# Advanced Analysis and Model Development on Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)

H3Art
*International School*
*Jinan University*
Guangzhou, China

***Abstract*—Emotion recognition is a crucial aspect of artificial intelligence and machine learning, with significant implications for creating more intuitive and responsive systems. This report focuses on advanced data mining techniques applied to the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS). The study explores various tasks, including anomaly detection, imbalanced learning, classification, and clustering, using the RAVDESS dataset. Key findings highlight the effectiveness of SMOTE for handling imbalanced data, with neural networks and ensemble models, particularly Light Gradient Boosting Machine (GB), demonstrating strong performance in classification tasks. Clustering analysis using Dynamic Time Warping (DTW) provided more coherent groupings of time series data. The results underscore the importance of appropriate preprocessing, balancing techniques, and classification algorithms to improve emotion recognition systems. Future work could further optimize deep learning models and integrate additional modalities to enhance emotion recognition accuracy.**

***Index Terms*—Audio Analysis, Emotion Recognition, Imbalance Learning, Data Mining**

## I. Introduction

Emotion recognition is a crucial aspect of artificial intelligence and machine learning, with significant implications for creating more intuitive and responsive systems. In this report, I focus on advanced data mining techniques applied to the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS). The primary objective of this study is to explore various data mining tasks, including anomaly detection, imbalanced learning, classification, and clustering, using the RAVDESS dataset. The RAVDESS dataset, developed by Livingstone & Russo (2018), is a validated multimodal dataset comprising audio-visual recordings of 24 professional actors vocalizing two lexically-matched statements in a neutral North American accent [1]. This dataset is significant for applications across various domains, including human-computer interaction, psychological studies, and entertainment industries. Existing work in this field has employed various techniques such as support vector machines (SVM), neural networks (NN), and ensemble models to classify emotions. This study aims to enhance these methods' accuracy and robustness, particularly for imbalanced datasets. The report will provide a comprehensive analysis of the RAVDESS dataset, evaluate anomaly detection methods, investigate imbalanced learning techniques, compare classification algorithms, apply clustering techniques, and summarize findings with insights into future trends in emotion recognition systems.

### A. Data Semantics

Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) is a validated multimodal dataset developed by Livingston & Russo (2018) which consists of audio-visual recordings of 24 professional actors vocalizing two lexically-matched statements in a neutral North American accent. The data employed in this study is a modified version of the original RAVDESS where alongside the original categorical attributes (Table I) numerical attributes are created by extracting quantitative statistics from the raw audio signals (Table II).

TABLE I: Categorical Attributes

| Name | Type | Description |
|---|---|---|
| modality | Nominal | Recording mode |
| vocal_channel | Nominal | Type of vocal communication |
| emotion | Nominal | Emotion expressed |
| emotional_intensity | Ordinal | Degree of emotional involvement |
| statement | Nominal | Statement uttered |
| repetition | Ordinal | Repetition of the statement |
| actor | Nominal | Actor's ID |
| sex | Nominal | Actor's sex |
| filename | Nominal | Record's ID |

Further attributes have been created by dividing each time series into 4 non overlapping windows and computing all the quantitative statistics described in Table II at a local level. The names referring to such features can be easily derived from the expression:

$$\text{NAME\_w}N$$

and by replacing NAME with the name of the numerical attribute and $N$ with the index (1, 2, 3 or 4) of the window considered. The only exception is represented by the global-level feature frame_count, which is replaced locally with the denomination length_w$N$.

## II. Methods

### A. Data Cleaning and Pre-processing

The RAVDESS dataset used in this study includes audio-visual recordings of 24 actors, capturing various emotional

TABLE II: Global-level Numerical Attributes

| Name(s) | Type | Description |
|---|---|---|
| frame_count | Interval | Number of frames per sample |
| mean, std, min, max, skew, kur, q_01, q_05, q_25, q_50, q_75, q_95, q_99 | Ratio | Statistics of original audio signal |
| lag1_sum, lag1_mean, lag1_std, lag1_min, lag1_max, lag1_kur, lag1_skew, lag1_q01, lag1_q05, lag1_q25, lag1_q50, lag1_q75, lag1_q95, lag1_q99 | Ratio | Statistics of Lag (difference between each observation and the antecedent) |
| zc_sum, zc_mean, zc_std, zc_min, zc_max, zc_kur, zc_skew, zc_q01, zc_q05, zc_q25, zc_q50, zc_q75, zc_q95, zc_q99 | Ratio | Statistics of Zero Crossing Rate |
| mfcc_sum, mfcc_mean, mfcc_std, mfcc_min, mfcc_max, mfcc_q01, mfcc_q05, mfcc_q25, lag1_q50, mfcc_q75, mfcc_q95, mfcc_q99, mfcc_kur | Ratio | Statistics of Mel-Frequency Cepstral Coefficients |
| sc_sum, sc_mean, sc_std, sc_min, sc_max, sc_kur, sc_skew, c_q01, sc_q05, sc_q25, sc_q50, sc_q75, sc_q95, sc_q99 | Ratio | Statistics of Spectral Centroid |
| stft_sum, stft_mean, stft_std, stft_min, stft_max, stft_kur, stft_skew, stft_q01, stft_q05, stft_q25, stft_q50, stft_q75, stft_q95, stft_q99 | Ratio | Statistics of Short-Time Fourier Transform |

expressions. The data is divided into training (TR) and test (TS) sets, with 34.1% reserved for testing. To begin, a global understanding of numerical attributes was achieved by analyzing their skewness and kurtosis distributions, as shown in Figure 1.

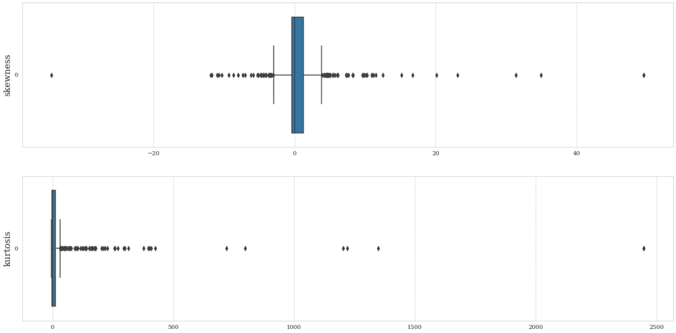


Fig. 1: Distributions of skewness and kurtosis of numerical attributes. Data is restricted to TR as it is assumed that TS data are drawn from the same probability distributions.

