Machine Learning-Based Prediction of Bee Swarming with Audio Feature Extraction

Hoang Trung Nguyen
FPT University Da Nang, Vietnam
de180300hoangtrungnguyen@gmail.com

Tran Ly Minh Hoang
FPT University Da Nang, Vietnam
minhhoangtran041105@gmail.com

Nguyen Thi Van Anh
FPT University Da Nang, Vietnam
anhntvde180371@fpt.edu.vn

Abstract

Although playing a critical role, bee populations are currently declining significantly. Swarming has a significant effect on the manufacturing procedure. Detecting swarming using conventional techniques, like temperature monitoring, takes a lot of work and time. Sound analysis has thus emerged as a viable strategy and has been extensively pursued. By applying machine learning (ML) and ensemble learning methodology and techniques to identify bee noises, this study seeks to strengthen swarming detection accuracy and efficiency. To assess sound signals from audio data gathered from bee colonies, features including MFCC, STFT, and chroma are retrieved. SVM, Random Forest, KNN, Gradient Boosting, Extra Trees, and Naïve Bayes are among the machine learning models that are taught to distinguish between swarming and non-swarming sounds, with the accuracy reaching the peak of 96.42% for the validation set and 99.48% for the test set and 99.57% for the validation one and 99.6% for the test one on SB1 and SB2, respectively. In addition, to boost the model performance, Ensemble Learning strategies including Stacking, Weighted Voting, Hard Voting, and Soft Voting are utilized, witnessing an overall accuracy of 97.33% on the validation set and 99.17% on the test set. The study's findings demonstrate that integrating these characteristics with machine learning models vastly increases the effectiveness of swarming detection, with the best-performing model yielding an F1-score of 99.6%, assisting beekeepers in managing their colonies proactively.

Keywords

Bee sound · Swarming detection · Audio feature extraction · Machine learning · MFCC · STFT · Chroma features

1 Introduction

Honey Bees play an important role in ecosystems in general and in agriculture in particular, especially in pollinating plants and maintaining biodiversity. In recent years, the bee population has

significantly declined due to several factors, such as diseases and inadequate care from beekeepers. Additionally, the occurrence of swarming behavior in bee colonies affects both their stability and production capacity.

Previously, the identification of swarming behavior relied solely on human observation, which required significant patience. However, this also made detecting swarming ineffective at large scales. As a result, non-invasive research on bee colonies has been strongly promoted. Experiments analyzing the sound produced by bee colonies can detect early signs of swarming, enabling timely and optimal interventions.

One commonly used feature for describing sound signals is MFCC (Mel Frequency Cepstral Coefficients), which is based on Mel frequency and has been proven effective in identifying sounds. For instance, Phan et al. demonstrated that MFCC plays an important role in bee sound recognition (Phan, T.T.H., Nguyen, H.D., Doan, D.N.: Evaluation of feature extraction methods for bee audio classification. ICIT 2022). Furthermore, the combination of MFCC with other methods has shown promising results in accurately detecting swarming behavior (Truong, T.H., et al.: A deep learning-based approach for bee sound identification. Ecological Informatics, 2023).

Additionally, the use of STFT (Short-Time Fourier Transform) helps monitor frequency fluctuations in bee colonies, which can also be used to detect abnormalities (Ferrari, S., et al.: Monitoring of swarming sounds in bee hives for early detection. Computers and Electronics in Agriculture, 2008). Moreover, the use of chroma features allows for tracking changes by observing the transformation of critical frequency bands. Phan et al. utilized chroma in combination with MFCC and STFT to detect the state of the bee colony (Phan, T.T.H., et al.: Short-time Fourier transform for detecting the queen bee state. Intelligence of Things: Technologies and Applications, 2024).

In previous research, many algorithms have been applied with the aim of combining features and algorithms. Alongside using individual models such as KNN, SVM, NB, RF, Extra Trees, and GB, we have employed ensemble methods—such as Soft Voting, Hard Voting, Weighted Voting, and Stacking—to evaluate their effectiveness. These methods ensure greater stability and better performance when applied at a large scale for detecting both swarming and non-swarming behavior.

2 Related Work

In recent years, a lot of research has been dedicated to using audio analysis and machine learning to keep an eye on bee colonies and spot swarming behavior. Phan et al. (2022) showed that Mel Frequency Cepstral Coefficients (MFCC) are really effective at picking up the sounds bees make when they swarm. Their findings opened the door for using audio feature extraction techniques in analyzing bee sounds.

Building on this foundation, Truong et al. (2023) took it a step further by combining MFCC with other features, like chroma, to boost accuracy. Additionally, Ferrari et al. (2008) looked into using Short-Time Fourier Transform (STFT) to track frequency changes, which provided more insights into swarm detection through audio analysis.

Our focus is on choosing the best features to make the swarm detection process more effective. We aim to filter out features that help the model cut down on noise and enhance accuracy. Understanding feature importance allows us to pinpoint the most crucial features. Fisher's criterion employs statistical methods to evaluate and differentiate key features. Mutual information assesses how relevant features are to the labels. Plus, Mutual Information captures both linear and nonlinear relationships, making it a practical tool for the feature selection phase. PCA (Principal

Component Analysis) helps us reduce the number of features while keeping the important ones intact. Through correlation and VIF analysis, we compare feature pairs and eliminate one to prevent multicollinearity. With this thoughtful feature selection strategy, we've ensured that our model is not only optimized but also better equipped to detect swarming events.

3 Methodology

Figure 1 is a general diagram of the process of recognizing swarming signals of beehives based on bee sounds using machine learning (ML) algorithms. The whole process includes 3 stages. In stage 1, we collect data using feature extraction methods such as MFCC (Mel-Frequency Cepstral Coefficients), STFT (Short-Time Fourier Transform), and Chroma. Feature extraction is the process of transforming raw audio data into a set of features that are more relevant for machine learning algorithms.

The next step is feature selection. This process aims to trim down the data's dimensionality while keeping the most crucial information intact. By picking out the most relevant features, we can boost the model's performance and simplify the data. This step is vital for enhancing the efficiency and accuracy of the machine learning algorithms that help us spot swarming behavior in beehives.

Moving on to the final stage, we have model training. Once we've divided the data into training, validation, and test sets, we employ several models to train on the data. After the training phase, we assess the models using both the validation and test data. The top-performing models are then chosen for ensemble learning, which merges multiple models to elevate overall performance and accuracy. In the end, we compare the selected models to find the best fit for the task, ensuring that the final system is both robust and precise in detecting signals of bee swarming.

3.1 Feature Extraction Techniques

3.1.1 MFCCs

The Mel-Frequency Cepstral Coefficients (MFCCs) is a very effective approach to extracting sound features. The general usage of this method is to detect human speech perception, but now due to its effectiveness in capturing the frequency of sound signal spectral features, it's been utilized in audio processing. In this context, we use MFCCs to break down the audio signal by applying multiple techniques. it involves preprocessing the signal through pre-emphasis, framing, and windowing it before turn into the frequency domain using the Fast-Fourier Transform. Then, a Mel-Filter Bank is applied to capture the relevant frequency components, follow by apply logarithmic compression, and then finally, the Discrete Cosine Transform is used to decorrelate the features.

3.1.2 STFT

The Short-Time Fourier Transform (STFT) is a nifty technique for sound extraction that shifts signals from the time domain to the frequency domain in small chunks. This approach allows us to keep tabs on how frequencies change over time.

When it comes to analyzing the sounds made by bees, STFT is particularly useful. It helps us pick up on frequency shifts in the hive's sounds, which can be key in spotting any unusual signs of swarming behavior.

