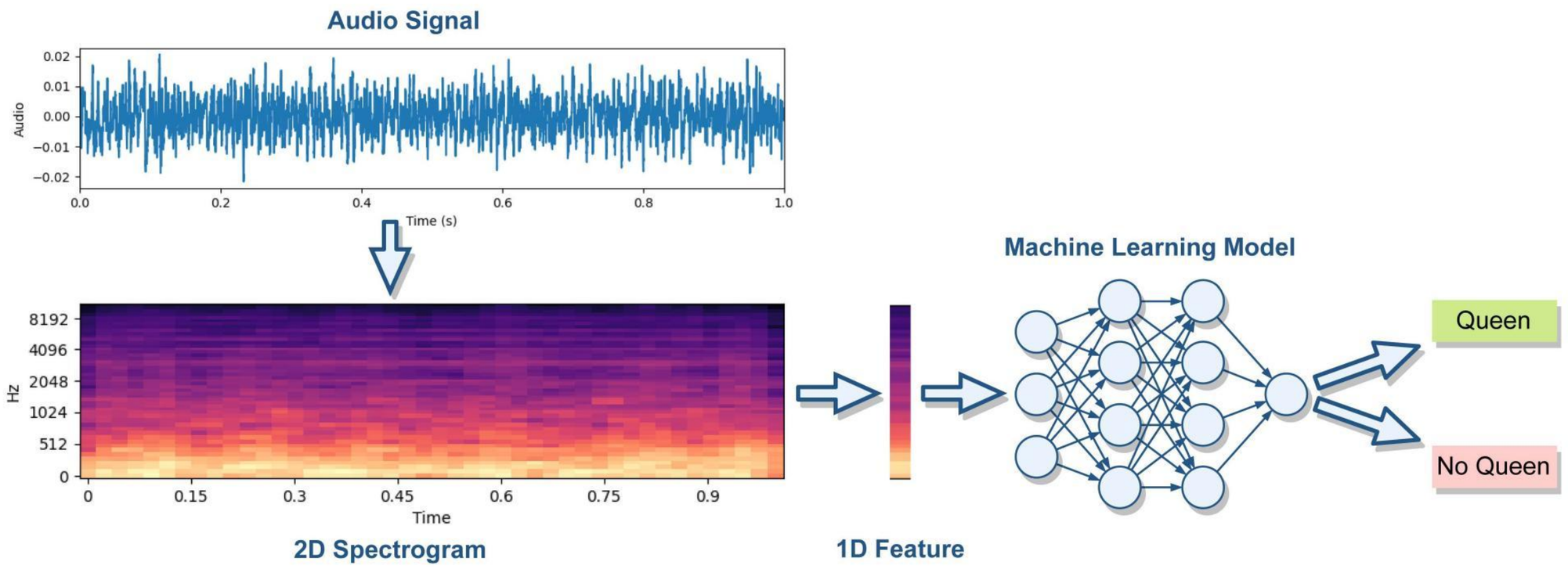

Fig. 2. Step for extracting the MFCC from the chunk audio file, and then compressed into a single dimension vector computing the average value. Finally the machine learning model will use them to predict the presence of the queen bee.

TABLE I
DATASET AUGMENTED SIZES WITH HOP SIZE OF 5 S AND DIFFERENT CHUNK LENGTHS

Chunk size	Dataset 1 [4]	Dataset 2 [24]	total
0.5 s	85 200	67 524	152 724
1 s	85 200	67 497	152 697
3 s	85 200	67 342	152 542
5 s	78 100	67 049	145 149

lasting approximately 10 min. In total, there are 576 audio files, half labeled as "Missing Queen" and the other half as "Normal Beehive." These recordings represent the activities of the two hives over two distinct days.

B. Audio Chunk Split

Despite utilizing two datasets with a substantial amount of data, this study incorporates a data augmentation technique to expand the number of input samples. This involves breaking down audio files into smaller segments. Adjusting the hop size, which represents the space between the start of consecutive chunks within the original audio, it is possible to increase the number of chunks. This technique is described in [31] with the name "time slicing window" and can be found in other works [32] with similar names like "TimeShiftRange" and "RandXTranslation." We subsequently evaluate the impact on the results by exploring various chunk sizes: 0.5, 1, 3, and 5 s, while maintaining the same number of chunks and avoiding overlap. To attain this objective, a hop size of 5 s was utilized, leading to varying numbers of samples as detailed in Table I. The inconsistency in the count of resulting segments arises from the consideration of only complete chunks, with no employment of padding techniques to ensure uniform length across all audio files.

C. Feature Extraction

Using raw audio data for sound classification is not convenient due to its high dimensionality, leading to computational challenges and increased processing costs. Raw waveforms may not directly capture relevant features for classification, requiring sophisticated feature extraction techniques. In addition, raw audio is susceptible to environmental noise, making models sensitive and less robust. To address these issues, different preprocessing techniques are commonly employed to enhance model efficiency and performance by providing more compact and informative representations of the audio signals. In this study are compared two types of features: mel-frequency cepstral coefficients (MFCC) and the STFT spectrograms. These preprocessing techniques are used to produce the input vector used by the machine learning model as shown in Fig. 2.

Mel-Frequency cepstral coefficients: The first method is a widely-used technique for acoustic-applications [33], in particular, to capture relevant characteristics of the human auditory system by transforming the audio signal into a compact representation with a user-defined number, denoted as n, of coefficients. Results were compared across varying n values from 10 to 50. By extracting a relatively small number of coefficients, MFCCs reduce the dimensionality of the feature space while retaining essential information about the audio signal. This is crucial for efficient processing and classification. The MFCC extraction is done on each audio chunk of the dataset, this process is explained by the following sequence of steps necessary to obtain the MFCC coefficients vector.

1) Division of the audio into windows of 2048 audio samples with partial overlap, using a hop length of 512 samples.
2) Application of the Hann windowing function to smooth the signal at the window edges.
3) Use of discrete Fourier transform (DFT) to convert the signal to the frequency domain.
4) Application of the mel filterbank set of triangular filters, evenly spaced on the Mel scale.
5) Application of discrete cosine transform to obtain coefficients for each window.
6) Computation of the mean value for each coefficient across all windows, resulting in the n features that will be used as input for the classifiers.

Short Time Fourier Transform: The second method uses only the Fourier transform as explained in [34], to obtain a reduced