Data balancing for categorical attributes was assessed through visual inspection and computation of relative frequencies. Notable imbalances were observed in the vocal_channel and emotional_intensity attributes, with more significant imbalance in the emotion attribute. No inconsistencies, missing values, or duplicates were found, allowing the elimination of continuous attributes with null variance and categorical attributes with unique values, reducing the dimensionality from 434 to 383. Standardization, one-hot encoding, and feature reduction were applied as necessary.

Then I perform anomaly detection, various unsupervised algorithms were compared, including Histogram-Based Outlier Score (HBOS), Deviation-Based (DB), Elliptical Envelope (EE), K-Nearest Neighbors (KNN), Local Outlier Factor (LOF), DBSCAN, Angle-Based Outlier Degree (ABOD), Lightweight Online Detector of Anomalies (LODA), and Isolation Forest (IF). Pre-processing steps included standardization of numerical attributes, one-hot encoding of categorical attributes, and PCA for feature reduction where needed.

Each method was set with a contamination rate of 1%, targeting the top 19 data points with the highest outlier scores. Outliers identified by the methods were managed by replacing continuous attribute values with the median, a robust measure due to the asymmetry of numerical attributes. A final assessment ensured no overlap between new and previous outliers, as illustrated in Figure 2.

### *B. Imbalance Learning*

Imbalanced learning is a critical step in preparing the RAVDESS dataset for accurate classification tasks. This section aims to address the class imbalance problem through three techniques: random undersampling, Synthetic Minority Oversampling Technique (SMOTE), and class weight adjustment.

For this study, I focused on three binary classification tasks: vocal_channel, sex, and emotional_intensity. The provided test data (TS) was not used in this phase; instead, operations were conducted solely on the training data (TR). Pre-processing steps for each target included:

- **Standardization**: Numerical attributes were standardized using min-max scaling.
- **One-Hot Encoding**: Categorical attributes (excluding the target) were transformed into a binary format.
- **Label Encoding**: The target attribute was label encoded.

To create an imbalanced setting, random values were removed from the majority class of the binary targets to achieve a 96%-4% proportion. This artificially created imbalance maximizes the data quantity before applying re-balancing techniques. The distribution of the target attributes before imbalancing is illustrated in Figure 3.

For the K-Nearest Neighbors (K-NN) classifier, which is sensitive to feature selection, a filter strategy was employed to reduce training features. Non-parametric Spearman correlation coefficients ($\rho$) were calculated between the target variable and each continuous attribute. Attributes with correlation $\rho > 0.7$ or $\rho < -0.7$ for vocal_channel and sex, and $\rho > 0.4$ or $\rho < -0.4$ for emotional_intensity were retained.

Model selection for Decision Tree (DT) and K-NN classifiers was performed using a randomized search with repeated stratified 5-fold cross-validation. The tested hyper-parameters and their values for DT and K-NN are detailed in Tables III and IV, respectively.

To ensure the reliability of random undersampling, this process was iterated 10 times, and the mean F1-scores of DT and K-NN were considered as performance indicators.

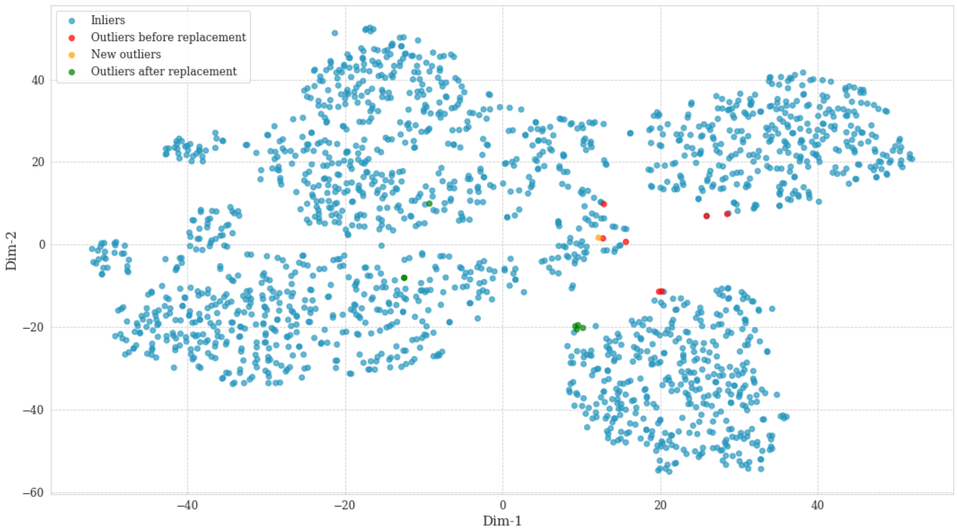
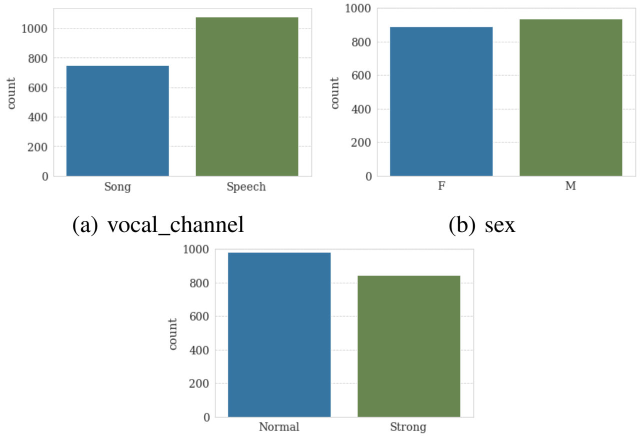


Fig. 2: Distributions of skewness and kurtosis of numerical attributes. Data is restricted to TR as it is assumed that TS data are drawn from the same probability distributions.



(a) vocal_channel

(b) sex

(c) emotional_intensity

Fig. 3: Data distributions before imbalancing.

TABLE III: Tested hyper-parameters for Decision Trees

| **Hyperparameter** | **Description** | **Tested Values** |
|---|---|---|
| Criterion | Metric to choose the best split | Gini, Entropy, Log-Loss |
| Max Depth | Maximum depth of the tree | Discrete interval [2, 200] |
| Min Samples Split | Minimum number of samples for split | Log-uniform distribution between [0.01, 1] |
| Min Samples Leaf | Minimum number of samples in leaf | Uniform distribution between [0.001, 0.2] |

TABLE IV: Tested hyper-parameters for K-NN

| **Hyperparameter** | **Description** | **Tested Values** |
|---|---|---|
| K | Number of neighbors | Discrete interval [2, $N/2$] |
| Weights | Weight function in prediction | Uniform, Distance |
| Metric | Distance metric | City-Block, Euclidean, Cosine, Chebyshev |

This precaution is not necessary for SMOTE or class weight adjustment.