STFT works by breaking the original signal into small frames, with each frame overlapping the one before it. This overlap helps to ensure that we don't lose any important information at the

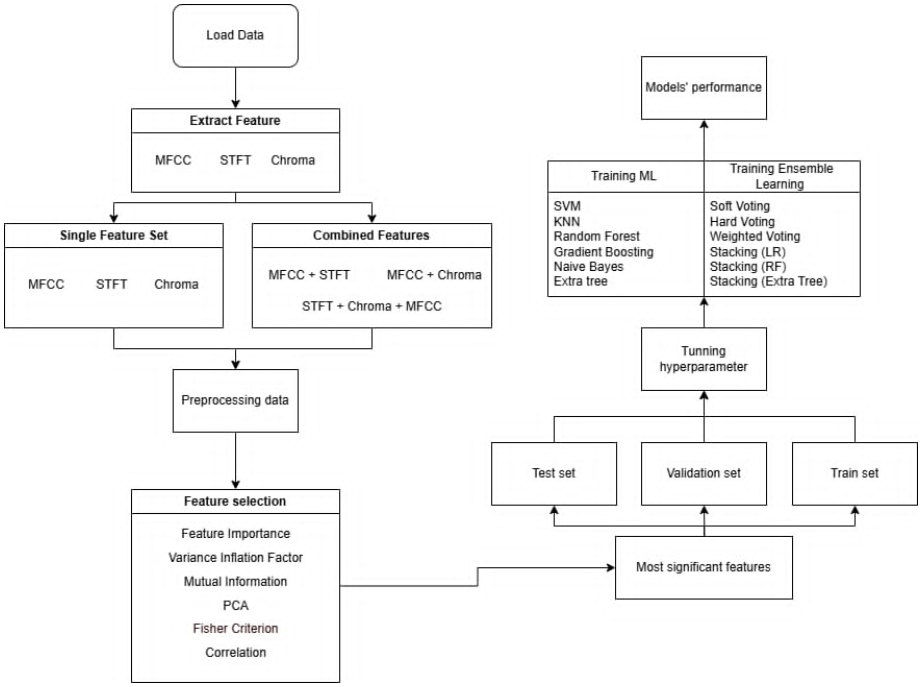


Figure 1: Bee swarming classification pipeline.

edges of the frames. Typically, this overlap is set at a percentage—like 50%—which means each frame shares half of its data with the previous one. Once the frames are set, STFT applies a window function, such as Hamming or Hanning, to each frame. This step is crucial as it helps to minimize spectral leakage by gently tapering the signal at the edges, which in turn reduces the energy that leaks into other frequencies.

Then, STFT uses FFT on each frame to convert the signal from the time domain to the frequency domain in each frame. FFT 1D calculation formula is:

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j\frac{2\pi}{N}kn} \quad \text{where} \quad \begin{aligned} X(k) &\text{ is the frequency domain output at index } k, \\ x(n) &\text{ is the time domain input signal at index } n, \\ N &\text{ is the total number of samples in a frame,} \\ j &\text{ is the imaginary unit,} \\ k &\text{ is the frequency index,} \\ n &\text{ is the time index.} \end{aligned}$$

The result of FFT will be a sequence of frequency values, describing how the frequency components of the signal are distributed in the frequency domain. The amplitude can tell you the intensity of each frequency and the phase will tell you how each frequency changes over time.

STFT generates a spectrogram, which visually represents how the frequency content of the signal evolves over time. Figure 3 shows a spectrogram of a swarming sample. In this spectrogram,

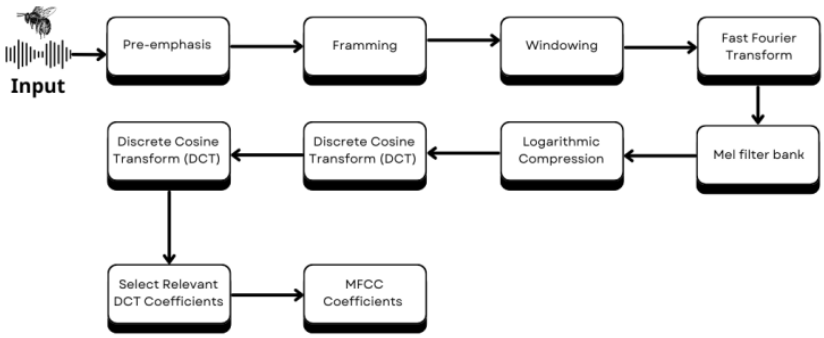


Figure 2: MFCCs Feature selection pipeline.

the vertical axis represents the frequency components, while the horizontal axis corresponds to time intervals. The intensity of each frequency is indicated by the color or brightness at each time frame, with amplitude providing information about the strength of each frequency and phase showing how the frequencies change over time.

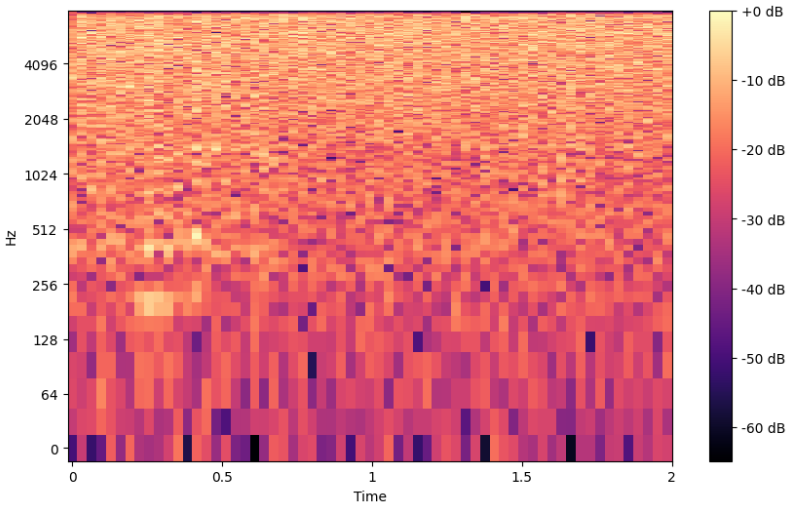


Figure 3: Spectrogram of swarming sound data.

3.1.3 Chroma

In audio processing applications such as speech recognition, chroma is a commonly used feature extraction method. By mapping the entire frequency spectrum to 12 pitch classes (C, C#, D, D#, ..., B), regardless of octaves, chroma extracts the pitch and timbre structure of the signal and represents the audio content according to the pitch classes of the audio signal. First, the audio is transformed into the frequency domain for analysis using the Short-Time Fourier Transform

(STFT). Then, the energy spectrum is converted into 12 pitch classes to represent the distribution within the signal.

3.2 Feature selection

3.2.1 Importance Score

Importance Score is a metric that calculate the contribution of each feature to the model's prediction. Each feature have a score that reflects its importance in the classification process. In this study, we use the Random Forest Classifier and the Gini Impurity criterion to assess feature importance. The model evaluates how much each feature contributes to reducing impurity when it is used to split the nodes in the decision trees.(Louppe, G., Wehenkel, L., Sutera, A., & Geurts, P. (2013). Understanding variable importances in forests of randomized trees. Advances in Neural Information Processing Systems (NeurIPS), 26.)

The Gini Impurity for a dataset S is calculated as:

$$Gini(S) = 1 - \sum_{i=1}^{m} p_i^2$$

where:

- m is the number of classes in the dataset S,
- p_i is the proportion of elements of class i in the dataset S.

After calculating the Importance Score, we rank and choose a threshold to remove or apply PCA (Principal Component Analysis) to the columns, reducing the dimensionality of the data and selecting the most important features to include in the models.

3.2.2 Variance Inflation Factor

VIF (Variance Inflation Factor). The Variance Inflation Factor is a value that is used to check the multicollinearity in linear regression models. When independent variables in a model are highly correlated with each other, this can make it difficult to estimate parameters accurately and reduce the accuracy of the model.(Marquardt, 1970 and O'Brien, 2007) The VIF for a variable is calculated as: The formula for calculating the VIF of an independent variable X_i is given by:

$$VIF(X_i) = \frac{1}{1 - R_i^2}$$

Where:

- $VIF(X_i)$ is the VIF of the independent variable X_i,
- R_i^2 is the R-squared value when the variable X_i is predicted using the remaining independent variables in the model.

