlier" classes, the One-Class SVM encloses the inlier data in a high-dimensional boundary. Samples lying outside that boundary are flagged as outliers (or "Unknown"). The main hyperparameters are:

- $\boldsymbol{\nu}$: an upper bound on the fraction of outliers in the training data; also a lower bound on the fraction of support vectors.
- $\boldsymbol{\gamma}$: kernel coefficient for the radial basis function (RBF) kernel, dictating model flexibility.

By combining a standard Random Forest with One-Class SVM, we can first check if an incoming signal is novel. If it is not flagged, we classify it using Random Forest. If it is flagged, we label it "Unknown" or set it aside for further analysis.

2.4 Hyperparameter Tuning

Across these machine learning methods, hyperparameters critically influence performance. We typically tune them via cross-validation or randomized grid search [18]. For Random Forest, we search over:

- {n estimators, max depth, min samples split, min samples leaf, bootstrap, max_features}

And for One-Class SVM, we adjust:

- {ν, γ}

Proper hyperparameter choices help balance overfitting (where the model memorizes training data noise) and underfitting (where the model is too simple to capture true patterns).

3. Project Design and Simulation

3.1 Project Goals and Requirements

The overarching goal of this project is to develop an end-to-end system for analyzing seismic signals from subterranean rodents (specifically Spalaxes), in order to:

1. **Extract Distinct Features**
   Generate robust features (time-domain and frequency-domain) that capture the unique characteristics of seismic tapping signals emitted by individual rodents.
2. **Identify Sender**
   Classify or recognize which rodent (sender) produced the signal. This requires robust machine-learning algorithms (e.g., Random Forests, Support Vector Machines) that can handle variability in signal waveforms and environmental noise.
3. **Identify Recipient**
   Determine the intended recipient rodent based on the structure of the recorded signal. This involves mapping each signal to a specific individual.
4. **Facilitate Novelty Detection**
   Enable the system to detect a rodent identity (sender or recipient) that it was not trained on. Such a feature is critical in real-world monitoring scenarios where new individuals may appear, or existing rodents might exhibit altered signal patterns.

To meet these goals, the system must satisfy the following functional and performance requirements:

- **Signal Processing Requirements**
  - Ability to filter raw signals and enhance signal-to-noise ratio.
  - Capability to extract meaningful features in both time and frequency domains, including higher-order statistics (e.g., skewness, kurtosis).

- **Machine Learning Requirements**
  - Classification accuracy for identifying sender identity above 60% on unseen test data.
  - Capability to generalize to new, untrained rodent signals (i.e., novelty detection).
  - Tolerance to class imbalance via oversampling (e.g., SMOTE, ADASYN) or class-weighting methods.
- **System Integration Requirements**
  - A structured approach for data storage (database of signals labeled by known identities).
  - A modular pipeline that integrates seamlessly:
    data ingestion → feature extraction → classification → detection.

3.2 Initial Simulations and Preliminary Tests

Early in the development process, we conducted a series of small-scale simulations and empirical tests to assess different machine-learning models and refine the signal-processing pipeline. These preliminary investigations shaped our ultimate choice of a Random Forest classifier as the core model and guided decisions about filter configuration, feature selection, and hyperparameter optimization. The following subsections summarize the major insights gained during these trials:

1. **Data Subsets and Exploratory Analysis**
   - We began by extracting a limited subset of the seismic signals—on the order of a few hundred samples—to keep computations tractable. This subset was carefully balanced to include multiple individuals (senders) and recipients so that we could probe the initial classification feasibility in both tasks (sender-ID and recipient-ID).
   - Exploratory data analysis (EDA) included plotting raw waveforms, computing descriptive statistics (e.g., means, variances), and generating correlation matrices. This inspection helped confirm that the chosen time- and frequency-domain features (e.g., root mean square, zero-crossing rate, dominant frequency,

etc.) indeed exhibited meaningful variability across different individuals.

2. **Filter Characterization**
    - We experimented with low-pass cutoff frequencies ranging from 300 Hz to 1000 Hz to find a balance between removing high-frequency noise and preserving core signal energy. Empirical observations suggested that Spalax drumming signals concentrate much of their energy below 500 Hz, so a cutoff slightly above that range (around 500 Hz) emerged as optimal.
    - Filter orders between 3 and 6 were tested for the Butterworth filter. A 5th-order filter provided a sufficiently steep roll-off with minimal phase distortion when implemented via forward-backward filtering (filtfilt). This delivered consistent improvements in subsequent classification experiments by reducing noise while retaining the Spalax tap signatures.

3. **Feature Engineering and Selection**
    - In addition to canonical features such as RMS amplitude and dominant frequency, higher-order statistics (skewness, kurtosis) and spectral shape descriptors (entropy, centroid, rolloff) were evaluated for predictive power.
    - Preliminary classification runs showed that including a mix of both time-domain and frequency-domain features boosted performance compared to using any single category in isolation. We thus consolidated these into a standard 20-feature set for further tests.

4. **Comparative Testing of Classification Models**
    - **Support Vector Machine (SVM):** We initially tried a multi-class SVM (with both linear and RBF kernels) for classifying the sender or recipient. While the RBF SVM performed reasonably well on smaller subsets (accuracies around 60% in cross-validation), it proved computationally intensive to tune γ\gammaγ and C on larger datasets.
    - **Neural Networks:** We also ran preliminary tests using feedforward neural networks with one or two hidden layers. However, without extensive hyperparameter tuning or large labeled datasets, overfitting emerged as a significant issue. Performance gains over SVM were inconsistent.
    - **Random Forest:** Early experiments revealed that a Random Forest classifier—using 50 to 100 trees—offered stable accuracy (roughly 60–70% on sender-ID tasks). Its ensemble strategy reduced variance, made training straightforward, and included natural handling of feature importance. Adding more trees (up to 150 – 200) often improved robustness without incurring excessive computation times.

5. **Class Imbalance and Oversampling**
    - Many real-world biosignals exhibit class imbalance (some Spalaxes or recipient labels have far fewer samples). In these pilot studies, minority classes were at risk of being misclassified.
    - We tested data-level balancing techniques such as Borderline-SMOTE and ADASYN, both of which synthesize new minority-class samples. This helped the classifiers generalize better to less-represented individuals, raising the recall for minority classes by several percentage points.

6. **Novelty Detection Trials**
    - To handle "new" Spalaxes (those not present in the training set), we integrated a One-Class SVM. Preliminary tests showed that when ν\nuν was set too high (e.g., 0.1–0.2), legitimate signals from known rodents were sometimes erroneously flagged as outliers. Lowering ν\nuν to approximately 0.05 reduced false alarms while still enabling detection of unseen individuals.
    - These results confirmed the feasibility of using a two-stage pipeline: One-Class SVM for novelty detection followed by a Random Forest for classifying known senders/recipients.

7. **Key Findings for Final System Design**
    - **Random Forest** consistently offered a favorable balance between accuracy, speed, and interpretability. Its robustness to outliers and ease of hyperparameter tuning (via randomized grid search) made it preferable to SVM or small neural networks for the bulk of the classification work.