### C. *Classification*

The classification tasks in this study aim to accurately categorize different attributes of the RAVDESS dataset, specifically vocal_channel, sex, emotional_intensity, and emotion (multi-class classification task). Various machine learning models were employed to evaluate their effectiveness in these tasks, including Logistic Regression (LG), Support Vector Machines (SVM), Neural Networks (NN), Decision Tree Bagging (DTB), Random Forests (RF), AdaBoost (AB), and Gradient Boosting (GB). LG has also been regarded as a baseline model for evaluation. The classification performance of these models was evaluated on the test data (TS), providing insights into their relative strengths and weaknesses in emotion recognition. The pre-processing steps involved standardizing numerical attributes, one-hot encoding categorical attributes (excluding the target), and label encoding the target attribute.

*1) Logistic Regression:* Logistic Regression (LG) is a fundamental classification algorithm widely used for binary classification tasks. It models the probability that a given input belongs to a particular class, using a logistic function to ensure the output lies between 0 and 1. In this study, LG was used as a baseline model for comparison with more complex classifiers.Model selection for Logistic Regression was performed using a grid search with 5-fold cross-validation. The hyper-parameters and their corresponding values tested are shown in Table V. Since some penalties are incompatible with certain solvers, L2 regularization was set for all candidates, and the maximum number of iterations required for the solver to converge was set to 800.

*2) Support Vector Machines:* Support Vector Machines (SVM) are powerful classification algorithms that aim to find the optimal hyperplane that separates different classes

TABLE V: Logistic Regression Tested Hyperparameters

| **Hyperparameter** | **Description** | **Tested Values** |
|---|---|---|
| C | Inverse of regularization strength | Log-uniform distribution between [$10^{-4}$, $10^{3}$] |
| Solver | Algorithm to use in the optimization problem | L-BFGS, LIBLINEAR, Newton-CG, Newton-Cholesky, SAG, SAGA |

in the feature space. SVMs can handle both linear and non-linear classification tasks by using various kernel functions to transform the input space. In this study, both linear and non-linear SVM classifiers were assessed.Initially, linear SVM classifiers were evaluated. Model selection for linear SVM was performed using a randomized search with 5-fold cross-validation, focusing on the C hyper-parameter, which controls the trade-off between achieving a low training error and a low testing error.

TABLE VI: Linear SVM Tested Hyperparameters

| **Hyperparameter** | **Description** | **Tested Values** |
|---|---|---|
| C | Regularization parameter | Log-uniform distribution between [$10^{-4}$, $10^{4}$] |

Since the linear SVM classifiers did not achieve competitive results compared to the baseline (Logistic Regression), the evaluation was extended to non-linear SVM classifiers. This involved a similar randomized search procedure over a broader hyper-parameter space, including different kernel functions (linear, polynomial, and RBF), the penalty parameter C, and the kernel coefficient $\gamma$. A finer grid search was subsequently conducted around the best values identified in the initial search.

TABLE VII: SVM Tested Hyperparameters

| **Hyperparameter** | **Description** | **Tested Values** |
|---|---|---|
| C | Regularization parameter | Log-uniform distribution between [$10^{-4}$, $10^{4}$] |
| $\gamma$ | Kernel coefficient | Log-uniform distribution between [$10^{-4}$, $10^{4}$] |
| Kernel | Kernel function | Linear, Polynomial, RBF |

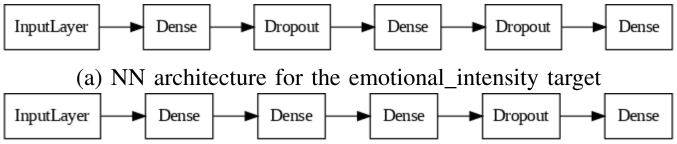
*3) Neural Networks:* Neural Networks (NN) are a class of machine learning models inspired by the human brain, capable of modeling complex relationships in data through layers of interconnected nodes. For this study, I initially considered shallow feedforward neural networks (FFNN) with a single hidden layer and logistic activation functions. The model weights were initialized using the Glorot normal distribution, and optimization was performed using mini-batch gradient descent with L2 regularization.For binary classification tasks (vocal_channel, sex, and emotional_intensity), logistic activation was used in the output layer with binary cross-entropy as the loss function. For the multi-label classification task (emotion), softmax activation was used in the output layer with sparse categorical cross-entropy as the loss function.Model selection involved a randomized search with 3-fold cross-validation over a broad hyper-parameter space, including the number of units in the hidden layer, learning rate, momentum coefficient, L2 regularization coefficient, and number of epochs.

A fraction of 20% of the training data was separated as validation data to visualize learning curves and control convergence time. The tested hyper-parameters are listed in Table VIII.

TABLE VIII: Neural Network (I) Tested Hyperparameters

| **Hyperparameter** | **Description** | **Tested Values** |
|---|---|---|
| Size | Number of units in hidden layers | Powers of 2 within [2, 28] |
| Epochs | Number of epochs | 10, 20, 50, 100, 200 |
| $\eta$ | Learning rate | Log-uniform distribution between [$10^{-3}$, 1] |
| $\alpha$ | Momentum coefficient | Log-uniform distribution between [$10^{-3}$, 1] |
| $\lambda$ | L2 regularization coefficient | Log-uniform distribution between [$10^{-3}$, 1] |

For the emotional_intensity and emotion targets, the simple FFNN did not outperform the baseline (Logistic Regression). Thus, deeper architectures with multiple hidden layers, optimized with the Adam optimizer and regularized with dropout, were tested. These deeper networks used the same logistic activation for hidden layers but incorporated additional regularization strategies to prevent overfitting. The architectures of these deeper networks are depicted in Figure 4, and the tested hyper-parameters are listed in Table IX.



(a) NN architecture for the emotional_intensity target

(b) NN architecture for the emotion target

Fig. 4: Architectures of the deeper NNs tested for the targets

TABLE IX: Neural Network (II) Tested Hyperparameters

| **Hyperparameter** | **Description** | **Tested Values** |
|---|---|---|
| Size | Number of units in hidden layers | Powers of 2 within [2, 28] |
| Epochs | Number of epochs | 10, 20, 50, 100, 200 |
| $\eta$ | Learning rate | Log-uniform distribution between [$10^{-3}$, 1] |
| $p$ | Dropout rate | 0.2, 0.4, 0.6 |

This thorough approach ensures the selection of the most suitable neural network configuration for each classification task, balancing model complexity with generalization performance.