A high VIF value shows a high correlation between the independent variable and other variables, which causes to multicollinearity problems.

3.2.3 Mutual Information

MI (Mutual Information) is a quantity that measures a relationship between two variables. It can be broken down as measuring how much two random variables have in common. Unlike correlation, MI captures both linear and non-linear relationships, making it a very useful technique for analyzing complex datasets(Cover & Thomas, 2006). In general, higher MI values show a stronger correlation between two variables, while in the other hand, the lower value or the zero indicated complete independence. (Shannon, 1948)

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)}$$

In this article, our aim is to use the MI to evaluate the relationship between the extracted features and the labels - the presence of swarming or non-swarming. By using this, we can identify which feature has the most impact on two separate cases, which will then become very useful in feature selection and enhancing the performance of the models.

3.2.4 Principal Component Analysis

PCA (Principal Component Analysis) is a data analysis technique used to reduce the dimensionality of data while retaining most of the important information in the data. PCA helps to find new features of the data, which are uncorrelated with each other and can explain most of the variation in the original data (Pearson, 1901). The more components retained in PCA, the more information is preserved from the original dataset. Each principal component captures a certain amount of variance, and by keeping more components, we maintain more of the information in the data. However, this also increases the dimensionality, which can lead to more complexity for the model. (Jolliffe, 2002)

3.3 Machine learning methods

3.3.1 K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a simple and effective machine learning algorithm used for both classification and regression tasks. It works by finding the K closest data points to a given input and making predictions based on the majority vote of the K nearest neighbors. The label of the new data point is determined by the most frequent label among its K nearest neighbors. One of the disadvantages of KNN is that it does not perform well on imbalanced datasets, as the majority class can dominate the voting process. (Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21-27.)

KNN does not require a training phase, making it suitable for tasks where model simplicity is important. However, the choice of K (the number of neighbors) and the distance metric (e.g., Euclidean distance) can significantly impact the model's performance. Adjusting these hyperparameters is critical to achieving good results, and the model's performance can decrease with larger datasets due to the computational cost of calculating distances for each query point.

3.3.2 Support Vector Machine

Support vector Machine is one type of supervised machine learning model used to evaluate classification and regression problems. Introduced by Vladimir Vapnik and Alexey Chervonenkis, SVM

has proven to be a very highly effective model and in some cases can achieve better performance than KNN. The main idea of SVM is to find a hyperspace or hyperplane that will divided data into two separate classes while maximizing the distance between the nearest data point of each class and the margin of its hyperplane.(Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297.) The whole idea is shown below.

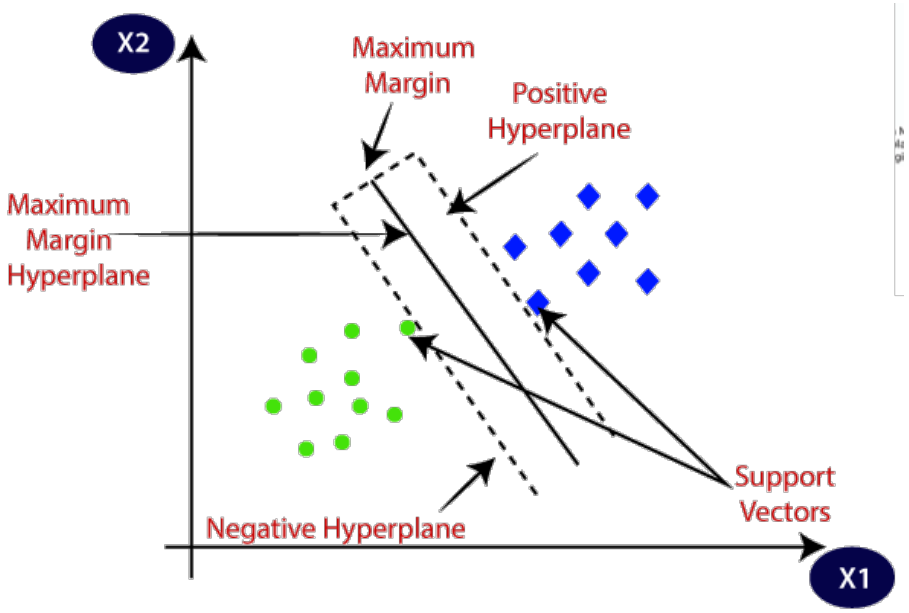


Figure 4: Support vector machine.

In general, SVM was initialized to binary class classification problems, but in some cases, they can be extended to handle multi-dimensional spaces by using whats called the "kernel trick". Particularly in our problem, we use the RBF kernel due to its effectiveness, which has been proven through many trials.

3.3.3 Naive Bayes

The Naive Bayes (NB) is a supervised machine-learning algorithm based on the Bayes theorem that is used for classification problems. the main idea of this method is to find the principle of probability to perform classification tasks.(Lewis, D. D. (1998). Naïve Bayes at Forty: The Independence Assumption in Information Retrieval. Machine Learning: ECML-98, 4-15.)

in our study, the Naive Bayes plays a role in classifying the presence of swarming or non-swarming states based on extracted features through various methods. The NB is particularly sensitive to irrelevant features because it assumes the feature is independent. However due to its simplicity, NB serves as a baseline model to compare performance against other models.

3.3.4 Random Forest

The Random Forest (RF) algorithm is a widely used machine learning model in both classification and prediction scenarios. it operates alongside an ensemble of decision trees, each takes samples using the bagging technique and then trains on a random subset of the dataset. The final result is obtained by aggregating all the outputs of the created decision tree through majority voting for the classification case or averaging for the regression case. (Tin Kam Ho, "The random subspace method for constructing decision forests," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 8, pp. 832-844, Aug. 1998, doi: 10.1109/34.709601)

In our study, the RF plays as a key method alongside the Ensemble learning method to predict the swarming behavior based on extracted features. The model constructed multiple decision trees using bagging, and voting for the final result. This method helps to reduce overfitting, noise effectiveness, increase generalization,... and many problems other machine learning model encountered.

3.4 Ensemble learning methods

Ensemble learning is a machine learning method that combines multiple machine learning models to improve predictive performance compared to each model alone. The goal of ensemble learning is to use multiple models to make more accurate predictions. Ensemble learning methods can help reduce problems such as overfitting, variance, and bias, thereby improving performance on real-world datasets.

3.4.1 Soft Voting

Soft Voting is an ensemble learning technique in machine learning that combines the predictions from multiple models to make a final decision. Unlike Hard Voting, where the majority class is selected, Soft Voting aggregates the predicted probabilities for each class from all models in the ensemble and averages them. The class with the highest average probability is chosen as the final prediction. (Dietterich, T. G. (2000). Ensemble methods in machine learning. Proceedings of the First International Workshop on Multiple Classifier Systems, 1–15.)

$$P(y = c) = \frac{1}{N} \sum_{i=1}^{N} P_i(y = c)$$

where:

- $P(y = c)$ is the final predicted probability for class c.
- N is the number of models in the ensemble.
- $P_i(y = c)$ is the probability predicted by the i-th model for class c.

3.4.2 Hard Voting

Hard Voting is an ensemble learning technique used to make predictions by combining the results from multiple individual models. In Hard Voting, each model in the ensemble has a vote for a class label, and the class label that receives the majority of votes is chosen as the final prediction. Hard Voting is simple to implement and works well when the individual models in the ensemble have similar performance levels. However, it doesn't take into account the probabilities of the

predictions, unlike Soft Voting, which uses probability values for a weighted decision. (Dietterich, T. G. (2000). Ensemble methods in machine learning. Proceedings of the First International Workshop on Multiple Classifier Systems, 1–15.) The difference between soft and hard voting is shown in Figure 5.

