electronics

MDPI

Article

Identifying Queenlessness in Honeybee Hives from Audio Signals Using Machine Learning

Stenford Ruvinga [1,*], Gordon Hunter [1,*], Olga Duran [2] and Jean-Christophe Nebel [1]

[1] School of Computer Science and Mathematics, Kingston University, London KT1 2EE, UK
[2] School of Engineering and the Environment, Kingston University, London SW15 3DW, UK
* Correspondence: k1103714@kingston.ac.uk (S.R.); g.hunter@kingston.ac.uk (G.H.)

Abstract: Honeybees are vital to both the agricultural industry and the wider ecological system, most importantly for their role as major pollinators of flowering plants, many of which are food crops. Honeybee colonies are dependent on having a healthy queen for their long-term survival since the queen bee is the only reproductive female in the colony. Thus, as the death or loss of the queen is of great negative impact for the well-being of a honeybee colony, beekeepers need to be aware if a queen has died in any of their hives so that appropriate remedial action can be taken. In this paper, we describe our approaches to using acoustic signals recorded in beehives and machine learning algorithms to identify whether beehives do or do not contain a healthy queen. Our results are extremely positive and should help beekeepers decide whether intervention is needed to preserve the colony in each of their hives.

Keywords: honeybees; queen bee; bee colony; audio signal; CNN; LSTM; MLP; logistic regression; FFT; MFCC; spectrograms

Citation: Ruvinga, S.; Hunter, G.; Duran, O.; Nebel, J.-C. Identifying Queenlessness in Honeybee Hives from Audio Signals Using Machine Learning. *Electronics* **2023**, *12*, 1627. https://doi.org/10.3390/electronics12071627

Academic Editors: Gwanggil Jeon and Chunjie Zhang

Received: 27 January 2023
Revised: 9 March 2023
Accepted: 21 March 2023
Published: 30 March 2023

1. Introduction

Recent advances in sensor technology and algorithms for the processing and interpretations of the signals they capture have now made "smart" monitoring applied to various environmental problems possible, with a view to resolving or mitigating them.

Among such problems, of major concern is the serious decline in populations of pollinating insects. These are vital to the survival of many flowering plant species, including many crops for both human and animal consumption, notably fruit. Hence, the well-being of such pollinators is essential for both the agricultural industry and the wider environment. Foremost among such insects are honeybees, which are possibly the most productive of pollinating insects. Over recent years, honeybee numbers in much of Europe and North America have been dwindling. This has been attributed to a wide range of factors—from pesticides such as neonicotinoids [1] to the rise of largely monoculture agriculture in many areas [2], and even the microwave electromagnetic radiation used in mobile telecommunications [3].

Monitoring the well-being of beehives offers a way of ensuring individual colonies do not fall into terminal decline. If major problems are detected sufficiently promptly, remedial action can be taken, ensuring that the colony does not die off. However, monitoring should ideally be carried out in a manner that is as non-invasive as possible in order to cause the minimum possible disruption to the natural life cycle of the bees, minimizing stress and disturbance to them and their productivity as pollinators. Monitoring the sounds produced by the bees is an appropriate modality for this as the sounds can be used as an indicator of important events occurring or being impending within the colony. Both anecdotal accounts from beekeepers [4] and previous scientific studies [5] have indicated that the spectral profiles of the sounds produced by honeybees change if their colony lacks a healthy queen [6,7], or if the colony is about to swarm [8,9]. These are both major issues of concern to beekeepers. In the former case, since the queen is the only reproductive female

in the colony, loss of the queen results in the population of the colony steadily declining, eventually resulting in the colony dying out unless the queen is replaced. In the latter case, swarming leads to a high proportion of the bees in a hive leaving suddenly, so an accurate prediction of when this is going to occur would enable the beekeeper to plan appropriate action to prevent the loss of those bees.