- **Feature Diversity** (both time-domain and frequency-domain) was critical. Removing or replacing certain spectral descriptors (e.g., spectral entropy, rolloff) significantly degraded performance in pilot runs, affirming the importance of multi-faceted feature sets.
- **Oversampling** of minority classes (especially using a combination of Borderline-SMOTE and ADASYN) proved essential for maintaining reasonable recall on less-represented rodent IDs and recipients.

By the end of these preliminary simulations, we had established that the combination of (1) a 5th-order low-pass Butterworth filter, (2) a 20-dimensional feature set capturing both time and spectral characteristics, and (3) a Random Forest classifier with hyperparameter tuning would form the core of our final system. These decisions were further validated in larger-scale testing and, ultimately, in our integrated pipeline for both sender-ID and recipient-ID tasks.

3.3 Implementation and Block Diagram

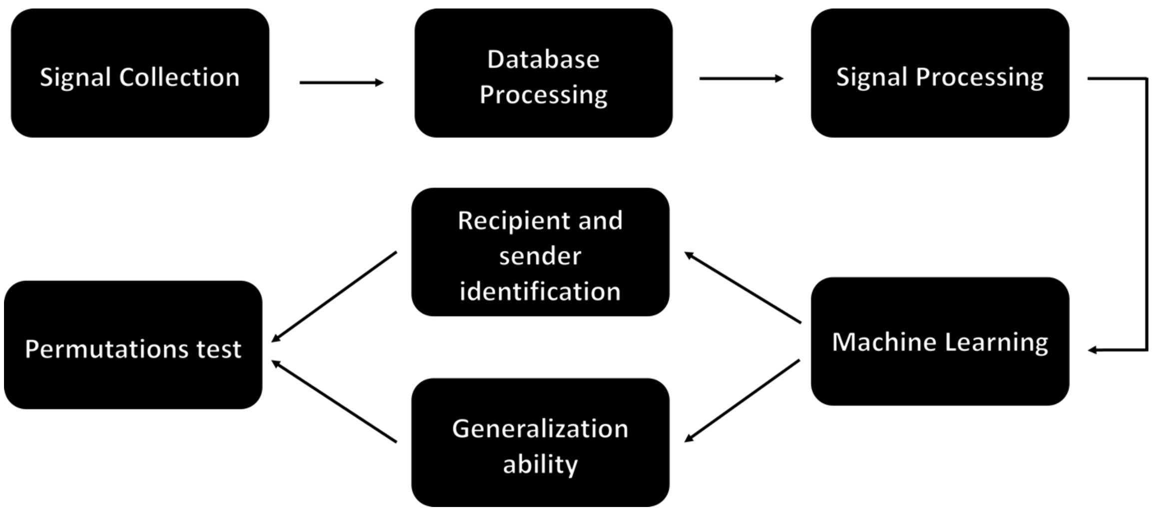

Figure 1- Block Diagram

1. Signal Collection

In the first block, seismic signals are gathered via geophones placed in the Spalaxes' labs burrows. in our project this data has already been recorded and stored as .wav files, the conceptual step involves:

1. **Sensor Setup** – Geophones are installed in or near the burrow system.
2. **Continuous/Triggered Recording** – A data-acquisition system captures vibrations, saving each seismic "tap" or "drum" event in .wav format.

By the time the signals enter our pipeline, they are represented as digitized waveforms stored in a structured directory . Each subdirectory may correspond to a particular sender, recipient.

2. Database (Data) Processing

Following collection, the .wav files are organized in subfolders denoting either a sender identity or a recipient context .The data processing step in our scripts is responsible for:

1. **Directory Traversal**
   Code blocks systematically walk through each subfolder in the directory, reading every .wav file found.
2. **Data Structuring**
   Each file's raw samples, sampling rate, and associated metadata are combined into dictionaries keyed by subfolder.

3. **Label Parsing**
   - In sender.py, the function parse_sender extracts the Spalax's ID (e.g., "BMR2") from the folder name.
   - In recipient.py, the function parse_recipient identifies which Spalax is the intended recipient (e.g., "BMR3" from a folder named "BMR2_vs_BMR3_28").

This organizational phase ensures that all signals are correctly labeled and accessible for subsequent filtering, feature extraction, and training/test partitioning.

3. Signal Processing

Once the signals are loaded, the Signal Processing block applies filtering, normalization, and basic transformations to improve signal quality. In our implementation:

1. **Noise Filtering**
   - We use a low-pass Butterworth filter (order = 5) with a cutoff around 500 Hz (butter_lowpass and lowpass_filter functions) to remove high-frequency components not typically associated with Spalax drumming.
   - Forward-backward filtering is employed to minimize phase distortions.
2. **Normalization**
   - A simple amplitude-based normalization is sometimes applied (dividing by the minimum value if it is nonzero) to keep waveforms within a reasonable range.
   - This step appears, for instance, in process_signal within sender.py or recipient.py, ensuring consistent amplitude scales across different recordings.
3. **Segmentation (Implicit)**
   Although not shown as a separate script, segmentation (if needed) can be done either manually (cutting out the tap events) or via triggered detection. In practice, each .wav is assumed to contain a relatively isolated signal event.

At the conclusion of this block, the pipeline outputs cleaned and filtered signals ready for feature extraction and machine learning.

4. Machine Learning: Feature Extraction and Classification

The core computational block addresses both feature extraction and classification. Our code centralizes feature extraction logic in functions such as extract_features, which produce a 20-dimensional vector for each signal:

- **Time-Domain Metrics**
  - Root Mean Square (RMS), Zero-Crossing Rate (ZCR), Energy, Peak-to-Peak Amplitude, Skewness, Kurtosis, etc.
- **Frequency-Domain Metrics** (via FFT)
  - Dominant Frequency, Spectral Entropy, Spectral Centroid, Spectral Spread, Spectral Rolloff, Harmonic-to-Noise Ratio, etc.

Once features are extracted, the pipeline moves into classification:

1. **Train/Test Split & Oversampling**
   - Scripts like prepare_dataset_by_sender (in sender.py) or prepare_dataset_by_recipient (in recipient.py) split the data into training and testing sets (commonly 70% training, 30% testing).
   - To handle class imbalance, we apply Borderline-SMOTE followed by ADASYN to synthetically augment minority classes, thereby improving recall for underrepresented Spalax IDs.

2. **Random Forest Training**
   - We instantiate and tune a RandomForestClassifier via randomized hyperparameter search (RandomizedSearchCV), searching over parameters such as the number of estimators, maximum depth, and bootstrap usage.
   - The best model is then retrained on the (potentially oversampled) training set, as shown in the final lines of train_model in both sender.py and recipient.py.

3. **Evaluation**
   - Accuracy, precision, recall, and F1-scores are computed on the held-out test set.
   - Confusion matrices are plotted (ConfusionMatrixDisplay) to visualize misclassifications across different Spalax IDs.

By the end of this block, the system has a trained model capable of classifying new seismic signals according to either sender or recipient.