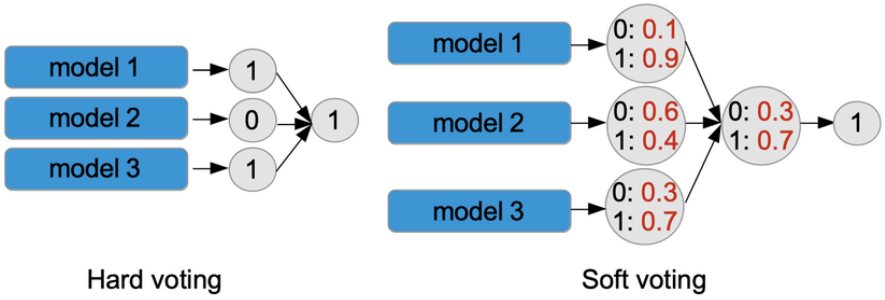


Figure 5: The difference between soft and hard voting.

3.4.3 Weighted Voting

Weighted Voting is a method in machine learning used to combine multiple prediction models to improve accuracy. In Weighted Voting, each member model is assigned a weight "w" that represents the importance of that model in making the final prediction. When predicting, instead of just averaging the probabilities as in Soft Voting, Weighted Voting multiplies the predicted probabilities of each model by the corresponding weights, and then adds them together to calculate the final probability.

$$P(y = c) = \frac{\sum_{i=1}^{N} w_i \cdot P_i(y = c)}{\sum_{i=1}^{N} w_i}$$

where:

- $P(y = c)$ is the final predicted probability for class c.
- N is the number of models in the ensemble.
- w_i is the weight assigned to the i-th model.
- $P_i(y = c)$ is the probability predicted by the i-th model for class c.

3.4.4 Stacking

Stacking, also known as stacked generalization, Is one of the ensemble learning method that combines multiple base classifiers models to improve the performance. introduced by Wolpert in 1992 (Wolpert, D. H. (1992). Stacked generalization. Neural Networks, 5(2), 241–259.),

The main idea is first to training base learners (base classifiers model). These classifiers can be the same algorithm with different parameters or different algorithms. The prediction results were then collected to form a new dataset. final, a meta-learner (level 1 model), is trained on this new dataset and combines the base learners's predictions to produce the final output.

3.4.5 Extra Tree

Extra Tree, as Extremely randomized trees, is an ensemble learning method which first introduced by Geurts, Ernst, and Wehenkel in 2006 (Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. Machine Learning, 63(1), 3–42.)

The stacking introduced by Wolpert (1992) method is to train the base model independently, and their result is used as an input feature for the meta-learner. This way stacking leverages the diversity of different classifiers to improve generalization. In 2006, Geurts introduced another approach to this method by applying additional randomness in feature selection and split point determination. This extreme randomization helps reduce variance and improve generalization.

4 Data description

This article mainly conducted experiments on two datasets with two distinct scenario: SB1 (Swarming Bee sound 1) and SB2 (Swarming Bee sound 2), containing audio recordings of bee activity, capturing variations in acoustic patterns associated with swarm behavior. Overall, SB1 challenges the generalization ability, while SB2 focuses on diverse training data to improve the comprehensiveness of the model itself. Table 1 presents a detailed overview of the dataset, including the distribution of samples across training, testing, and validation sets for both Swarming and Non-Swarming categories. The dataset is divided into two subsets, SB1 and SB2, each recorded at different sampling rates of 16000 Hz and 32000 Hz, respectively.

Dataset	Category	Train	Test	Val
SB1	Non_Swarming	6300	2400	1800
SB1	Swarming	6300	2400	1800
Sampling rate SB1: 16000				
SB2	Non_Swarming	9329	5600	2200
SB2	Swarming	9673	5600	2249
Sampling rate SB2: 32000				

Table 1: Dataset Information with Sampling Rates

The SB1 is designed to test the model generalization capability by completely separating the training, testing, and validation data based on 3 main factors: data collection time, location, and swarming state. This ensures that the model will always encounter unseen data in the testing and validation phases, creating a challenge to test its robustness in handling real-world scenarios.

The SB2 in another way emphasized diversity within the training set by collecting samples on different days and across various recording devices. This approach aims to capture the extensive range of non-swarming bee behavior, increasing in model's adaptability. by maintaining the independence of training, testing, and validation set. SB2 guarantees an unbiased evaluation of the model's performance.

5 Hybrid Feature Selection and Dimensionality Reduction

5.1 Problem

5.1.1 MFCC

After experimenting with some methods to reduce the number of features while still maintaining the performance, we finally came up with a new approach to change the feature number. In three training, testing, and validation sets, we use the Mutual information technique to calculate each feature's contribution to the label (Swarming or non-swarming). The contribution of each feature on the label is not the same, since some features may not play an important role in the training set but can still carry information for the other sets. So when we try to reduce the number of features using feature importance score as a threshold to remove those mean values is lower, we remove some features carrying information in other sets in some cases. Figure 6 describes the contribution of each feature in 3 sets.

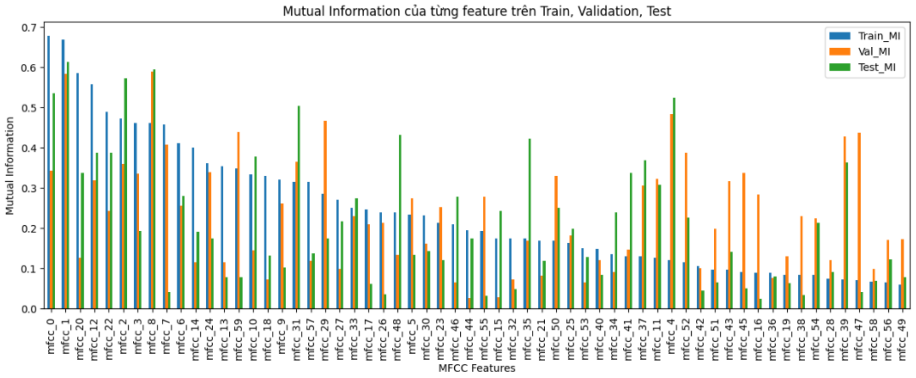


Figure 6: MI table

So to avoid this problem. We tried various of method and finally decided to create a hybrid approach where we combined the feature importance with PCA to lower the amount of features.

5.1.2 STFT

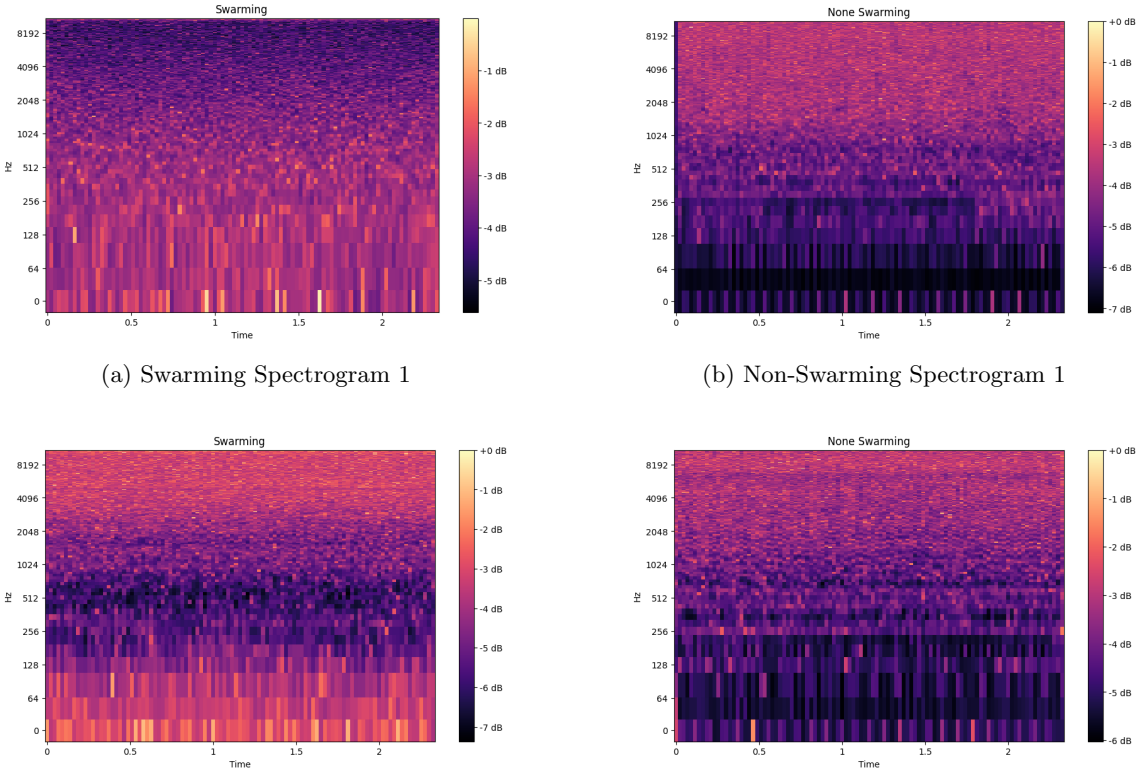
Figure 7 shows some sample spectrograms of the swarming and non-swarming samples. After conducting an in-depth study and observing the spectrograms of both swarming and non-swarming audio samples, we identified three primary features extracted from the Short-Time Fourier Transform (STFT) for further analysis:

- Mean (Average Amplitude): The mean represents the average amplitude across each frequency band over the entire audio sample. A high mean value indicates that the amplitude at that frequency band is consistently strong, suggesting that the sound at that frequency is dominant. Conversely, a low mean value implies weaker amplitude, indicating that the sound at that frequency is less prominent.

- Variance: Variance measures the dispersion of amplitude values around the mean within each frequency band. A high variance indicates significant fluctuations in amplitude over time at that

frequency, suggesting instability in the sound. A low variance reflects more stable amplitude values, indicating greater consistency in sound intensity at that frequency.

- Max (Maximum Amplitude): The max value captures the peak amplitude observed in each frequency band. A high max value suggests the presence of distinct energy peaks, where certain moments exhibit strong sound intensity at that frequency. In contrast, a low max value implies the absence of noticeable energy peaks, indicating more stable or weaker sound intensity at that frequency.



(a) Swarming Spectrogram 1

(b) Non-Swarming Spectrogram 1

(c) Swarming Spectrogram 2

(d) Non-Swarming Spectrogram 2

Figure 7: Comparison of Swarming and Non-Swarming Spectrograms

5.2 Methodology

5.2.1 Feature selection

A Random Forest Classifier was trained to calculate the feature importance. Features with a higher importance score than the mean threshold are saved, while those with lower scores were processed using the PCA.

5.2.2 Dimensionality Reduction with PCA

Instead of remove the less importance features, PCA was applied to reduce their dimension while keeping as much variance as possible. Multiple trials were set up to decide the best number of components and end with five components.

5.2.3 Correlation and Multicollinear Features

In audio signals, an increase in amplitude at a specific frequency, such as 140 Hz, often corresponds with increased amplitudes at neighboring frequencies, like 130 Hz and 150 Hz. This phenomenon results in high correlation between adjacent frequency features, leading to multicollinearity. Furthermore, extracting three features — mean, variance, and max — from the same spectrogram may introduce redundancy in information, further contributing to multicollinearity.

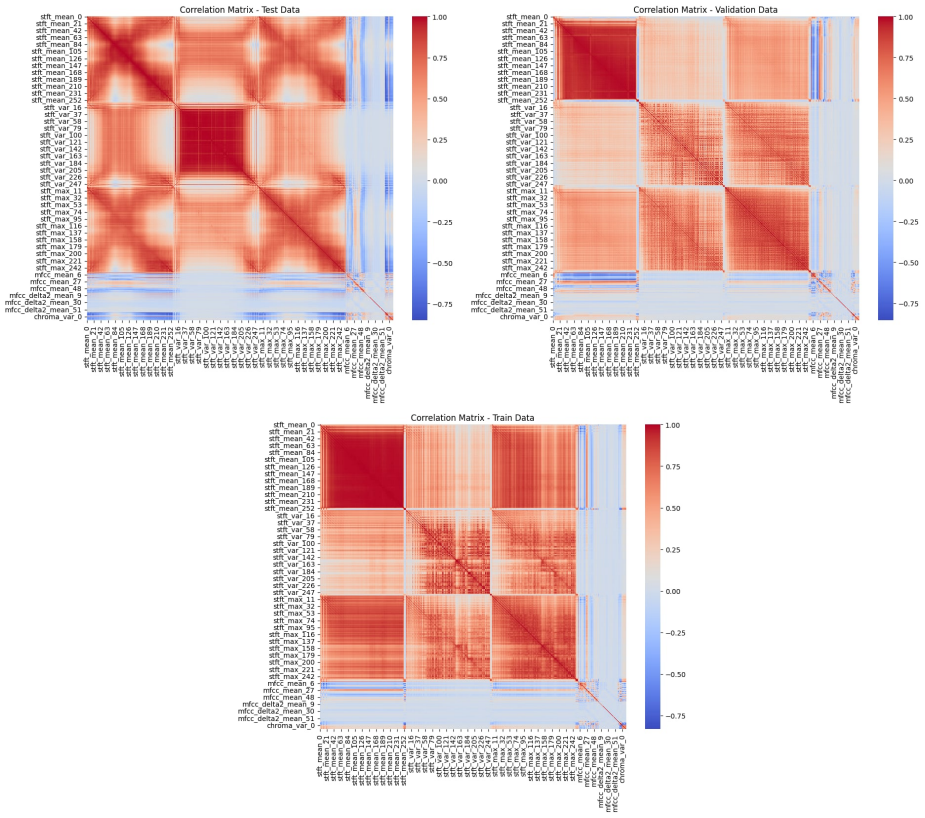


Figure 8: Heatmap of train, test, and validation process on STFT

In Figure 8, a significant number of features exhibit high correlation, indicating a strong likelihood of multicollinearity. This phenomenon can adversely affect the performance of machine learning models and linear regression by undermining the assumption of independent predictors. Specifically, multicollinearity may lead to instability in coefficient estimates, reduce the interpretability of the model, and increase variance in predictions.

5.2.4 Model Performance

After training with the optimized hyperparameters, the model achieved the following performance, shown in below table:

RF (FS)	99.48	99.49	92.56	92.05	Features	23
RF (FS + PCA)	99.36	99.36	96.42	96.3		28

Figure 9: Performance Report

The result shows strong generalization capabilities, with a significant increase in validation set performance, to prevent overfitting.

6 Experiments and Results

6.1 Experiments

In this study, we tried various techniques to evaluate performance towards 2 datasets: SB1 and SB2. Aiming to maximize the performance of machine learning and ensemble learning models by observing the Accuracy and F1 score of each one.

Initially, we use various visualization techniques to analyze the data structure and assess the distinctions between the two sample categories in each dataset. This way we can be enabled to establish a comprehensive approach for further investigation. The visualized results are presented in the subsequent figures.

After extracting the features using the designated extraction methods, we performed data preprocessing to ensure consistency and accuracy. These processed features were then used to train machine learning and ensemble models to evaluate their performance effectively. Figures 9 to 12 present fundamental visualizations of our dataset, providing insights into its underlying structure and characteristics.

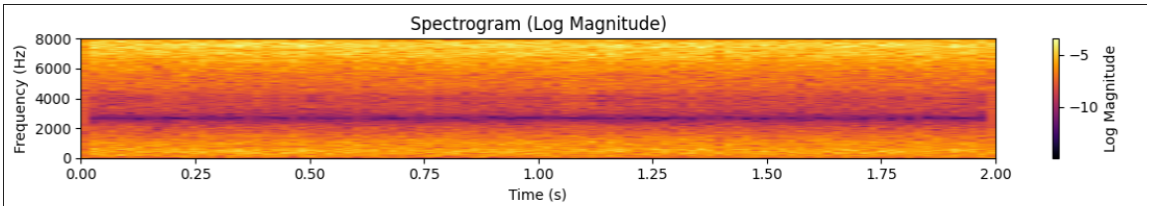


Figure 10: Spectrogram STFT.