In this paper, we describe our approaches to addressing the first of these problems, making use of acoustic data recorded from beehives, where some were known to have had the queen removed and others where the presence of the queen was known. Some of the preliminary results from some of the methods presented in this paper were previously published in Proceedings of the 17th IEEE International Conference on Intelligent Environments [10]. However, this present paper includes additional approaches not considered in [10], notably the use of Convolutional Neural Network to analyse sound spectrograms, and the validation of the model by applying it to a second dataset, acquired in a different location and over a different time period. The main contribution of this paper is, to the best of our knowledge, the first applications (other than [10]) of advanced Machine Learning approaches to detect the absence of a queen in a hive purely from analysis of the audio signals. As excellent classification (as Queen Present or Queen Absent) performance is obtained on available data, this work could prove highly valuable to beekeepers as a means of promptly alerting them if the queen had died in any of their hives.

2. Related Previous Work

The importance of honeybees and keeping them healthy has been noted in Section 1 above. Furthermore, beekeepers want to maintain stable or increasing populations of healthy bees, so monitoring them regularly is essential. However, excessive human inspection of hives and their bees causes stress and disruption to the bee colonies and can even spread diseases and parasites. Many different approaches have been proposed and prototyped to automate the monitoring of bees and their behaviour. However, each of these is associated with both advantages and disadvantages [11].

Video monitoring is somewhat impractical inside the hive. Indeed, usage of visible light would require a source of illumination situated inside the beehive much of the time, which would cause disturbance to the bees' natural daily cycle. Instead, some authors [12,13] have used infra-red imaging inside the hive. This approach allows estimation of the hive's population of bees, but not necessarily accurate prediction of swarming or queen loss. Moreover, this approach is sensitive to external temperature and sunlight levels, and cooling effects of wind. Alternatively, video monitoring of bees in the human-visible parts of the spectrum can be used outside a hive—particularly close to its entrance [14–16]. Whilst this can be useful for monitoring levels of bee activity and potentially identify attacks by predators such as wasps or hornets, it tends to require relatively expensive equipment and generates very large volumes of data per day which requires a lot of bandwidth and/or memory storage capacity.

Acoustic monitoring of bees in hives has been carried out over many years [4,17] and has been used to detect when a swarm has occurred. Most such studies did not employ any Artificial Intelligence or Machine Learning approaches, relying instead on manual inspection of the spectrograms and the hive temperature signal, when available. Ferrari et al. [9] monitored the sounds produced by the bees in a hive, along with its temperature and humidity, in the period around a swarm occurring. They performed spectrographic analysis of the sounds, noting that the dominant frequency bands in terms of signal power moved from 100 to 300 Hz at times not including a swarm to 500–600 Hz immediately before a swarm. This increase was attributed to intensive flitting of the bees' wings in preparation for flight. Eren et al. [18] also used analysis of sound signals, focusing on detecting the presence of a queen bee in a hive. They determined that notable power was observable in the sound signals produced by worker bees up to around 11 kHz, with peaks in the spectra between 200 and 270 Hz and between 400 and 550 Hz, with the variability occurring due to the precise conditions under which each recording was made. In contrast,

they obtained no evidence of queen bees producing sounds at frequencies lower than 400 Hz, with the spectra of queen sounds showing one major peak between 400 and 550 Hz, plus higher power in the bands between 500 and 5000 Hz. They exploited their findings to attempt to control the behaviour of bees by playing them artificially generated sounds at specific frequencies. However, their results proved rather inconclusive. Finally, Žgank [19] may have been the first to use Machine Learning to monitor and classify sounds from bees to detect swarming activity by applying techniques commonly used to analyse human speech, i.e., Mel Frequency Cepstral Coefficients (MFCCs) and Hidden Markov Models.

Regarding the problem of detection of the presence of the Queen, Boys [4] reported that the "warble" sound produced by workers at between 225 and 285 Hz would decrease in amplitude, largely replaced by a "moaning" sound at a rather lower frequency if the queen was inactive or had died. However, more sophisticated studies have been performed more recently which take advantage of Machine Learning. Howard et al. [6] employed the Stockwell Transform for frequency analysis of Queen Present and Queen Absent beehives, using a Self-Organising Map and Power Spectral Density to compare the sound signals from the two different types of hives. Although their work did show some different patterns in the sounds from the Queenless hives relative to the Queen Present ones, their results were somewhat inconclusive, and they suggested that usage of MFCCs might be more appropriate than the less commonly used Stockwell transform. Indeed, good success rates were obtained by Robles-Guerrero et al. [20] by using MFCCs as inputs to Lasso logistic regression and Singular Value Decomposition for classification of sounds made by Queen Present and Queenless beehives. Also relying on MFCCs as features, Peng et al. [21] employed a Multi-Layer Perceptron (MLP) artificial neural network classifier with two hidden layers and a hyperbolic tangent activation function to identify whether or not beehives were queenless. Although they obtained accuracies close to 90% with "clean" audio signals, the success rate declined to below 61% if the signals were corrupted by substantial amounts of noise. However, their paper did not disclose any details of their dataset or ways it was acquired.