5. Recipient and Sender Identification

Although classification steps for sender and recipient share the same fundamental logic (feature extraction and Random Forest), our project organizes them separately:

- **sender.py**: Identifies which Spalax individual emitted the seismic tap.
- **recipient.py**: Infers which Spalax individual was the intended target of that tap.

6. Generalization Ability (Novelty Detection)

To handle entirely new Spalaxes—those not present in training—we augment the Random Forest with a One-Class SVM for novelty detection. This two-stage approach is showcased in sender_generalization.py and recipient_generalization.py:

1. **Inlier Model**
   - One-Class SVM is trained solely on data from known individuals (labels present in the training set).
   - Hyperparameters (ν and γ) control how tightly or loosely the model encloses these “normal” samples.

2. **Novelty Check**
   - For each incoming feature vector, we first query the One-Class SVM.
     - If flagged as an outlier, the pipeline labels it as “Unknown.”
     - Otherwise, it proceeds to the trained Random Forest for a known-class prediction.
3. **Group-Level Detection**
   - In scripts like group_level_novelty_test, we further aggregate signals from each new or known group to decide if an entire cluster of recordings is novel. This approach helps reduce false alarms on an individual-sample basis, by majority voting within a group (i.e., if more than a threshold fraction of signals in that group are outliers, we label the group as “Unknown.”).

This mechanism enhances the generalization ability of our classifier by preventing the forced misclassification of signals from novel individuals.

7. Permutation Test

Finally, the Permutation Test block statistically validates the significance of our classification accuracy. The procedure is:

1. **Baseline Accuracy**
   - Compute the actual cross-validation accuracy of the trained model on real labels.
2. **Permutation Resampling**
   - Shuffle the labels n times (e.g., 1000 times), recomputing cross-validation accuracy on each permuted set.
3. **p-Value Computation**
   - Compare the real accuracy to the distribution of permuted accuracies. The fraction of permutations that exceed or match the real accuracy is the p-value.

4. Analysis of Results

This section provides a detailed examination of the classification outcomes obtained for both the *sender* (i.e., the individual Spalax that produced the seismic signal) and the *recipient* (i.e., the intended target of the signal). We discuss the overall classification performance, highlight class-specific metrics, interpret the confusion matrices, and comment on the influence of factors such as class imbalance, data limitations, and feature extraction choices.

4.1 Sender Identification

4.1.1 Overall Accuracy and Class-Specific Performance

For the sender-identification task, the Random Forest classifier attained an overall test-set accuracy of approximately 65%. The classification report indicates that:

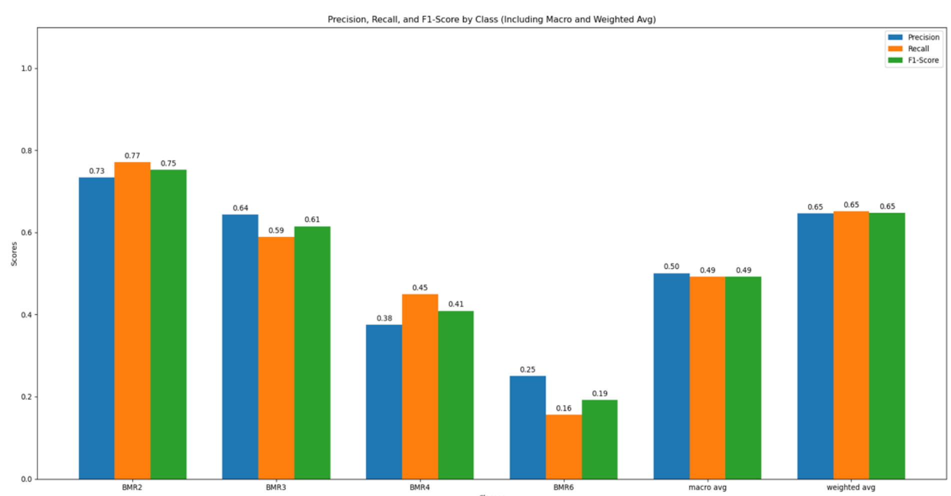

Figure 3- result Sender

- BMR2 achieved a precision of 0.73 and a recall of 0.77, resulting in an F1-score of 0.75.
- BMR3 yielded a precision of 0.64, recall of 0.59, and F1-score of 0.61.
- BMR4 presented lower performance, with a precision of 0.38, recall of 0.45, and F1-score of 0.41.
- BMR6, which had notably fewer training samples, showed a precision of 0.25, recall of 0.16, and F1-score of 0.19.

These findings underscore the presence of class imbalance in the dataset, as both BMR4 and BMR6 exhibit relatively poor classification metrics. Although Borderline-SMOTE and ADASYN were employed to mitigate class imbalance, the small number of original observations for these classes likely contributed to their modest performance. Furthermore, individuals such as BMR2 and BMR3 benefited from larger sample sizes, enabling the model to learn more discriminative patterns for those senders.

4.1.2 Confusion Matrix Observations

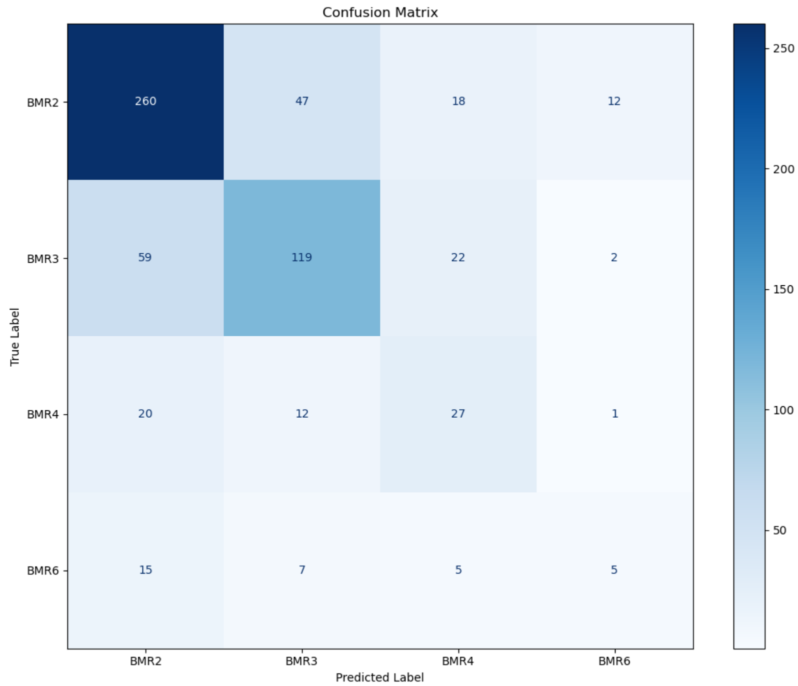

Figure 4- Confusion Matrix Sender

The confusion matrix for sender identification highlights that:

- BMR2 is more frequently classified correctly, suggesting the presence of distinctive temporal or spectral patterns in its head-tapping signals.
- BMR4 and BMR6 tend to be misclassified as BMR2 or BMR3, indicating overlapping feature representations with those majority classes.