*4) Ensemble Models:* Ensemble models combine multiple base classifiers to improve overall performance, leveraging the strengths of different algorithms and reducing the risk of overfitting. In this study, I evaluated the performance of four ensemble methods: Decision Tree Bagging (DTB), Random Forest (RF), AdaBoost (AB), and Light Gradient Boosting Machine (GB).

- **Decision Tree Bagging (DTB)**: Bagging, or Bootstrap Aggregating, involves training multiple decision trees on different subsets of the training data and combining their predictions. Model selection for DTB was performed using a randomized search with 3-fold cross-validation. The hyper-parameters tested included the maximum number of samples and features used for training the base estimators. The base estimator hyper-parameters were the same as those for decision trees, as detailed in Table III.

TABLE X: Decision Tree Bagging Tested Hyperparameters

| Hyperparameter | Description | Tested Values |
|---|---|---|
| Max Samples | Maximum number of samples to train each base estimator | 0.5, 0.6, 0.7, 0.8 |
| Max Features | Maximum number of features to train each base estimator | Discrete interval [2, $N$] |

- **Random Forest (RF)**: Random Forests extend the bagging approach by introducing feature randomization. Each tree in the forest is trained on a random subset of features, enhancing model diversity. The hyper-parameters for RF included the maximum number of features for selecting the best split. The hyper-parameter space for the base estimators was the same as that for decision trees.

TABLE XI: Random Forest Tested Hyperparameters

| Hyperparameter | Description | Tested Values |
|---|---|---|
| Max Features | Maximum number of features to choose best split | $\sqrt{N}$, $\log_2(N)$, $N$ |

- **AdaBoost (AB)**: AdaBoost (Adaptive Boosting) sequentially trains classifiers, each focusing on the errors made by the previous ones. The final model is a weighted sum of the individual classifiers. Model selection for AB involved tuning the learning rate, which controls the contribution of each classifier at each boosting iteration. The base estimator was assumed to be a decision stump.

TABLE XII: AdaBoost Tested Hyperparameters

| Hyperparameter | Description | Tested Values |
|---|---|---|
| Learning Rate | Weight applied to each classifier at each boosting iteration | Log-uniform distribution between [$10^{-4}$, 1] |

- **Light Gradient Boosting Machine (GB)**: Gradient Boosting builds an additive model by sequentially fitting a base learner to the residuals of the combined model. LightGBM is an efficient implementation of gradient boosting. Model selection for LightGBM involved tuning the boosting type, the number of boosted trees, the learning rate, and the maximum number of leaves for base learners. Additionally, categorical features were automatically transformed for LightGBM, but one-hot encoding yielded better performance.

TABLE XIII: Light Gradient Boosting Tested Hyperparameters

| Hyperparameter | Description | Tested Values |
|---|---|---|
| Boosting type | Gradient boosting algorithm | GBDT, GOSS, DART |
| Estimators | Number of boosting trees | Uniform distribution between [50, 500] |
| $\eta$ | Learning rate | Log-uniform distribution between [$10^{-4}$, 1] |
| Leaves | Maximum number of leaves per base learner | Uniform distribution between [5, 50] |

Each ensemble method was evaluated based on its ability to handle the complexities of the classification tasks in the RAVDESS dataset. This comprehensive approach ensures the identification of the most effective ensemble techniques, leveraging their combined predictive power for improved accuracy and robustness in emotion recognition.

### *D. Clustering*

For the clustering task, I aimed to group similar patterns in the time series data extracted from the RAVDESS dataset. Clustering was performed using two primary algorithms: K-Means and Hierarchical Agglomerative Clustering.

The raw audio signals from the RAVDESS dataset were sampled at a rate of 8 kHz, resulting in time series with 50,718 timestamps. Due to a large number of missing values in the terminal timestamps, the time series were reduced by calculating the average length without missing values and removing all but the first average length values from each series. Remaining missing values were replaced with the average value of the corresponding time series. Noise smoothing was applied using a moving average with a window size of 3.

*1) K-Means Clustering:* For K-Means clustering, time series were approximated using Symbolic Aggregate Approximation (SAX) with 600 segments and 14 symbols to improve computational efficiency. The optimal number of clusters (k) was determined using the ”elbow method,” which minimizes the Sum of Squared Error (SSE). I tested four versions of K-Means with different distance metrics:

- Euclidean distance
- Dynamic Time Warping (DTW)
- DTW with Sakoe-Chiba band constraint
- DTW with Itakura parallelogram constraint

*2) Hierarchical Agglomerative Clustering:* Hierarchical Agglomerative Clustering was also applied to the time series data, using both Euclidean distance and DTW in combination with various proximity measures: Single Link, Complete Link, Group Average, and Ward’s method (the latter only for Euclidean distance). The evaluation of potential clusterings was conducted through visual inspection of dendrograms to identify optimal cutting points for cluster formation.

*3) Dimensionality Reduction:* Dimensionality reduction techniques such as t-SNE and PCA were used to visualize the clusters obtained from both K-Means and Hierarchical Agglomerative Clustering.

This clustering approach provides a comprehensive method for identifying and grouping similar patterns within the time series data of the RAVDESS dataset, facilitating deeper insights into the underlying structure of the data.

## III. Results

### A. *Imbalance Learning Results*

To address the class imbalance in the RAVDESS dataset, three techniques were evaluated: random undersampling, Synthetic Minority Oversampling Technique (SMOTE), and class weight adjustment. The performance of these techniques was measured using F1-scores for two simple classifiers: Decision Tree (DT) and K-Nearest Neighbors (K-NN), before and after re-balancing the data.

*1) Decision Trees:* For Decision Trees, the balancing methods had a negligible effect on the classification of vocal_channel and sex but played a significant role in the classification of emotional_intensity. The balancing methods improved the model's ability to detect the minority class (normal), although this improvement came at the cost of a decrease in the F1-score for the majority class (strong). The results are visualized in Figure 5.

*2) K-Nearest Neighbors:* For K-NN, the F1-scores for the minority classes (speech, M, normal) in the imbalanced learning scenarios were null. Both SMOTE and random undersampling significantly improved the model's ability to recognize the minority class. SMOTE outperformed random undersampling, likely due to the relatively small size of the data, and also performed better than class weight adjustment, which did not improve the Decision Tree's ability to recognize minority class instances of emotional_intensity. The results for K-NN are shown in Figure 6.