In this present paper, using both Fourier (STFT) and Mel (MFCC) audio features, we apply more sophisticated machine learning approaches (i.e., LSTM and CNN) to address the hive queenlessness problem. For comparison to simpler alternative approaches, we also made use of Logistic Regression and a Multi-Layer Perceptron neural network applied to the same datasets.

3. Datasets and Methods

3.1. Data

Two honeybee audio datasets were primarily used in this work, one for training and validation, the other for testing. The first dataset was provided by Arnia Ltd. (www.arnia.co.uk (accessed on 27 January 2023)) and consists of one-minute duration Waveform (.wav) format audio files sampled hourly, recorded from the 3 to the 9 of August 2012 [10], at a sampling rate of 44.1 kHz. It comprises recordings from both Queen Present (QP) and Queen Absent (QA) hives—four separate hives of the same size, in the same location. For our experiments, only recordings from the 5 to the 9 of August were used to ensure equal numbers of samples for both hive states. Thus, there were 120 one-minute duration recordings for each hive giving a total of 480 recordings. Two of the hives had bees of the Italian sub-species Apis mellifera ligustica and the other two had the Slovenian honeybee sub-species Apis mellifera carnica. One hive of each species which had a queen throughout was used as control, whilst the other two hives, one from each species, had their queens removed at about midday on the 4th of August 2012, becoming queenless at that point. For details of the days of recording and the sub-species in each hive, see Table 1. The second dataset was recorded during the summer of 2020 in rural Surrey, UK during July, and August 2020. It consists of two batches: one with 398 and the other with 450 one-minute waveform samples recorded at a sampling rate of 20.38 kHz. Both batches had the queen

present in the hives used, which contained "Buckfast" hybrid honeybees, Apis mellifera buckfast.

Table 1. The status of each of the four hives for the 7 days of recording of the first (Arnia Ltd.) dataset. C stands for control group, QA for Queen Absent and QP for Queen Present. The QP/QA entries indicates that the queen was removed from that hive around midday that day. Aug stands for August.

Recording Date Apis Mellifera Sub-Species	3 August 2012	4 August 2012	5 August 2012	6 August 2012	7 August 2012	8 August 2012	9 August 2012
Ligustica	QP	QP/QA	QA	QA	QA	QA	QA
Ligustica ©	QP	QP	QP	QP	QP	QP	QP
Carnica	QP	QP/QA	QA	QA	QA	QA	QA
Carni ©(C)	QP	QP	QP	QP	QP	QP	QP

3.2. Methodologies

In this work, we propose the use of Mel Frequency Cepstral Coefficients (MFCCs) and spectrograms as input features in a Long Short-Term Memory (LSTM) classifier [22] and a Convolutional Neural Network (CNN) classifier [23], respectively, to classify bee audio signals from hives with a queen present and those without a queen. To evaluate the performance of our solutions, we compare the results to those of two standard classifiers, Logistic Regression, and a Multi-Layer Perceptron (MLP) neural network trained using MFCCs.

3.2.1. Features

(i) Mel Frequency Cepstral Coefficients

MFCCs are a compact representation of audio signals [24] motivated by observations in human speech recognition [25,26]. They have been successfully implemented and applied to various audio related fields, among them environmental sound classification [27], music genre classification [28], heart sound recognition [29] and automatic speaker recognition [26]. They are extensively used due to their classification and identification effectiveness compared to other audio features as they have a good representation of the continually relevant aspects of the short-term audio spectrum [30]. Another important strength of MFCC features is that they are uncorrelated, which results in them containing less redundant information than alternative approaches. In most applications, only the first few MFCC features, 13 in this present work, are used as they represent most of the signal's important spectral information. A summary of the 7-step MFCC feature extraction process is given in Figure 1. A more detailed description of the steps involved in this process follows below.