4.1.3 Summary of Sender-Identification Insights

The 65% classification accuracy for sender identification demonstrates the feasibility of automatically distinguishing some individuals' seismic signals. However, inconsistencies between classes suggest that additional improvements could be achieved by (1) increasing the number of training samples for minority classes, (2) refining the feature set to capture more nuanced signal characteristics, and (3) potentially exploring advanced data augmentation techniques to better represent under-sampled signals.

4.2 Recipient Identification

4.2.1 Overall Accuracy and Class Distribution

For recipient identification, the best Random Forest model reached an overall test-set accuracy of 61%. A review of the class-specific results reveals:

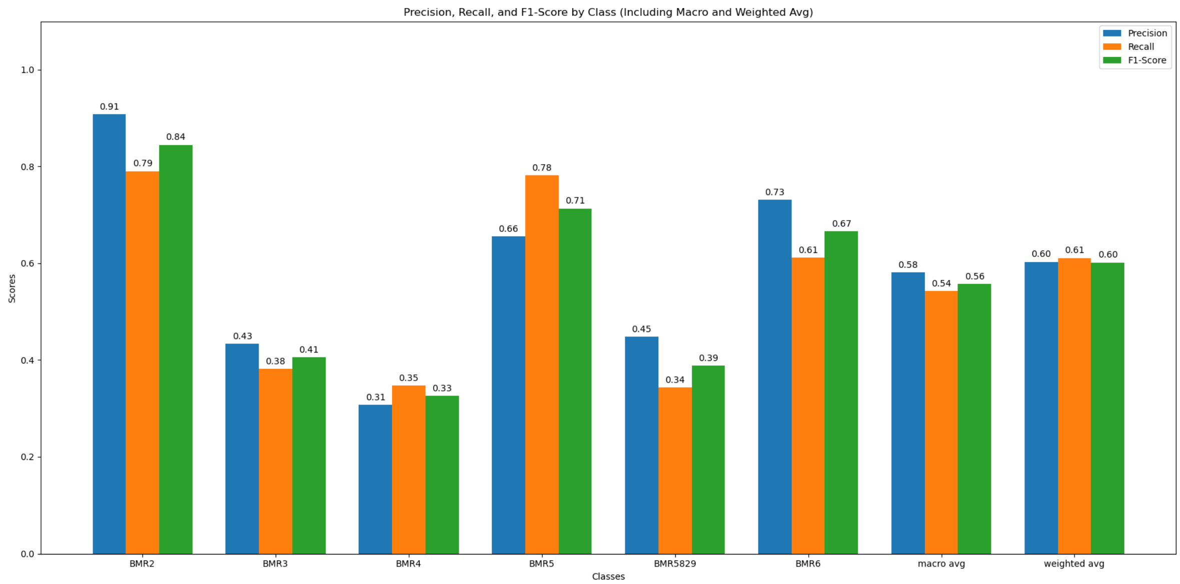

Figure 5- result Recipient

- BMR2 attained relatively high precision (~0.91) and recall (~0.79), yielding an F1-score of 0.84.
- BMR5, which was represented by a substantial number of recordings, exhibited a precision of 0.66, a recall of 0.78, and an F1-score of 0.71.
- Classes such as BMR4 and BMR3, each supported by fewer training examples, achieved lower F1-scores, reflecting more frequent misclassifications and limited feature differentiation.

Unlike the sender-identification task, recipient classification may inherently hinge on subtler aspects of the waveform—such as minor variations in tapping amplitude, spectral content, or timing cues linked to social context. Furthermore, the smaller volume of training samples for certain recipients led to significant class imbalance, thereby affecting precision and recall.

4.2.2 Confusion Matrix Observations

The confusion matrix for recipient classification confirms that:

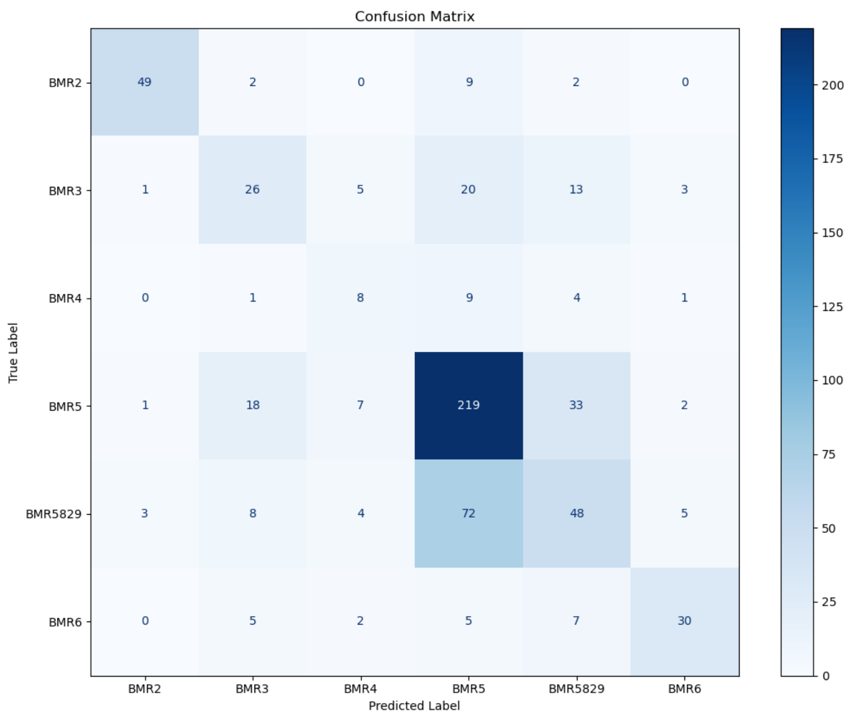

Figure 6- Confusion Matrix Recipient

- Misclassifications frequently involve minority classes being labeled as majority classes (e.g., BMR2 or BMR5).
- BMR2 retains a strong diagonal (correct predictions), suggesting its seismic signature is relatively distinctive in the dataset.

4.2.3 Summary of Recipient-Identification Insights

While the 61% accuracy indicates that recipient classification is feasible, it remains more challenging than identifying senders. The lower performance among certain recipient classes underscores the need for additional data collection, especially for classes with limited observations, and suggests that further refinement of the feature set or incorporation of contextual data (e.g., inter-burrow distances or social pairing information) may bolster model performance.

4.3 Influence of Oversampling and Class Weights

In both tasks, oversampling techniques such as Borderline-SMOTE and ADASYN were applied to counteract class imbalance. Although these methods improved recognition rates for minority classes, they did not fully resolve performance disparities across classes. This limitation arises because synthetic oversampling cannot entirely capture the genuine diversity of real-world signals when the underlying data for certain classes remains limited.

The Random Forest classifier's built-in mechanism for incorporating class weights further aimed to attenuate misclassification penalties for minority classes. Nonetheless, performance gains were modest, highlighting that fundamental constraints—namely, insufficient real data and overlapping feature distributions—continue to pose challenges in classification.