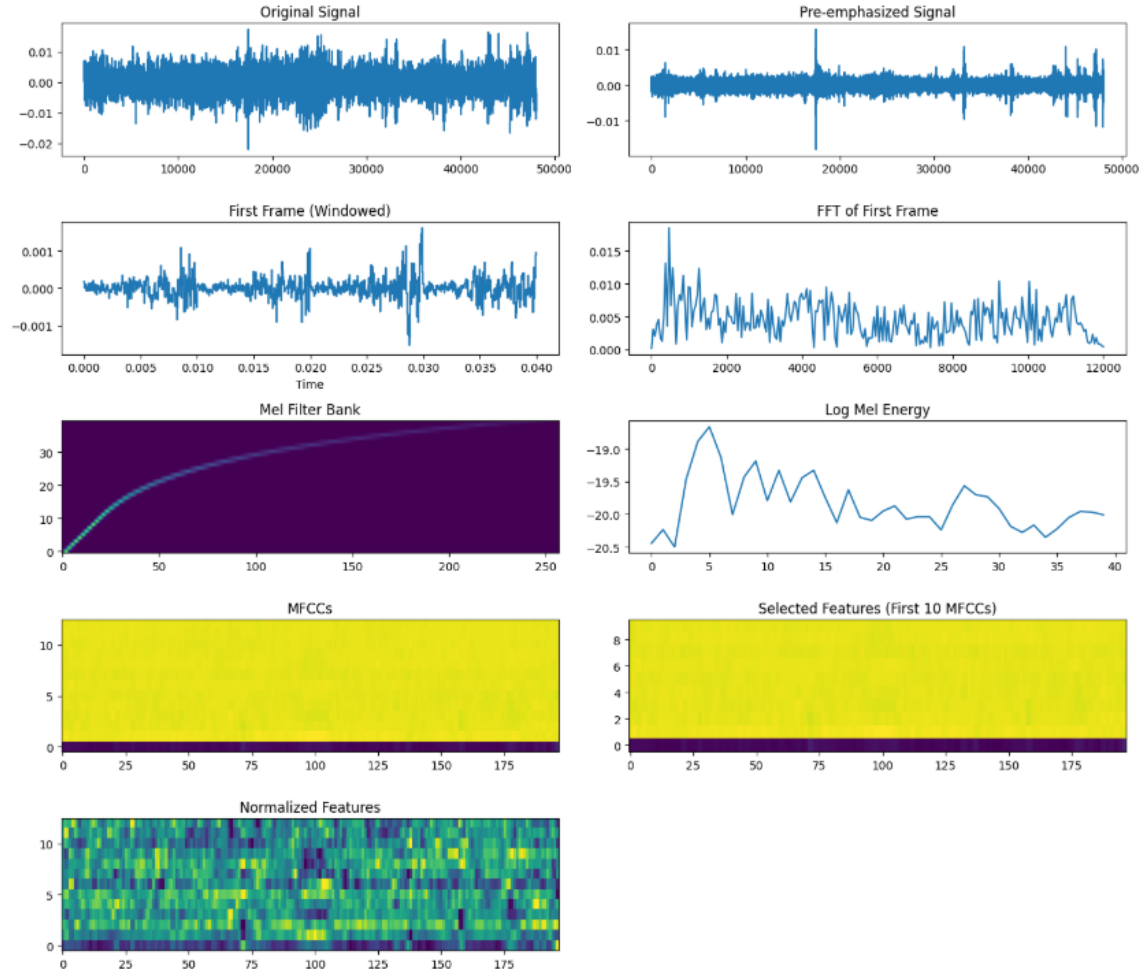


Figure 11: Swarming Signal using MFCC.

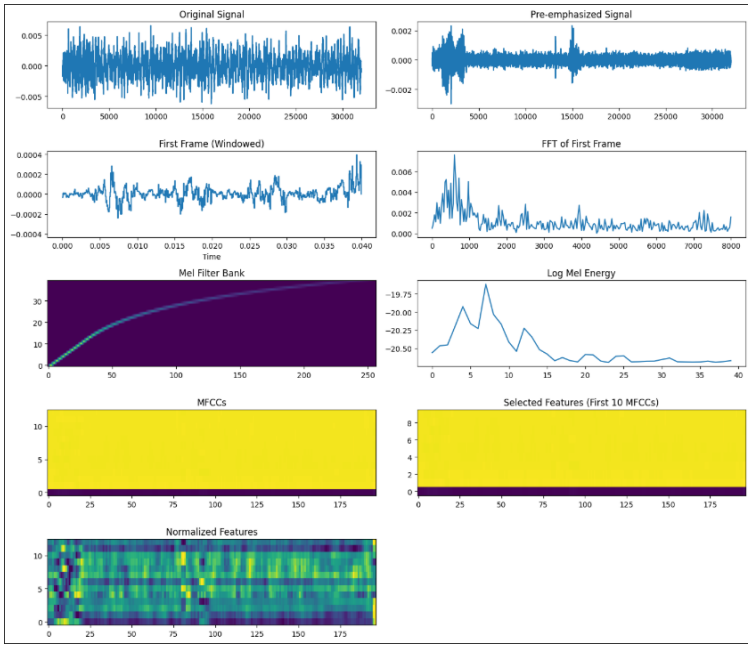


Figure 12: Non-Swarming Signal using MFCC.

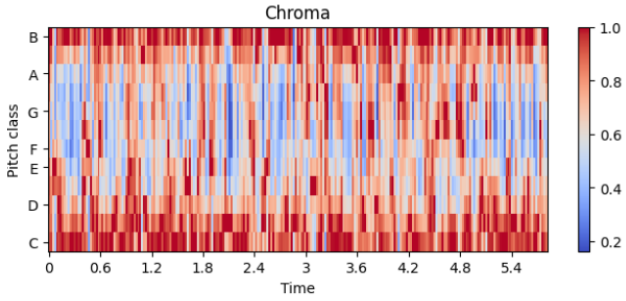


Figure 13: Spectrogram Chroma.

6.2 Results

6.2.1 SB1 model performace

In regard to SB1, RF exhibits the best results overall, particularly with MFCCs, who achieve over 99 accuracy with FS and RF (FS + PCA). More precisely, RF (FS) achieves an astounding 99.48 accuracy with an F1 score of 99.49 on the test sample, while RF (FS + PCA) is slightly lower at 99.36 for both accuracy and F1 score. MFCC also does wonders for SVM, yielding a remarkable 97.93 score for accuracy and 97.99 for F1. Chroma features, on the other hand, do not do so well: the best performing model is Naive Bayes with a 66.21 accuracy and 64.91 F1 score. These ensemble

methods, along with Hard Voting and Weighted Voting, seem to help with performance on MFCC but do not do as well with STFT and Chroma features. The conclusion one can draw from this is that MFCC is the most optimal feature extraction method for advanced features selection and ensemble models. All results shown in figure 14.