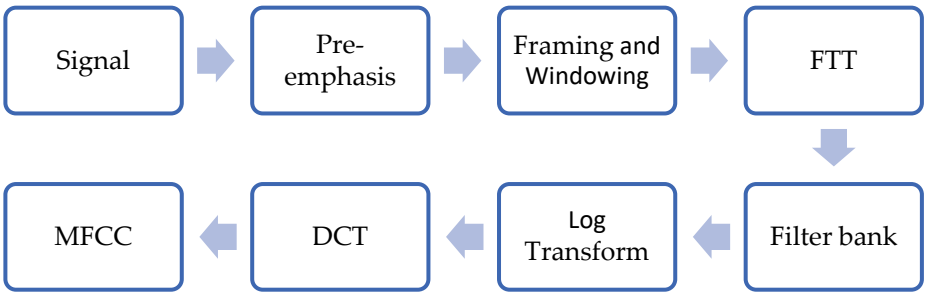


Figure 1. The MFCC feature extraction process, where FFT is the Fast Fourier Transform, and DCT is Discrete Cosine Transform.

Pre-Emphasis Filtering: A preliminary investigation of our data indicated that useful information for improving classifier performance was present in relatively high frequency bands (3000–6000 Hz) which tend to be suppressed during the signal recording and compression process. Thus, the signal is filtered to recover energy in the high frequency bands

that was previously compressed during the audio digital recording process. One of the widely used filter is defined as

$$H_{pre}(z) = 1 + a_{pre}z^{-1} \tag{1}$$

where z is the Z-transform variable with $z = e^{i\omega T}$, $T = 1/fs$ is the time between successive samples, ω the angular frequency and $i^2 = -1$, and where $-1 \leq a_{pre} \leq -0.4$ is a constant that controls the filter slope [31].

Framing and Windowing: Natural audio signals are typically non-stationary. To mitigate for that, the signal is framed, i.e., partitioned into overlapping segments, called frames, of equal length of about 20–40 ms in duration [32]. This reduces the effects of non-stationarity while retaining most of the spectral information [31].

Windowing is then applied to individual consecutive frames to prevent discontinuity of the signals generated by the framing process. This involves multiplying each frame by the window function (see Figure 2), where the Hamming and the Hanning windows are commonly used. These two window functions are defined as $w[n]$:

$$w[n] = (1 - \omega) - \omega * \cos\left(\frac{2\pi n}{L - 1}\right) \tag{2}$$

for $0 \leq n \leq L - 1$, where L is the window width, $\omega = 0.5$ for the Hanning window and $\omega = 0.46164$ for the Hamming window [31].

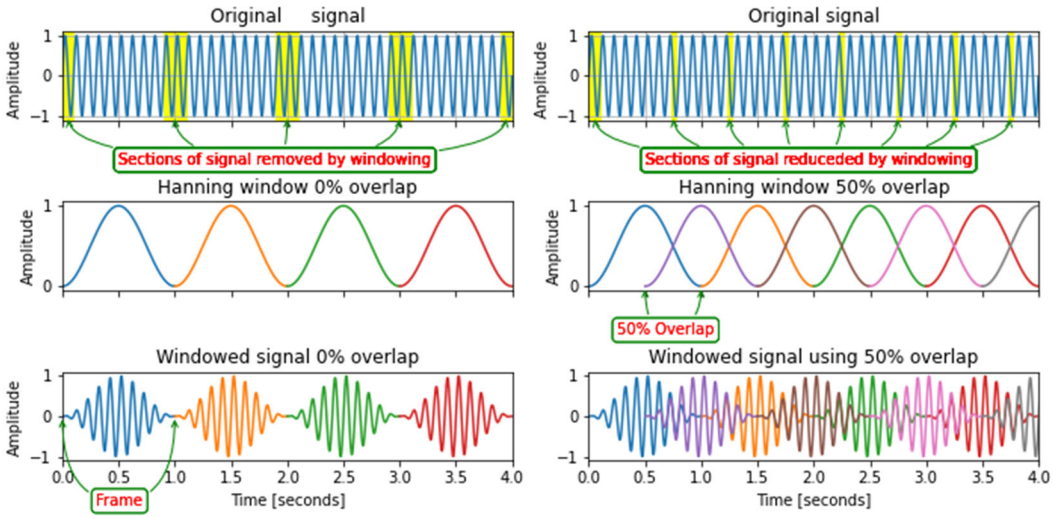


Figure 2. Illustration of the framing and windowing process using a Hanning window: (**a**) with 0% overlap; (**b**) with 50% overlap.