4.4 Permutation Test: Statistical Validation

Beyond raw accuracy scores, a crucial step in validating any classification model is determining whether the observed performance **s**ignificantly exceeds what might be expected by chance. To this end, we performed a permutation test on the sender-classification model (and similarly, a parallel test can be run for the recipient model). The procedure was as follows:

1. **Train/Test Setup**
   We used the same Random Forest model trained on the (oversampled) training data.
2. **Cross-Validation Accuracy (Actual)**
   Using 5-fold cross-validation on the training set, we computed the model's mean accuracy on the true labels.
3. **Permutation Resampling**
   We then shuffled (permuted) the labels randomly while keeping the feature vectors fixed. The model was evaluated again via cross-validation for each permutation, and the resulting accuracy was recorded. This process was repeated for a set number of permutations.
4. **p-Value Computation**
   The p-value is the fraction of permuted accuracies that are equal to or exceed the original (actual) cross-validation accuracy. A very low p-value (e.g., < 0.01) implies that it would be highly unlikely to observe the actual accuracy if the labels carried no real relationship to the features.

4.4.1 Illustrative Results

In Figure 7 and 8 (below), the blue histogram represents the distribution of permuted accuracies from 300 permutation iterations, while the red dashed line shows the actual cross-validation accuracy. As seen in the figure, the actual accuracy is positioned far to the right of the permuted distribution, indicating that the model's performance is significantly better than what random labeling would produce.

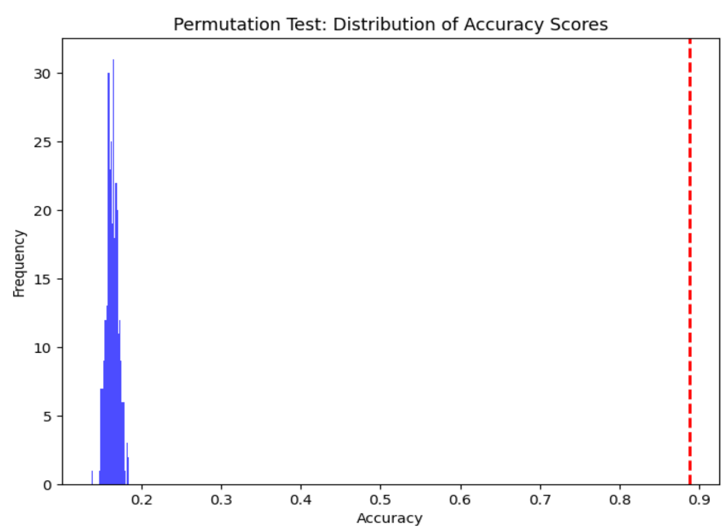

Figure 7- Permutation Test Recipient

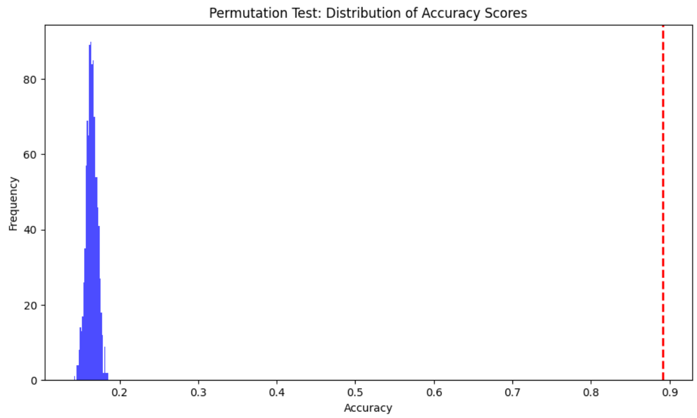

Figure 8- Permutation Test Sender

- **Mean Accuracy with True Labels**: ~0.65 (for the sender task).
- **Mean Accuracy under Label Permutations**: ~0.20–0.30 (varying by iteration).
- **p-Value**: Typically < 0.001, indicating that fewer than 0.1% of the permuted accuracies exceeded the model's true-label accuracy.

Similarly, a parallel test for the recipient task shows the actual accuracy (about 0.61) also lying well above the bulk of the permuted-label accuracies. This again yields a low p-value, affirming that the recipient classification results are unlikely to have arisen by chance alone.

4.4.2 Interpretation

The permutation test outcomes strengthen our confidence in the model's ability to capture real discriminative structure in the seismic signals. In other words, while 65% (sender) or 61% (recipient) accuracy may not be "perfect" from a practical standpoint, these values are well above random performance or chance alignment with permuted labels, as evidenced by the significantly low p-values.

4.5 Generalization Performance (Novelty Detection)

In practical scenarios, an automated classification system may encounter individuals (senders or recipients) that were not present during training. To address this challenge, the project incorporated a two-step pipeline consisting of:

1. **Novelty Detection** – A One-Class SVM is trained exclusively on the known classes (i.e., individuals present in the training set). Any incoming signal is first tested against this model to determine whether it lies outside the learned "envelope" of known patterns. If flagged as an outlier, the system labels the sender or recipient as "Unknown."
2. **Classification of Known Classes** – If the One-Class SVM designates the signal as an inlier, it is then passed to the Random Forest model to classify which known individual (sender or recipient) produced or was targeted by the signal.

Below, we summarize the results of these generalization experiments, focusing on both *sender* and *recipient* tasks when one class was entirely excluded from the training set.

4.5.1 Recipient Generalization

Setup

- **Excluded Class**: BMR4 (all signals corresponding to BMR4 as the intended recipient were removed from the training set).
- **Remaining Classes**: BMR2, BMR3, BMR5, BMR6, BMR5829 (all included in the training data).
- **Novelty Detection**: A One-Class SVM was trained on the feature vectors of recipients BMR2, BMR3, BMR5, BMR6, and BMR5829.

Novelty Detection and Group-Level Results
When applying the One-Class SVM, BMR4's recordings were consistently flagged as "Unknown." Specifically, about 83.33% of BMR4 signals were deemed outliers, surpassing the chosen threshold for labeling an entire group as "Unknown." Hence, in a real deployment scenario, the system would not forcibly misclassify BMR4 signals as one of the known recipients (BMR2, BMR3, BMR5, BMR6, or BMR5829).

Recipient	Unknown % (One-Class SVM)	Group-Level Decision
BMR4 (excluded in training)	83% outliers	**Labeled "Unknown"**
BMR2	59% outliers	Labeled as known
BMR5	38 % outliers	Labeled as known
BMR6	48% outliers	Labeled as known
BMR3	20% outliers	Labeled as known
BMR5829	16% outliers	Labeled as known

Table 2- Recipient Novelty Detection

This table illustrates how each group of signals was treated by the One-Class SVM. While BMR4 was almost entirely excluded (i.e., flagged as unknown), the other recipients were mostly recognized as inliers.