SB1												
	MFCC				STFT				Chroma			
	Test		Validation		Test		Validation		Test		Validation	
Model	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1
KNN	96.3	96.44	97.42	97.48	81.95	81.31	**75.50**	**74.11**	60.28	60.26	56.06	55.25
SVM	97.93	97.99	96.31	96.43	**83.43**	**82.94**	74.44	72.70	61.95	61.84	53.14	53.11
NB	92.13	92.67	96.03	96.1	70.77	67.92	50.31	50.32	**66.21**	**64.91**	**63.39**	**62.60**
RF (FS)	**99.48**	**99.49**	**92.56**	**92.05**	73.81	71.90	76.17	75.30	51.46	51.45	54.54	54.64
RF (FS + PCA)	**99.36**	**99.36**	**96.42**	**96.3**	62.39	56.19	62.39	5619	61.83	61.82	53.08	51.07
Soft Voting	92.34	92.3	90.44	90.1	67.16	63.06	74.67	73.04	63.40	63.39	54.36	52.80
Hard Voting	99.88	94.1	93.75	99.88	61.39	59.74	51.33	51.10	63.11	63.06	53.67	51.48
Weighted Voting	92,34	92,35	90,39	90,4	67.16	63.06	74.64	73.02	63.23	63.23	53.67	51.87
Stacking	92,34	92,36	90,08	91,2	57.82	57.34	49.03	47.72	61.25	61.21	53.89	52.64
Extra Tree	99,57	99,57	93,89	93,53	67.01	62.85	75.03	73.37	x	x	x	x

Figure 14: SB1 model performance.

6.2.2 SB2 model performance

Per the performance comparison analysis with respect to the SB2 dataset, MFCC features are best compared with STFT and Chroma features. The Extra Tree model obtains the best results with MFCC featureson the test set with an accuracy of 99.6, an F1 score of 99.6, and is closely followed by RF who obtained 97.3 and an F1 score of 97.37. Out of all conventional machine learning models, the Naive Bayes model (NB) does surprisingly well.

For STFT, the pre-trained SVM model performed best on the test set with 94.42 accuracy and F1 score of 94.40. This means STFT features are more appropriate for SVM than any other model. On the other hand, Chroma features perform the worst for all models with Naive Bayes being the least worst at 94.93 F1 score and 94.92 accuracy.

Ensemble techniques like Hard Voting and Weighted Voting enhances the performance of MFCC and STFT but do not have much effect on Chroma features. The results confirm that MFCC is still the best feature extraction method to use in predicting the swarming state of bees when paired with sophisticated models such as Extra Tree and Random Forest. All reuslt shown in figure 15.

SB2												
	MFCC				STFT				Chroma			
	Test		Validation		Test		Validation		Test		Validation	
Model	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1
KNN	94.6	94.88	98.63	98.66	82.18	81.60	81.57	80.95	91.44	91.44	52.19	51.55
SVM	75.21	80.03	97.39	97.49	**94.42**	**94.40**	**94.54**	**94.52**	91.02	91.02	54.24	53.47
NB	97.07	97.15	98.54	98.57	65.38	64.54	80.58	80.17	**94.93**	**94.92**	44.37	43.59
RF	**97.3**	**97.37**	**99.57**	**99.58**	77.73	76.68	70.76	70.58	91.83	91.83	53.65	53.02
Soft Voting	95.43	95.47	98.22	98.26	91.66	91.60	93.54	93.51	93.03	93.03	51.75	50.97
Hard Voting	96.46	96.52	99.62	99.62	87.21	87.21	94.87	94.86	92.69	92.69	51.99	50.99
Weighted Voting	95.24	95.24	98.2	98.2	93.54	93.51	91.66	91.60	92.51	92.51	**52.71**	**51.88**
Stacking (RF)	95.29	95.29	98.34	98.34	81.36	81.07	89.10	89.08	93.25	93.25	52.42	51.91
Extra Tree	**99.6**	**99.6**	**99.57**	**99.58**	92.20	92.16	73.52	73.46	x	x	x	x

Figure 15: SB2 model performace.

6.2.3 SB1 merged features model performance

The experimental results reveal several noteworthy insights into model performance across different feature sets. Among the models tested, the Extra Trees Classifier consistently achieved the highest accuracy and F1-score across multiple scenarios. For instance, with the MFCC + Chroma feature set, it reached 99.17 accuracy and 99.17 F1-score on the test set, and 93.69 accuracy and F1-score on the validation set, showcasing its robustness in capturing complex patterns. Similarly, the inclusion of STFT features improved its performance, achieving 92.74 accuracies and 92.74 F1-score on the test set when using MFCC + Chroma + STFT. This indicates that the Extra Trees Classifier effectively leveraged the enriched feature space, making it the most reliable model overall.

In addition to Extra Trees, other tree-based models like the Random Forest also performed competitively, especially with the MFCC + STFT feature set, yielding 97.27 accuracy and 97.20 F1-score on the test set. This suggests that tree-based models effectively capture non-linear relationships and interactions between features. Ensemble methods such as Soft Voting, Hard Voting, and Weighted Voting further enhanced performance by combining predictions from multiple models. Notably, Weighted Voting achieved an impressive 97.33 accuracy and 97.12 F1-score on the validation set for MFCC + STFT, demonstrating the benefits of aggregating diverse classifiers.

Conversely, simpler models like K-Nearest Neighbors (KNN) and Naive Bayes (NB) struggled with the higher-dimensional feature spaces. KNN performed poorly, particularly on the MFCC + Chroma feature set, with only 49.87 accuracy and 57.04 F1-score on the test set, likely due to its sensitivity to irrelevant features and the curse of dimensionality. Naive Bayes showed moderate performance, benefitting slightly from the inclusion of STFT, though it remained outperformed by more complex models. Meanwhile, Support Vector Machines (SVM) exhibited noticeable improvement with the addition of STFT features, reaching 94.40 accuracy and 99.01 F1-score on the MFCC + STFT test set.

Overall, the results highlight the importance of feature engineering and model selection in achieving optimal performance. The integration of STFT features improved the predictive performance across most models, indicating that frequency-domain information captured by STFT complements the time-domain features from MFCC and Chroma. Tree-based models, especially Extra Trees and Random Forest stood out as the most effective approaches for handling the combined feature sets, while ensemble methods demonstrated their capability in enhancing performance through collaborative learning. In contrast, simpler models like KNN and Naive Bayes faced challenges with the high-dimensional data, further emphasizing the need for more sophisticated techniques when dealing with complex datasets. All results shown in figure 16.

SB1 merge												
	MFCC + Chroma				MFCC + STFT				MFCC + Chroma + STFT			
	Test		Validation		Test		Validation		Test		Validation	
Model	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1
KNN	49.87	57.04	49.31	36.57	79.01	78.83	94.44	92.12	93.05	92.99	78.39	78.12
SVM	67.01	75.35	75	80	99.40	99.01	85.50	85.28	88.77	85.56	98.86	97.92
NB	92.11	92.61	94.86	95	89.33	87.13	67.25	63.24	**98.61**	**97.85**	90.33	87.13
RF	**98.66**	**98.67**	**89.72**	**88.82**	97.27	97.20	95.36	95.12	97.83	97.22	95.56	94.22
Soft Voting	98.43	98.43	96.67	96.67	98.93	98.92	97.31	97.22	97.25	96.23	**96.78**	**95.78**
Hard Voting	99.75	99.8	96.56	96.58	**99.17**	**99.02**	95.72	95.22	98.60	98.11	95.78	94.12
Weighted Voting	91.9	91.9	94	94	98.84	98.56	**97.33**	**97.25**	96.59	92.31	97.33	97.12
Stacking	91.88	92	93.64	93.62	92.05	91.73	96.42	96.53	82.50	79.15	96.81	93.11
Extra Tree	**99.17**	**99.17**	**93.69**	**93.69**	99.13	98.93	91.31	97.95	93.53	**92.74**	92.83	92.34

Figure 16: SB1 merged feature model performance.

6.2.4 SB2 merged features model performance

The table presents the performance of various models across three feature combinations: MFCC + Chroma, MFCC + STFT, and MFCC + Chroma + STFT. The evaluation is conducted on both the test and validation sets, using Accuracy and F1 scores as performance metrics.

In the MFCC + Chroma combination, the models demonstrate consistently high performance. Notably, the Random Forest (RF) model achieves the highest scores, with an accuracy of 98.98% and an F1 score of 98.99% on the test set, while also maintaining exceptional results on the validation set (99.33% and 99.34%). Furthermore, the Hard Voting model shows a remarkable F1 score of 99.6% on the test set, indicating strong agreement among classifiers when combined through voting.