Since a Hanning or Hamming window is being used, a 50% overlap is ideal as it ensures that no part of the signal contributes more to the spectrogram than others. Indeed, in these windows, the middle part is weighted with 100%, but the beginning and end are weighted 0%. However, within a sequence of 50% overlapping windows of the same type, the sum of the weightings is equal to 1 exactly, except for the first half and last half windows of the sequence, meaning that every part of the signal contributes equally due to this symmetric property.

Fast Fourier Transform (FFT): The FFT is an efficient algorithm that computes the Discrete Fourier Transform (DFT) of a signal, converting the framed and windowed signal from the time domain to the frequency domain. For a signal x[n] = x (t = n/f_s) sampled at

regular discrete time periods, f_s is the sampling frequency and n = 1, 2, 3, . . . , $N - 1$. The DFT, $X[k]$, of the signal x[n] is defined as

$$X[k] = \sum_{n=0}^{N-1} \mathrm{x}[n] e^{-\frac{2\pi i k n}{N}}, \tag{3}$$

for $0 \leq k \leq N - 1$, $i^2 = -1$ and N = total number of samples under consideration.

Mel Filter Bank: In Mel Filter banks, the spectrum from the FFT is warped along its frequency axis f (in Hz) into the Mel-scale, created to model human perception of sound, using triangular overlapping windows [30] using the formula

$$f_{mel} = 2595 \log_{10} (1 + \frac{f}{700}) \tag{4}$$

where f denotes the physical frequency in Hz, and f_{mel} denotes the perceived frequency [31]. The resultant Mel frequency components are then filtered using Formula (5) below.

$$Y[m] = \sum_{k=1}^{N} \mathrm{W}_m[k] |X[k]|^2 \tag{5}$$

for $0 \leq k \leq N$, $0 \leq k \leq M$, where k is the DFT bin number, m is the Mel-filter bank number, and the $\mathrm{W}_m[k]$ are weighting functions.

Logarithmic Transformation: The Mel frequency components are then transformed using a logarithm to reduce the dynamic range [31].

Discrete Cosine Transform (DCT): Finally, the DCT is taken on the logarithmic outputs from the above. This results in decorrelated MFCC features. for $0 \leq n \leq \mathrm{C} - 1$ and $0 \leq m \leq M - 1$, where the $c(n)$ are the cepstral coefficients, C is the number of MFCCs being considered and M the number of filter banks.

$$c(n) = \sum_{m=0}^{M-1} \log_{10} (Y(m)) \cos(\frac{\pi n (m - 0.5)}{M}) \tag{6}$$

(ii) Spectrograms

A spectrogram is a time–frequency transformation which takes a one-dimensional sequence x[n] and converts it into a two-dimensional function of discrete time and discrete frequency. It is obtained by applying a Short-Time Fourier transform (STFT) to the signal [33]. The STFT is basically a DFT applied to equal, usually overlapping, portions of a finite length signal. For a signal x[n] and window $w[n]$ of length N samples, the STFT is defined as

$$X[m, k] = \sum_{n=0}^{N-1} \mathrm{x}[n] w[n - m] e^{-\frac{2\pi i k n}{N}} \tag{7}$$

The spectrogram, S, is then generated by computing the squared magnitude of the Short-Time Fourier Transform (STFT) of the signal x[n]:

$$S(m, k) = |\mathrm{X}[m, k]|^2 \tag{8}$$

Spectrograms are usually represented as 2D images with frequency on the y-axis and time on the x-axis with the colour or shading intensity representing the magnitude of X[m, k]. The main advantage of the STFT is that it retains both the time and frequency information of the signal, unlike the DFT which only retains the frequency information of a signal. The window length N is chosen in such a way that the spectrogram will capture the relevant frequency information whilst having good temporal resolution.