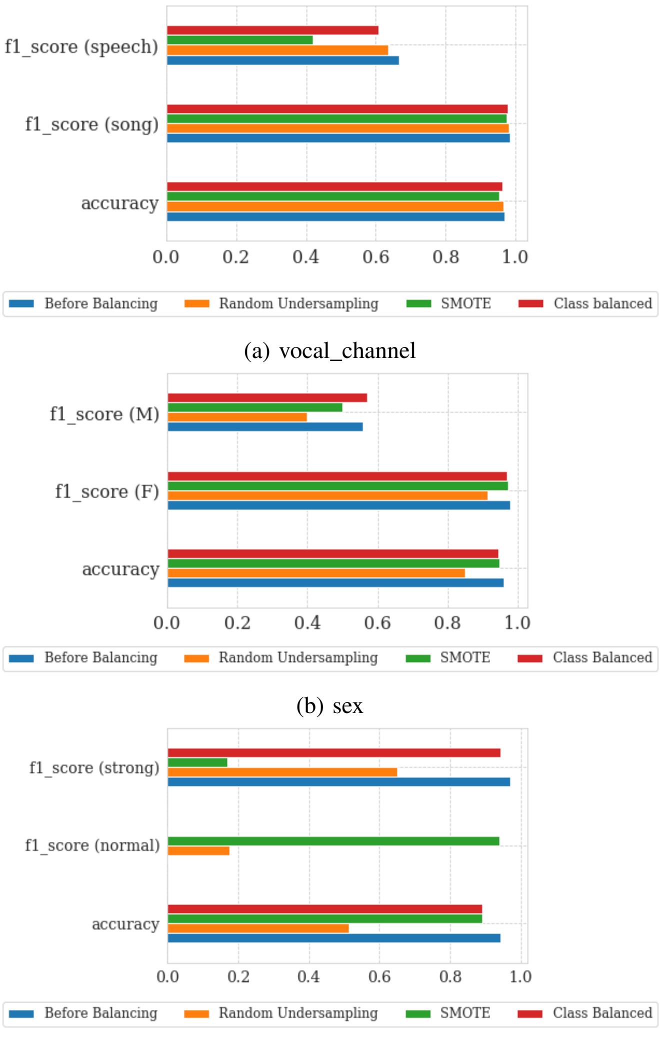
Overall, SMOTE proved to be the most effective technique for handling imbalanced data in this study, improving the classifiers' performance in recognizing minority classes across the different target attributes. These findings highlight the importance of selecting appropriate re-balancing techniques to enhance model performance in imbalanced learning scenarios.

### B. *Classification Results*

The classification tasks aimed to evaluate the performance of various machine learning models on the RAVDESS dataset. Three binary classification tasks (vocal_channel, sex, emotional_intensity) and one multi-class classification task (emotion) were addressed using different models, including Logistic Regression (LG), Support Vector Machines (SVM), Neural Networks (NN), Decision Tree Bagging (DTB), Random Forests (RF), AdaBoost (AB), and Gradient Boosting (GB). The results for each model and task are summarized below.

*1) Logistic Regression:* Logistic Regression served as a baseline model. The best results for each target are summarized in Table XIV.

*2) Support Vector Machines:* The performance of linear and non-linear SVM classifiers was evaluated. The best results for non-linear SVMs are summarized in Table XV.



(a) vocal_channel

(b) sex

(c) emotional_intensity

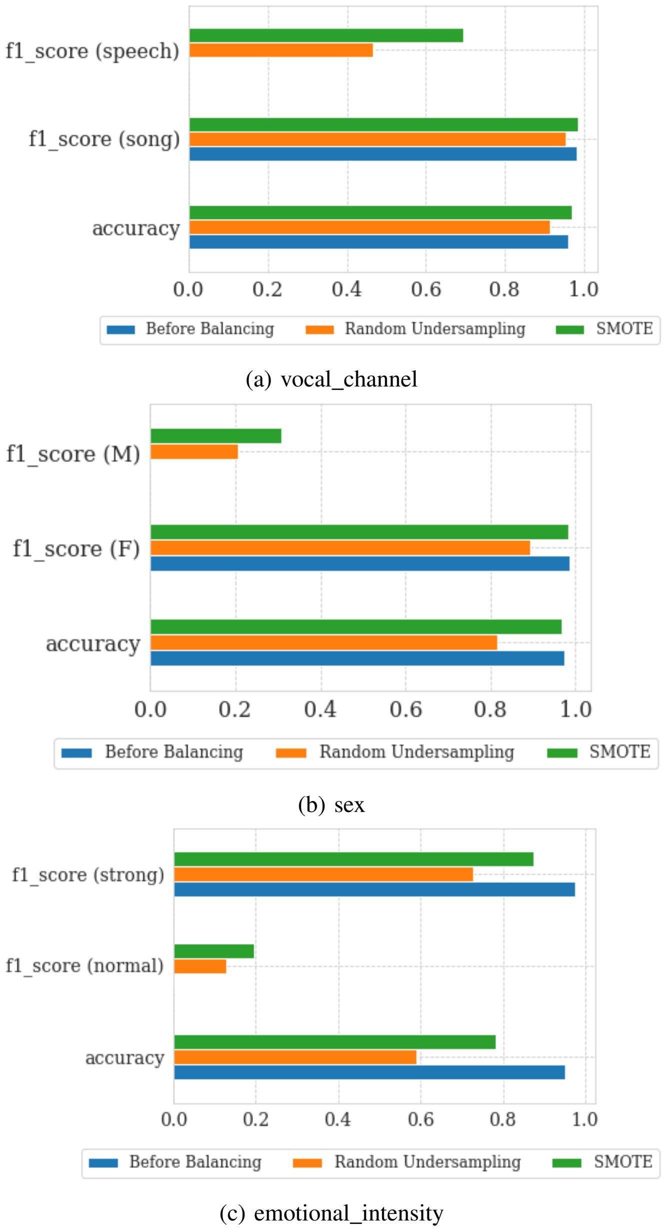
Fig. 5: Results of Decision Tree for each target before balancing and after the application of balancing methods.