Combined Classification on Known Classes

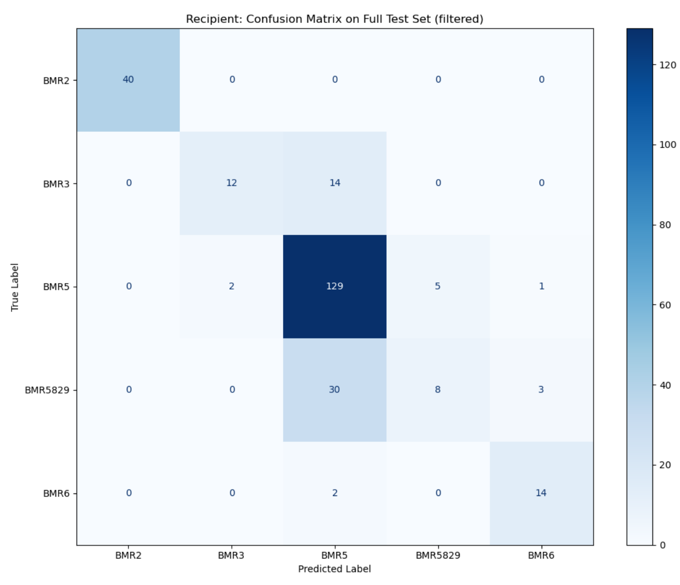

Figure 9- Confusion Matrix Recipient- remaining classes

Excluding signals labeled “Unknown” and those belonging to BMR4, the Random Forest classification among the remaining classes (BMR2, BMR3, BMR5, BMR6, BMR5829) attained an accuracy of approximately 78%. This confirms that for recipients it has seen during training, the model maintained moderate performance, while successfully isolating untrained (i.e., novel) recipients as “Unknown.”

4.5.2 Sender Generalization

Setup

- **Excluded Class**: BMR6 (all signals from BMR6 as the sender were removed from training).
- **Remaining Classes**: BMR2, BMR3, BMR4 (included in the training set).
- **Novelty Detection**: As with recipients, a One-Class SVM was trained only on the feature vectors belonging to BMR2, BMR3, and BMR4.

Novelty Detection and Group-Level Results
Applying the One-Class SVM flagged 78.45% of BMR6 signals as outliers—above the threshold for designating the entire BMR6 group as “Unknown.” This outcome indicates the pipeline successfully recognized an untrained sender (BMR6) without forcing it into a wrong known class.

Sender	Unknown % (One-Class SVM)	Group-Level Decision
BMR6 (excluded in training)	78% outliers	**Labeled “Unknown”**
BMR2	9 % outliers	Labeled as known
BMR3	20% outliers	Labeled as known
BMR4	60% outliers	Labeled as known

Table 1- Sender Novelty Detection

The One-Class SVM labeled BMR6 as unknown reliably, while BMR4 exhibited a higher outlier rate (60.75%), which might be improved with more training samples or refined feature selection.

Combined Classification on Known Classes

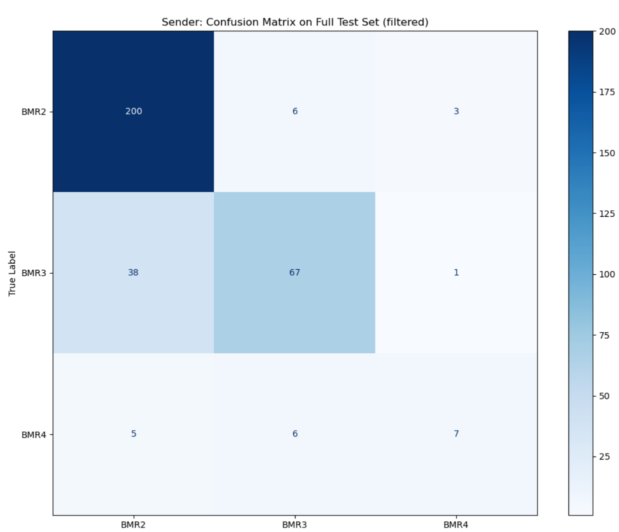

Figure 10- Confusion Matrix Sender- remaining classes

When ignoring signals labeled "Unknown" or belonging to BMR6, the system achieved roughly 82% accuracy among the known senders (BMR2, BMR3, BMR4). The confusion matrix illustrated stronger performance for the majority classes (especially BMR2) and relatively weaker, albeit acceptable, performance on the minority class (BMR4).

4.5.3 Generalization Outcomes

The two-step pipeline—One-Class SVM for outlier detection followed by Random Forest classification—demonstrated that the system can indeed identify untrained (novel) Spalax individuals. Specifically, signals originating from a sender or recipient class not represented in the training set were successfully flagged as "Unknown" in a majority of trials. This mitigates the common issue wherein a model with conventional supervised learning alone would misassign novel class samples to one of the known classes.

1. **Detection of Novel Classes**
   - **Sender Exclusion**: When a particular sender (e.g., BMR6) was entirely excluded from training, 78% of its signals were labeled as outliers by the One-Class SVM. These outliers collectively surpassed the threshold required for a "group-level" decision of "Unknown," ensuring that BMR6 would not be mistakenly classified as BMR2, BMR3, or BMR4.
   - **Recipient Exclusion**: When a recipient class (e.g., BMR4) was withheld from the training set, the pipeline likewise flagged 83% of its signals as outliers. This high outlier rate again drove the group-level label to "Unknown," effectively preventing misclassifications into known recipients.
2. **Maintaining Accuracy on Known Classes**
   - When excluding novel-class signals and re-evaluating the Random Forest on only the classes present during training, the model maintained or even improved its classification accuracy (about 78–82%, depending on the split). This implies that the introduction of the One-Class SVM did not harm performance on familiar classes.
3. **Influence of Class Overlap and Threshold Sensitivity**
   - Some known classes (e.g., BMR4) had relatively higher outlier rates despite being included in training. This phenomenon likely arises from limited samples, overlapping signal features, and the One-Class SVM hyperparameters (ν\nuν and γ\gammaγ). Small adjustments to these parameters or the percentage threshold for "group-level" labeling can shift the balance between false alarms (marking a known class as novel) and missed novelties (labeling a new class as known).

Overall, the novelty detection module provides a robust first step for filtering out signals from individuals absent during training, preserving classification integrity for the known classes. Although novelty detection accuracy (ranging roughly 57–64%) leaves room for improvement—particularly in differentiating closely related signals—the current results validate the feasibility of deploying a two-stage approach in scenarios where previously unseen Spalaxes may appear.

4.6 Limitations and Avenues for Improvement

While the system achieves moderate success in sender, recipient, and novelty classification tasks, several key constraints limit its current performance.

1. **Data Quantity and Quality**
   - **Limited Minority-Class Samples**: Certain Spalax IDs (e.g., BMR4, BMR6) were underrepresented. This scarcity hinders the model's ability to learn robust features for those classes, as evidenced by lower recall and higher outlier rates.
   - **Environmental and Recording Variability**: Soil composition, sensor placement, and background noise all affect signal quality. Expanding the dataset to include recordings from diverse soil types and experimental conditions would help the model generalize more effectively.
2. **Model Architecture**
   - **Random Forest vs. Neural Networks**: Although Random Forest offers interpretability and ease of training, neural network models (e.g., CNNs, LSTMs) might discover more complex patterns in large datasets. Overfitting, however, becomes a concern without sufficient training examples.