For the MFCC + STFT combination, the performance slightly declines compared to the previous feature set. The Hard Voting model exhibits the best test set performance, reaching an accuracy of 98.60% and an F1 score of 97.23%. Additionally, RF and Soft Voting achieve the highest accuracy in the validation set at 98.66%, reflecting their stability across different data splits.

In contrast, the MFCC + Chroma + STFT combination leads to a noticeable drop in performance across most models. The accuracy and F1 scores decrease considerably, particularly on the test set, suggesting that the inclusion of all three feature types may introduce redundancy or noise, potentially hindering model performance. However, the Weighted Voting model achieves the highest validation performance with both accuracy and F1 scores reaching 97.31%, indicating that ensemble methods remain effective in mitigating some of the negative impacts of feature expansion. All results show in Figure 17.

SB2 merge												
	MFCC + Chroma				MFCC + STFT				MFCC + Chroma + STFT			
	Test		Validation		Test		Validation		Test		Validation	
Model	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1
KNN	96.52	96.64	89.64	90.70	90.92	90.12	95.21	93.15	67.32	63.28	89.63	89.52
SVM	98.54	98.56	90.45	91.37	91.59	91.21	90.37	89.12	98.86	98.86	88.77	88.63
NB	96.52	96.64	97.81	97.91	97.91	95.77	90.31	87.14	78.39	77.48	93.05	93.04
RF	**98.98**	**98.99**	**99.33**	**99.34**	92.69	91.74	**98.66**	**97.23**	**98.66**	**98.66**	92.69	92.67
Soft Voting	98.99	98.99	96.96	96.96	96.78	94.56	97.25	97.11	96.78	96.77	97.25	97.25
Hard Voting	99.53	99.6	98.4	98.4	**98.60**	**97.23**	95.78	94.69	98.60	98.59	95.78	95.77
Weighted Voting	98.88	98.89	96.87	96.76	96.59	93.23	97.33	97.32	96.59	96.59	**97.31**	**97.31**
Stacking	98.88	98.9	96.9	96.9	93.53	91.34	92.83	92.32	96.61	96.61	92.83	92.81
Extra Tree	99.66	99.66	98.92	98.9	97.67	97.43	92.25	91.76	97.67	97.66	92.25	92.22

Figure 17: SB1 merged feature model performance.

7 Future work

Currently, the models achieve high accuracy ranging from 97% to 99%. However, there is still potential for further improvement to enhance the generalization capability of the models. Additional feature extraction techniques such as Constant-Q Transform (CQT), Hilbert Transform, and Fast Fourier Transform (FFT),... could be explored to uncover more informative representations and potentially improve model performance.

Moreover, the application of deep learning techniques may offer further insights and optimization opportunities for this problem. They often demonstrate superior generalization capabilities, contributing to more stable and robust performance across diverse scenarios.

Additionally, exploring various models and fine-tuning hyperparameters could further enhance performance, allowing the discovery of more optimal configurations and improving the overall results.

8 References

Phan, T.T.H., Nguyen, H.D., Doan, D.N.: Evaluation of feature extraction methods for bee audio classification. ICIT 2022

Tymoteusz Cejrowski, Julian Szymański, Higinio Mora, and David Gil. 2018. Detection of the Bee Queen Presence Using Sound Analysis. In Intelligent Information and Database Systems (Lecture Notes in Computer Science), Ngoc Thanh Nguyen, Duong Hung Hoang, Tzung-Pei Hong, Hoang Pham, and Bogdan Trawiński (Eds.). Springer International Publishing, Cham, 297–306.

Britanak, Vladimir; Yip, Patrick C.; Rao, K. R. (6 November 2006). Discrete Cosine and Sine Transforms: General Properties, Fast Algorithms and Integer Approximations.

Alfred DeMaris. 1995. A tutorial in logistic regression. Journal of Marriage and the Family (1995), 956–968.

Altman, N.S.: An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician 46(3), 175–185 (1992)

Boser, B.E., Guyon, I.M., Vapnik, V.N.: Support vector machines. Machine Learning 20(3), 273–297 (1992)

Cronbach, L. J. (1954). "On the non-rational application of information measures in psychology". In Quastler, Henry (ed.). Information Theory in Psychology: Problems and Methods. Glencoe, Illinois: Free Press. pp. 14–30.

T. Cover and P. Hart. 1967. Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13, 1 (1967), 21–27.

V. Tyagi and C. Wellekens (2005), On desensitizing the Mel-Cepstrum to spurious spectral components for Robust Speech Recognition, in Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on, vol. 1, pp. 529–532.

Müller, M., & Ewert, S. (2011). Chroma Feature Analysis and Synthesis. IEEE Transactions on Audio, Speech, and Language Processing, 19(7), 1532-1545.

Sahu, P. K.,& Raj, R. (2019). Speech Based Emotion Recognition. International Journal of Advances in Scientific Research and Engineering, 5(8), 34-40.

Kumar, V., & Bhatia, P. (2022). A Comparative Analysis of Machine Learning Algorithms for Classification. Procedia Computer Science, 187, 110-115

Prasetyo, E., & Setiawan, A. (2023). Performance Comparison of SVM, Naive Bayes, and Random Forest for Fake News Detection. Emerging Technologies Journal, 7(1), 45-50.

Zhang, Y., & Wang, S. (2024). An Ensemble Feature Selection Approach-Based Machine Learning System for COVID-19 Diagnosis. Computational Intelligence and Neuroscience, 2024, Article ID 8188904.

Ganaie, M. A., & Hu, M. (2023). A Comprehensive Review on Ensemble Deep Learning: Opportunities and Challenges. Journal of King Saud University - Computer and Information Sciences, 35(1), 1-16

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. Machine Learning, 63(1), 3–42. https://doi.org/10.1007/s10994-006-6226-1

Wolpert, D. H. (1992). Stacked generalization. Neural Networks, 5(2), 241–259. https://doi.org/10.1016/S0893-6080(05)80023-1

Kuncheva, L. I. (2004). Combining pattern classifiers: Methods and algorithms. Wiley-Interscience.

Dietterich, T. G. (2000). Ensemble methods in machine learning. Proceedings of the First International Workshop on Multiple Classifier Systems, 1–15.

Kuncheva, L. I. (2004). Combining pattern classifiers: Methods and algorithms. Wiley-Interscience.

Dietterich, T. G. (2000). Ensemble methods in machine learning. Proceedings of the First International Workshop on Multiple Classifier Systems, 1–15.

Lewis, D. (1998). "Naive Bayes at Forty: The Independence Assumption in Information Retrieval." In Machine Learning: ECML-98, Proceedings of the 10th European Conference on Machine Learning, Chemnitz, Germany (pp. 4–15). Berlin: Springer

Breiman, L. (2001). "Random Forests." Machine Learning, 45(1), 5-32.

Louppe, G., Wehenkel, L., Sutera, A., & Geurts, P. (2013). Understanding variable importances in forests of randomized trees. Advances in Neural Information Processing Systems (NeurIPS), 26.

Marquardt, D. W. (1970). Generalized inverses, ridge regression, biased linear estimation, and nonlinear estimation. Technometrics, 12(3), 591-612.

O'Brien, R. M. (2007). A caution regarding rules of thumb for variance inflation factors. Quality & Quantity, 41(5), 673-690.

Shannon, C. E. (1948). A mathematical theory of communication. Bell System Technical Journal, 27(3), 379-423.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. Philosophical Magazine, 2(11), 559-572

Sahidullah, Md., & Saha, G. (2012). Feature Extraction of Speech Signal for Robust Speech Recognition System. Retrieved from arXiv:1303.3614.

Rabiner, L. R., & Schafer, R. W. (2010). Theory and Applications of Digital Speech Processing. Pearson.