TABLE XIV: Best Results for Logistic Regression

| Target | C | Solver | Weighted F1 | Accuracy |
|---|---|---|---|---|
| Vocal Channel | 1 | L-BFGS | 0.98 | 0.98 |
| Sex | 1 | L-BFGS | 0.85 | 0.85 |
| Emotional Intensity | 1 | L-BFGS | 0.77 | 0.77 |
| Emotion | 1 | Newton-Cholesky | 0.43 | 0.49 |

TABLE XV: Best Results for SVM

| Target | C | $\gamma$ | Kernel | Weighted F1 | Accuracy |
|---|---|---|---|---|---|
| Vocal Channel | 95.454 | 0.0006 | RBF | 0.98 | 0.98 |
| Sex | 0.7 | 0.1 | RBF | 0.90 | 0.90 |
| Emotional Intensity | 1072.26 | 0.0003 | RBF | 0.76 | 0.76 |
| Emotion | 0.018 | 0.097 | Polynomial | 0.47 | 0.50 |



(a) vocal_channel

(b) sex

(c) emotional_intensity

Fig. 6: Results of K-NN for each target before balancing and after the application of balancing methods.

*3) Neural Networks:* Shallow feedforward neural networks (FFNN) and deeper architectures were tested. The best results for each target using the optimal neural network configurations are summarized in Table XVI. To ensure robustness, each model was re-trained with 10 different random weight initializations, and the average and standard deviation of performance metrics were reported in Table XVIII.

The performance of the best neural network model was compared with human raters' accuracy in recognizing emotions from audio-only sources as reported by Livingstone & Russo (2018) [1]. The recall of human classification and neural classification tends to follow a similar trend, as shown in Figure 7.

TABLE XVI: Best Results for Neural Network

| Target | Size(s) | Epochs | $\eta$ | $\alpha$ | $\lambda$ | $p$ | Weighted F1 | Accuracy |
|---|---|---|---|---|---|---|---|---|
| Vocal Channel | 2 | 35 | 0.1 | 0.01 | 0 | - | 0.98 | 0.98 |
| Sex | 128 | 61 | 1 | 0.001 | 0 | - | 0.95 | 0.95 |
| Emotional Intensity | 16, 8 | 118 | 0.001 | - | - | 0.2 | 0.77 | 0.78 |
| Emotion | 256, 256, 256 | 290 | 0.0001 | - | - | 0.4 | 0.52 | 0.53 |

TABLE XVII: Mean and Standard Deviation of Performance Metrics

| Target | Weighted F1 | Accuracy |
|---|---|---|
| Vocal Channel | 0.96± 0.02 | 0.96± 0.02 |
| Sex | 0.95± 0.00 | 0.95± 0.00 |
| Emotional Intensity | 0.77± 0.01 | 0.77± 0.01 |
| Emotion | 0.51± 0.01 | 0.52± 0.01 |

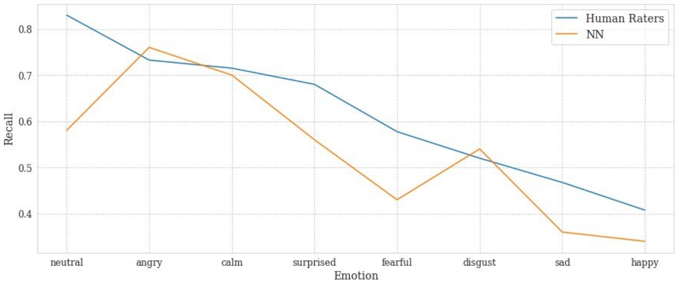


Fig. 7: Recall of Human Raters and NN in recognizing emotions.

The confusion matrix of the neural classification highlights similar mistakes made by humans in emotion recognition (Figure 8).

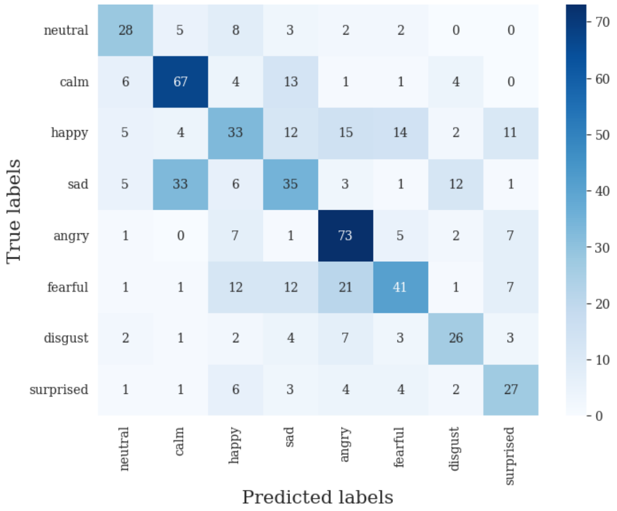
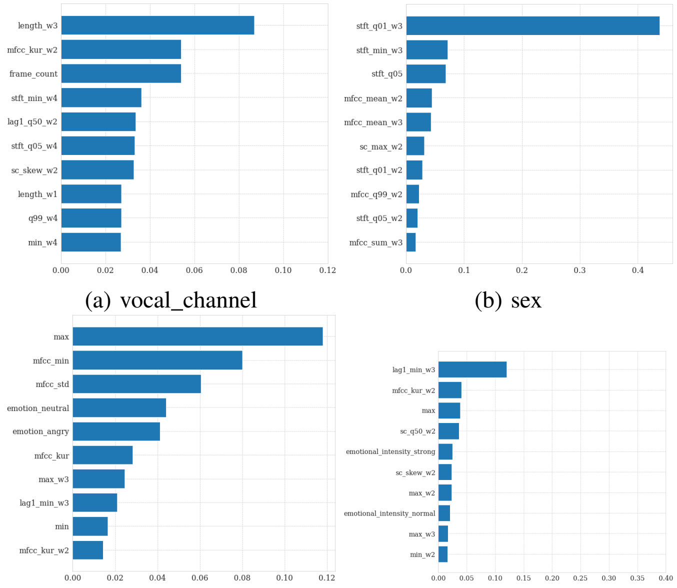


Fig. 8: Confusion matrix of NN classification.

*4) Ensemble Models:* Ensemble models, including Decision Tree Bagging (DTB), Random Forests (RF), AdaBoost (AB), and Light Gradient Boosting Machine (GB), were evaluated. The best results for each model are summarized in Tables XVIII, XIX, XXI, and XX.

A further analysis has concerned the computation of the importance of each input feature in RF's predictive performance (Figure 9), which allows to gain a more in-depth understanding of the information used by the model to discriminate between classes of the target variable.



(a) vocal_channel

(b) sex

(c) emotional_intensity

(d) emotion

Fig. 9: Top 10 most important input features for the classification of each target with Random Forest.

*5) Comparative Evaluation:* A comparative evaluation of the classification models was conducted for both binary and multi-class tasks. The accuracy of each model for each target is displayed in Figure 10.