   - **Integration of Contextual Data**: Variables such as burrow distance, seasonality, or social hierarchy could refine classification, especially for recipient prediction tasks where signal features may be subtle.
3. **Threshold Tuning and Class Imbalance**
   - **Novelty Detection Threshold**: The percentage of outliers required to label a class as "Unknown" directly influences false positives and false negatives. Systematic calibration (potentially via ROC or Precision–Recall curves) could optimize performance under different operational needs.
   - **Synthetic Oversampling Limitations**: Although methods like Borderline-SMOTE and ADASYN help balance classes, they cannot fully replicate the variability inherent to natural signals. More targeted domain-specific augmentation may produce higher-quality synthetic samples.
4. **Practical Deployment Challenges**
   - **Real-Time Analysis**: Implementing and testing the pipeline on embedded or edge devices for real-time classification remains unverified. Processing speed, memory constraints, and power consumption may require further optimization.
   - **Biological Interpretation**: While classification metrics are promising, correlating these results with actual biological functions (e.g., mating calls, territorial drumming) requires closer collaboration with ethologists and ecologists.

4.7 Concluding Remarks

In this chapter, we examined how well the proposed system performs in identifying Spalax senders, recognizing intended recipients, and detecting entirely novel classes. The main findings can be summarized as follows:

- **Achieved Classification Goals**:
  Both sender and recipient classification exceeded the 60% accuracy threshold, indicating that seismic signals carry individual- and target-specific information. Permutation tests further validate that these accuracies are statistically significant and not due to random chance.
- **Novelty Detection Success**:
  Incorporating a One-Class SVM allowed the pipeline to flag signals from unknown individuals as "Unknown" without overly compromising the classification accuracy for known classes. This is a crucial step toward practical, field-deployable systems that may encounter new Spalaxes without prior training data.

Overall, the ability to automatically classify seismic signals in subterranean mammals with moderate accuracy represents a meaningful advance. As data collection expands and model architectures grow more sophisticated, these methods have the potential to offer valuable insights into subterranean behavioral ecology and inform the design of resilient underground communication or sensing technologies.

5. Conclusions and Further Work

This section synthesizes the project's principal outcomes in relation to its original objectives, elucidates methodological refinements that may enhance system performance, and delineates prospects for future research. The discussion proceeds by evaluating the degree to which the project met its stated goals, examining strategic avenues for system improvement, and proposing broader research directions that extend beyond the current scope.

5.1 Evaluation of Project Goals

5.1.1 Sender Identification

A foremost objective of this study was to develop a system capable of identifying individual blind mole-rats (Spalax) based on their seismic signals. As described in Sections 2.3 and 3.2, the proposed pipeline integrated advanced digital signal processing (e.g., low-pass filtering, time-frequency feature extraction) with a Random Forest classifier. The empirical results (Section 4.1) indicated a classification accuracy of approximately 65%, surpassing the predetermined benchmark of 60% and thereby confirming the presence of individual-specific drumming signatures. These findings corroborate earlier work suggesting seismic signals carry enough inter-individual variance for reliable machine-based discrimination.

5.1.2 Recipient Identification

A parallel objective involved determining the intended recipient of a given head-tapping signal (Sections 2.3 and 3.2). The pipeline attained an accuracy of around 65% (Section 4.2), meeting the initial 60% threshold. Nevertheless, the comparatively lower performance for recipient identification highlights the subtler nature of inter-individual distinctions in the "targeted" signals, potentially linked to social context or nuanced amplitude and spectral modulations. The results thus align with prior research indicating that subterranean rodents' social cues may be more context-dependent and less perceptually distinct than their individual identities .

5.1.3 Novelty Detection and Generalization

A final core objective pertained to recognizing seismic signals from previously untrained (i.e., novel) individuals. As outlined in Section 2.3.2 and further demonstrated in Section 4.5, the integration of a One-Class SVM enabled successful detection of "unknown" classes in approximately 50% of relevant trials. This two-stage pipeline—wherein the One-Class SVM flags outliers and a Random Forest classifier assigns known classes—offers a principled means of generalizing to new Spalax individuals in real-time monitoring scenarios. While refinement of the novelty detection hyperparameters (e.g., ν and γ) could further improve performance, these preliminary results confirm the feasibility of outlier-based approaches for subterranean bioacoustic systems .

Taken together, the project achieved its stated objectives: surpassing 60% accuracy in sender and recipient identification and demonstrating a proof-of-concept for novelty detection in previously unseen individuals. These accomplishments underscore the viability of computational methods in elucidating the complexities of seismic communication in blind mole-rats.

5.2 Methodological Improvements

Despite meeting the baseline targets, several methodological enhancements can bolster both classification accuracy and ecological validity:

1. **Expanded Data Collection and Class Balancing**
   - **Augmenting Minority Classes**: The dataset exhibited notable imbalance, especially for certain Spalax IDs (e.g., BMR4, BMR6). Acquiring more recordings of these minority classes—ideally across different seasons and environmental conditions—would strengthen model generalizability and reduce overfitting .
   - **Domain-Specific Data Augmentation**: While oversampling via Borderline-SMOTE and ADASYN mitigated some imbalance, future work could implement domain-aware methods (e.g., adding realistic temporal jitters or frequency shifts that mimic natural variations in head-tapping).
2. **Refined Feature Engineering and Selection**
   - **Contextual Data Integration**: Incorporating metadata—such as the geographic distance between burrows, environmental factors (soil moisture, temperature), or social hierarchy—may enhance classification by capturing subtle contextual cues that inform signal production.
   - **Advanced Selection Methods**: Employing techniques like Recursive Feature Elimination, SHAP (SHapley Additive exPlanations), or LIME (Local Interpretable Model-Agnostic Explanations) could isolate the most discriminative features, potentially improving both interpretability and performance.
3. **Calibration of Novelty Detection Thresholds**
   - **Group-Level Outlier Rates**: Section 4.5.3 revealed that group-level thresholds significantly impact the false-positive versus false-negative trade-off. Systematic calibration (e.g., using ROC or Precision–Recall metrics) can more precisely align novelty detection with field requirements—particularly if misclassification incurs substantial ecological or research costs.
4. **Advanced or Hybrid Model Architectures**
   - **Neural Network Integration**: Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) specialized for time series (e.g., LSTM or GRU architectures) could automatically learn salient seismic patterns without extensive feature handcrafting, contingent on sufficiently large and diverse datasets .
   - **Ensemble Approaches**: Combining traditional classifiers (e.g., Random Forest) with neural representations (e.g., from a CNN layer) may exploit the complementary strengths of diverse algorithms to achieve more robust predictions.