3.2.2. Classifiers

Figure 3 gives an overview of the classification approaches used in this paper.

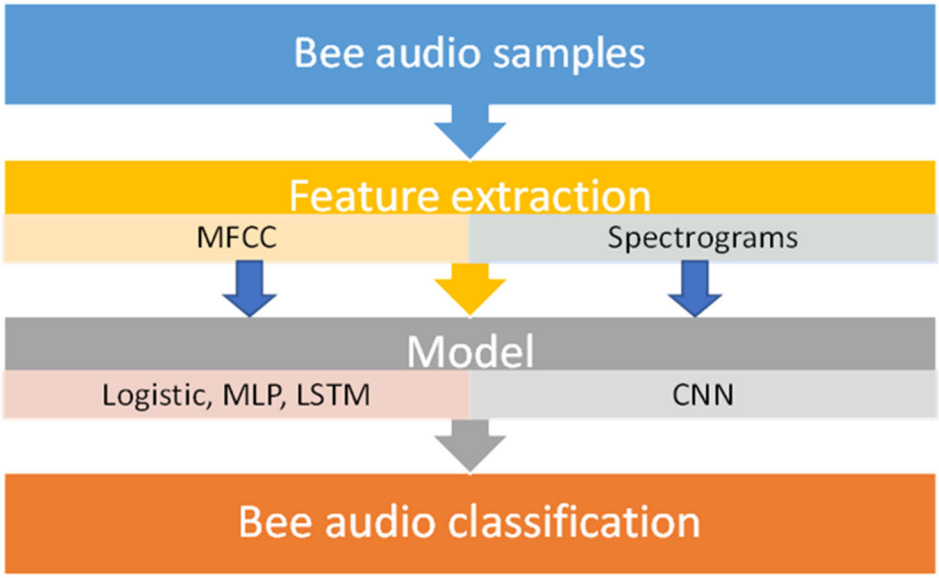


Figure 3. An overview of the classification pipeline.

(i) Multi-Layer Perceptron

A Multi-Layer Perceptron (MLP) network is a type of feedforward artificial neural network (ANN) [34]. Its architecture consists of at least three types of layers of nodes: an input layer, hidden layer(s), and an output layer; see Figure 4. Each layer consists of several nodes called neurons. An MLP network is usually fully connected, i.e., each node in one layer is connected to every node in the following layer, with weighted connections [35]. Neurons in the same layer do not share any connections. A sum of the weighted inputs is passed through an activation function in the neurons in the hidden and output layers [35]. The weights of such a network are usually trained using an algorithm such as error backpropagation.

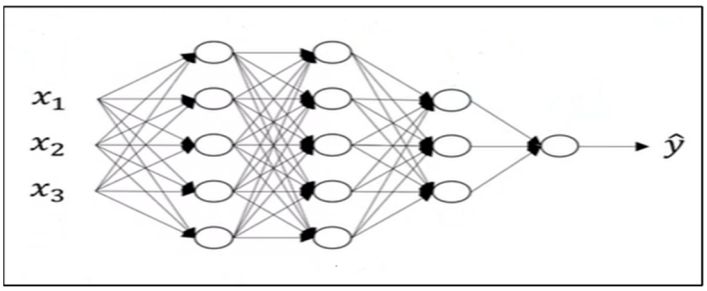


Figure 4. A Multi-Layer Perceptron network with 3 hidden layers. $\{x_1, x_2, x_3\}$ are the inputs and $\hat{y}$ is the single output.

MLPs use nonlinear activation functions, usually logistic (or sigmoid) and hyperbolic tangent, which enables them to solve problems that are not linearly separable. The two functions are defined respectively as

Logistic (sigmoid):

$$\sigma(x) = \frac{1}{1 + e^{-kx}} \tag{9}$$

Hyperbolic tangent:

$$\tanh x = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{10}$$

For this work, we employed the hyperbolic tangent activation function as it typically performs better than the logistic sigmoid [36]. MLPs utilize a supervised learning technique called error backpropagation for training. Training makes use of the delta rule method

which compares the output result with the target result and then adjusts the weights by some amount proportional to the error size [34]. The weights are updated after each forward pass. They are ideal for classification tasks as their output only depends on current input and not any past or future inputs. MLPs are the conceptual foundation of convolutional neural networks.