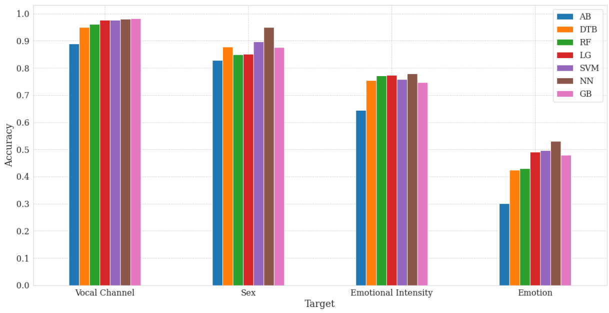
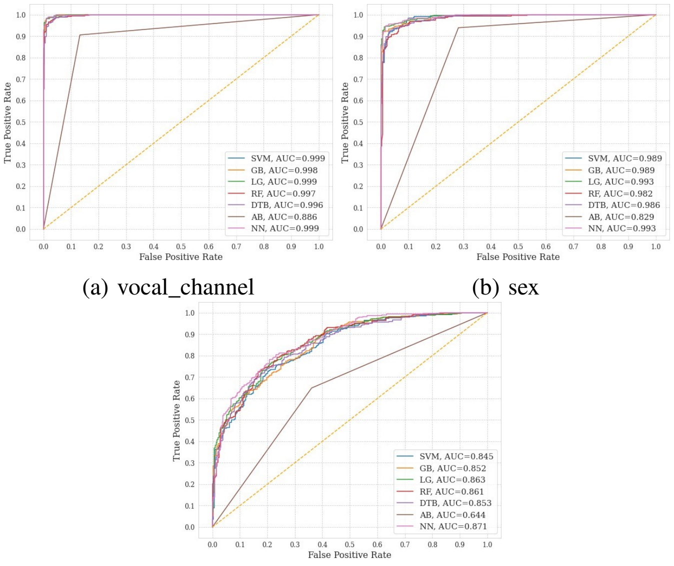


Fig. 10: Accuracy of classification models for each target.

The ROC curves and corresponding AUC for the binary classification tasks are shown in Figure 11, while the F1 scores for the multi-class task (emotion) are presented in Figure 12.



(a) vocal_channel

(b) sex

(c) emotional_intensity

Fig. 11: ROC curves of classification models for each binary target.

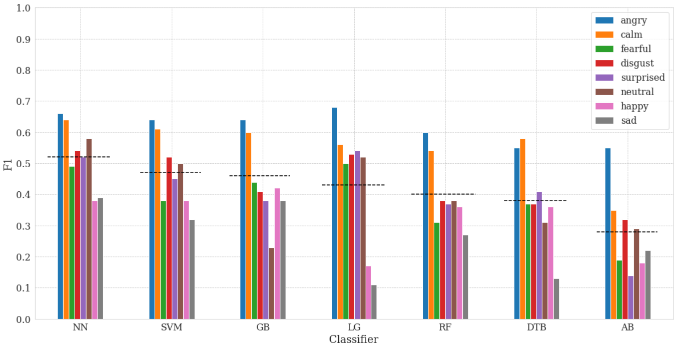


Fig. 12: F1 scores computed by each classifier for each value of the emotion target. Models are sorted w.r.t. their weighted averages F1, which are indicated by the black dashed lines.

Overall, the neural network models outperformed the baseline logistic regression and other classifiers in most tasks, particularly for the multi-class emotion classification. Ensemble models also demonstrated strong performance, with Light Gradient Boosting Machine (GB) achieving high accuracy across various targets. These results underscore the effectiveness of advanced classification techniques in emotion recognition tasks using the RAVDESS dataset.

## C. Clustering Results

The clustering analysis aimed to group similar patterns within the time series data extracted from the RAVDESS dataset. Both K-Means and Hierarchical Agglomerative Clustering algorithms were utilized to identify natural groupings in the data.

*1) K-Means Clustering:* The K-Means clustering algorithm was applied to the time series data, which were approximated using Symbolic Aggregate Approximation (SAX) for improved computational efficiency. Four versions of K-Means were tested, each with a different distance metric: Euclidean

TABLE XVIII: Best Results for Decision Tree Bagging

| Target | Criterion | Max Depth | Min Split | Min Leaf | Max Samples | Max Features | Weighted F1 | Accuracy |
|---|---|---|---|---|---|---|---|---|
| Vocal Channel | Entropy | 58 | 0.022 | 0.007 | 0.7 | 330 | 0.95 | 0.95 |
| Sex | Log-Loss | 51 | 0.010 | 0.0028 | 0.7 | 78 | 0.88 | 0.88 |
| Emotional Intensity | Gini | 16 | 0.013 | 0.016 | 0.6 | 381 | 0.75 | 0.75 |
| Emotion | Entropy | 23 | 0.027 | 0.0017 | 0.7 | 126 | 0.39 | 0.42 |

TABLE XIX: Best Results for Random Forest

| Target | Criterion | Max Depth | Min Split | Min Leaf | Max Features | Weighted F1 | Accuracy |
|---|---|---|---|---|---|---|---|
| Vocal Channel | Gini | 75 | 0.043 | 0.0044 | $\sqrt{N}$ | 0.96 | 0.96 |
| Sex | Entropy | 78 | 0.011 | 0.005 | $N$ | 0.85 | 0.85 |
| Emotional Intensity | Gini | 38 | 0.018 | 0.0040 | $N$ | 0.77 | 0.77 |
| Emotion | Log-Loss | 81 | 0.021 | 0.010 | $N$ | 0.41 | 0.43 |

TABLE XX: Best Results for Light Gradient Boosting

| Target | Boosting type | Estimators | $\eta$ | Leaves | Max depth | Weighted F1 | Accuracy |
|---|---|---|---|---|---|---|---|
| Vocal Channel | GOSS | 395 | 0.1 | 20 | 191 | 0.98 | 0.98 |
| Sex | GOSS | 422 | 0.3 | 21 | 45 | 0.88 | 0.88 |
| Emotional Intensity | GBDT | 467 | 0.2 | 16 | 64 | 0.74 | 0.74 |
| Emotion | GOSS | 432 | 0.1 | 49 | 35 | 0.46 | 0.48 |