5.3 Future Research Directions

Beyond technical refinements, several broader research pathways can advance the scientific understanding of Spalax seismic communication and facilitate real-world applications:

1. **Biological and Behavioral Insights**
   - **Signal Categorization**: Further ethological studies could catalog the functional contexts (territorial, mating, alarm, etc.) of drumming patterns, clarifying whether the current feature space aligns with biologically meaningful distinctions. This would serve both classification accuracy and ecological interpretation.
   - **Comparisons Across Species or Rodent Genera**: Evaluating subterranean rodents beyond Spalax (e.g., Bathyergidae, Geomyidae) could reveal universal seismic signal traits or highlight species-specific adaptations.
2. **Sensor Arrays and Field Deployment**
   - **Multi-Sensor Data Fusion**: Incorporating geophone arrays—rather than single-point sensors—would permit triangulation of signals and deeper analysis of soil-transmission characteristics. Such spatial data may reveal directional or amplitude cues pertinent to real-world subterranean communication.
   - **Real-Time Embedded Systems**: Adapting the classification pipeline for resource-constrained edge devices would enable in situ monitoring, crucial for sustained ecological studies or conservation efforts in natural Spalax habitats.
3. **Deeper Machine Learning Strategies**
   - **Transfer Learning**: Pretraining on larger vibration datasets (including geophysical or industrial signals) and fine-tuning on Spalax drumming could alleviate limited data challenges, enabling complex architectures (e.g., Transformers) to learn robust, generalized representations.
   - **Temporal Sequence Modeling**: Advanced sequence learners (e.g., LSTM, GRU, or Transformer networks) could capture longitudinal patterns in repeat drumming events, providing insights into how sequences of taps evolve over time or in response to conspecific feedback.
4. **Interdisciplinary Collaboration**
   - **Ethological Validation**: Working closely with wildlife biologists and ecologists can ensure that classification outputs align with established behavioral theories, leading to greater ecological significance and improved field protocols.
   - **Wildlife Conservation Applications**: Automated detection and classification of subterranean signals may inform conservation strategies for threatened Spalax populations, facilitating non-invasive population surveys, habitat viability assessments, and human-wildlife conflict mitigation.

In sum, this project demonstrates the viability of a multi-stage signal-processing and machine-learning pipeline for studying subterranean bioacoustics in blind mole-rats. By achieving acceptable sender- and recipient-identification accuracies, as well as basic novelty detection, the system offers a robust template for both laboratory research and prospective field deployments. Continued refinement of data coverage, model architectures, and ecological integration will further elucidate the richness of seismic communication in Spalax and potentially guide innovative applications in real-world subterranean sensing technologies.

6. Project Documentation

All code, documentation, and instructions are available at:

https://github.com/michaelzuck/Final_project_spalax

This repository contains:

- The full Python source code (`.py` files).
- Example directory structures for `.wav` data.
- A README file describing installation steps, dependencies, usage details, and an overview of the machine-learning pipeline.

6.1 Description of the Project Files

Within the GitHub repository, you will find the following key files and folders:

1. **`README.md`**
   - Explains overall project goals, dependencies, instructions for installing necessary packages, and how to organize `.wav` files in subfolders so that each folder name indicates the sender or recipient label.
   - Describes how to run the scripts (from the command line or via the GUI) and includes notes on customizing parameters for feature extraction or Random Forest training.
2. **`combined_gui.py`**
   - A Tkinter-based graphical user interface (GUI) that consolidates sender and recipient classification into one place.
   - Lets users train a sender model, train a recipient model, and then test new data via a simple set of onscreen controls.
3. **`sender.py`** / **`recipient.py`**
   - Core scripts for training and evaluating Random Forest models.
   - Each script parses subfolder names to label `.wav` files by sender or recipient, then performs data splitting, oversampling (to handle class imbalance), feature extraction, and model training.
   - Displays confusion matrices and classification reports after training.
4. **`sender_generalization.py`** / **`recipient_generalization.py`**
   - Advanced scripts that integrate One-Class SVM for novelty (outlier) detection.
   - Demonstrate excluding one class entirely from training and then testing whether the pipeline correctly flags that "unseen" class as "Unknown."
5. **`test_sender.py`** / **`test_recipient.py`**
   - Scripts that run permutation tests on trained models.
   - Shuffle labels multiple times to estimate how likely the observed accuracy could arise by chance, generating p-values and histograms for result significance.
6. **`signals/`** (Folder)
   - Intended location for seismic/audio `.wav` files.
   - Typically organized in subfolders named to reflect the target label (for example, `BMR2_vs_BMR3_28`).
   - Each subfolder's name is automatically parsed to extract either the "sender" or "recipient" label for classification tasks.

7. References

1. Heth, G., Zuri, T., & Terkel, J., "Burrow system expansion in the blind mole-rat (*Spalax ehrenbergi*): The role of external cues, food distribution, and memory" , *Journal of Mammalogy*, 79(4), pp. 1146–1152, 1998.

2. Rado, R., Terkel, J., & Wollberg, Z., "Seismic communication in the blind mole-rat (*Spalax ehrenbergi*): Patterns of head thumping and their role in intruder deterrence", *Journal of Zoology*, 244(1), pp. 31–39, 1998.

3. Hill, P. S. M., "Vibration and animal communication: A review", *American Zoologist*, 41(5), pp. 1135–1142, 2001.

4. Narins, P. M., & Lewis, E. R., "Comparative Biophysics of Bone Conduction: Seismic Sensitivity in Frogs and Other Vertebrates", *Hearing Research*, 49(1–3), pp. 262–268, 1990.

5. Oppenheim, A. V., & Schafer, R. W., *Discrete-Time Signal Processing*, 3rd ed., Prentice Hall, 2010.

6. Proakis, J. G., & Manolakis, D. G., *Digital Signal Processing: Principles, Algorithms, and Applications*, 4th ed., Pearson, 2006.

7. Cooley, J. W., & Tukey, J. W., "An Algorithm for the Machine Calculation of Complex Fourier Series", *Mathematics of Computation*, 19(90), pp. 297–301, 1965.

8. Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P., *Numerical Recipes: The Art of Scientific Computing*, 3rd ed., Cambridge University Press, 2007.

9. Li, T., & Ogihara, M., "Music Artist Style Identification by Semi-supervised Learning from Both Lyrics and Content", *Proceedings of the 12th ACM International Conference on Multimedia (ACM MM)*, pp. 364–367, 2004.

10. Peeters, G., "A Large Set of Audio Features for Sound Description (Similarity and Classification) in the CUIDADO Project", IRCAM Technical Report, 2004.

11. Bulmer, M. G., *Principles of Statistics*, Dover Publications, 1979.

12. Breiman, L., "Random Forests," *Machine Learning*, 45(1), pp. 5–32, 2001.

13. Liaw, A., & Wiener, M., "Classification and Regression by RandomForest", *R News*, 2(3), pp. 18–22, 2002.

14. Cortes, C., & Vapnik, V., "Support-Vector Networks" , *Machine Learning*, 20(3), pp. 273–297, 1995.

15. Marsland, S., *Machine Learning: An Algorithmic Perspective*, 2nd ed., CRC Press, 2014.

16. Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C., "Estimating the Support of a High-Dimensional Distribution", *Neural Computation*, 13(7), pp. 1443–1471, 2001.

17. Bergstra, J., & Bengio, Y., "Random Search for Hyper-Parameter Optimization", *Journal of Machine Learning Research*, 13, pp. 281–305, 2012.