TABLE XXI: Best Results for AdaBoost

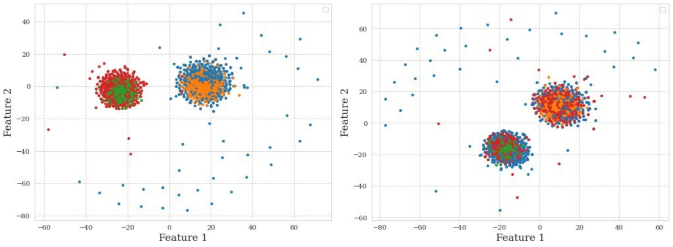
| Target | Learning Rate | Weighted F1 | Accuracy |
|---|---|---|---|
| Vocal Channel | 0.0001 | 0.89 | 0.89 |
| Sex | 0.0001 | 0.83 | 0.83 |
| Emotional Intensity | 0.0001 | 0.64 | 0.64 |
| Emotion | 0.0001 | 0.29 | 0.30 |

distance, Dynamic Time Warping (DTW), DTW constrained by the Sakoe-Chiba band, and DTW constrained by the Itakura parallelogram. The optimal number of clusters (k) was determined using the ”elbow method,” minimizing the Sum of Squared Error (SSE). The results for each distance metric are summarized in Table XXII.

TABLE XXII: K-Means Clustering

| Metric | k | SSE | Highest Purity |
|---|---|---|---|
| Euclidean | 4 | 84.86 | 0.75 (Vocal Channel) |
| DTW | 4 | 10.72 | 0.78 (Vocal Channel) |
| DTW (Sakoe-Chiba) | 4 | 55.84 | 0.78 (Vocal Channel) |
| DTW (Itakura) | 4 | 17.22 | 0.73 (Vocal Channel) |

The clustering obtained with DTW showed a significant decrease in SSE and an increase in purity with respect to vocal_channel. The clusterings using Euclidean distance and DTW are visualized with t-SNE in Figure 13.

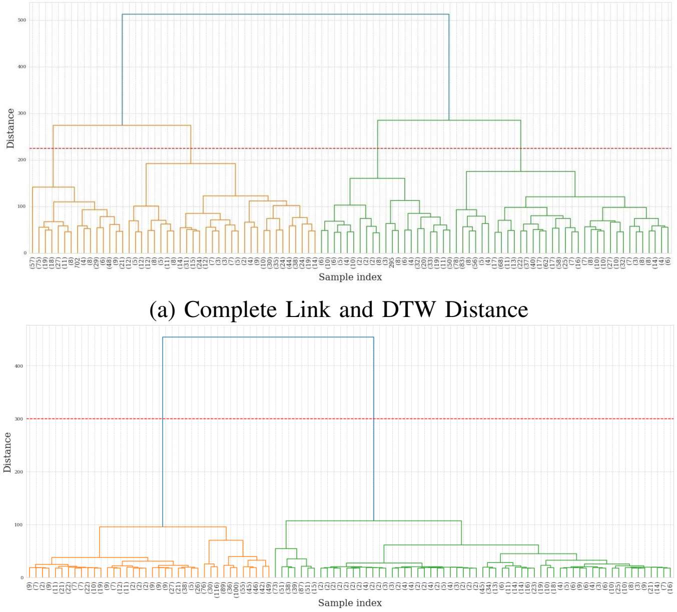


(a) Euclidean Distance

(b) DTW Distance

Fig. 13: K-Means clusterings (t-SNE).

*2) Hierarchical Agglomerative Clustering:* Hierarchical Agglomerative Clustering was performed using both Euclidean distance and DTW in combination with various proximity measures: Single Link, Complete Link, Group Average, and Ward’s method (only for Euclidean distance). The evaluation of potential clusterings was conducted through visual inspection of dendrograms. Optimal results were obtained using Complete Link for DTW and Ward’s method for Euclidean distance. The dendrograms for these methods are shown in Figure 14.

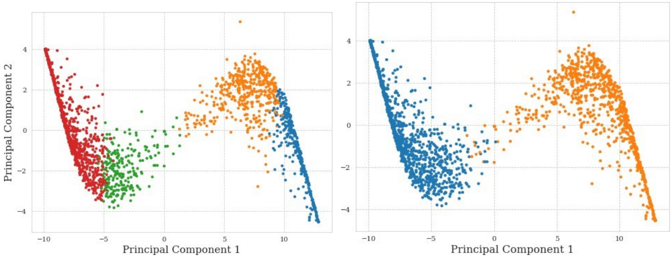


(a) Complete Link and DTW Distance

(b) Ward’s Method and Euclidean Distance

Fig. 14: Dendrograms using Complete Link and DTW (a) and Ward’s method and Euclidean distance (b). Red dashed horizontal lines indicate the cut locations.

Dimensionality reduction techniques such as PCA were used to visualize the clusters obtained from hierarchical clustering. The clusterings obtained with Complete Link and DTW, and Ward's method with Euclidean distance, are shown in Figure 15.



(a) Complete Link and DTW Distance (b) Ward's Method and Euclidean Distance

Fig. 15: Hierarchical clusterings (PCA).

Overall, the clustering results highlight the effectiveness of using DTW for capturing the temporal structure of the time series data, leading to more coherent and meaningful clusters. These findings provide a comprehensive view of the underlying patterns within the RAVDESS dataset, facilitating deeper insights into the data's structure and relationships.

## IV. Conclusion

In this report, I conducted a comprehensive analysis of the RAVDESS dataset, focusing on various data mining tasks, including anomaly detection, imbalanced learning, classification, and clustering. The study highlighted the significance of advanced data mining techniques in improving emotion recognition systems.

Key findings include the effectiveness of SMOTE in handling imbalanced data, significantly enhancing the performance of classifiers in recognizing minority classes. Among the classifiers, neural networks demonstrated superior performance, particularly for the multi-class emotion classification task, outperforming baseline models and other complex classifiers. Ensemble models, especially Light Gradient Boosting Machine (GB), also showed strong performance across various tasks, highlighting their robustness and accuracy in emotion recognition.

Clustering analysis using K-Means and Hierarchical Agglomerative Clustering revealed the importance of using appropriate distance metrics, such as Dynamic Time Warping (DTW), to capture the temporal structure of the time series data, resulting in more coherent and meaningful clusters.

Overall, the study underscores the importance of selecting suitable data preprocessing, balancing techniques, and classification algorithms to enhance the accuracy and robustness of emotion recognition systems. Future work could explore further optimization of deep learning models and the integration of additional modalities to improve the recognition of subtle emotional nuances.

## References

[1] S. R. Livingstone and F. A. Russo, "The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english," *PLOS ONE*, vol. 13, pp. 1–35, 05 2018.