(ii) Logistic Regression

Logistic regression can be viewed either as a conditional probabilistic model, with the logarithm of the "odds ratio" being modelled as a linear function of the predictor variables, or as a simple ANN with no hidden layers; see Figure 5. As an ANN, for binary classification tasks, the output layer uses a single unit with a logistic (or sigmoid) activation. A weighted sum of the input vector plus a bias vector, just as for an MLP, is then fed into the logistic function which outputs values in the range (0, 1). The neuron's output value is the estimated probability that the input vector $\boldsymbol{x}$ belongs to one of the classes, say, C1.

$$\mathrm{P}(C_1|\boldsymbol{x}) = \sigma(\mathbf{z}) = \hat{y} \tag{11}$$

where σ is the sigmoid function defined above and **z** is the activation vector of that neuron. The estimated probability that it belongs to the other class, say, C2, is then given by

$$\mathrm{p}(C_2|\boldsymbol{x}) = 1 - \hat{y} \tag{12}$$

combining (11) and (12) provides

$$p(y|\boldsymbol{x}) = \hat{y}^y(1-\hat{y})^{1-y} \tag{13}$$

since $y = 1$ if the class is C1 and $y = 0$ if the class is C2.

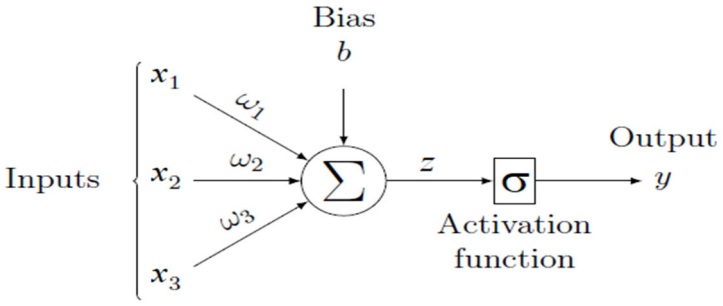


Figure 5. A Logistic Regression neural network [37], where $\boldsymbol{w} = (w_1, w_2, w_3)^T$ is a vector of weights, $\boldsymbol{x} = (x_1, x_2, x_3)^T$ is the input vector, b is the bias of the neuron, $z = \sum_i^3 w_i x_i + b$ is its activation, $\mathrm{y} = \sigma(\mathrm{z})$ is its output, and σ is the logistic (or sigmoid) function.

The training and updating of weights updating is as described in B2 (i) above. In this work, we employ binary logistic regression and thus we limit our discussion to that, although the above ideas can be extended to multinomial logistic regression for problems with more than two classes.

(iii) Long Short-Term Memory (LSTM)

An LSTM network is a type of recurrent neural network (RNN). Unlike traditional neural networks such as MLPs, RNNs have cyclic connections that enable them to retain and analyse sequential information. Another advantage of RNNs over traditional neural networks is that they share parameters at different time steps which help generalize to sequence lengths not seen during training, although this may make optimizing the parameters difficult [38]. A major drawback for RNNs is that they are not able to learn long-term dependencies as error signals flowing "backwards in time" tend to either blow up or vanish [22]. The LSTM network is a specialised RNN which overcomes this problem

by using an architecture which enforces constant error flow through each repeating cell [22]. The standard LSTM network has a repeating memory cell which has four interacting layers, a "forget gate" layer, an update/input gate layer, a hyperbolic tangent (tanh) layer, and an output layer, which control the flow of information [39]. Due to their ability to process sequence information, LSTM networks have been previously used in the analysis and modelling of time-varying sound signals [40].

In Figure 6, $\boldsymbol{y_t}$ is the y (outcome) estimate, $\boldsymbol{h_t}$ is the hidden state, $\boldsymbol{c_t}$ is the cell state, and $\boldsymbol{x_t}$ is the input sequence at time step t, σ is the sigmoid function, tanh is the hyperbolic tangent function, and $\boldsymbol{o_t}$, $\boldsymbol{i_t}$ and $\boldsymbol{f_t}$ are the output, input and forget gate results, respectively.

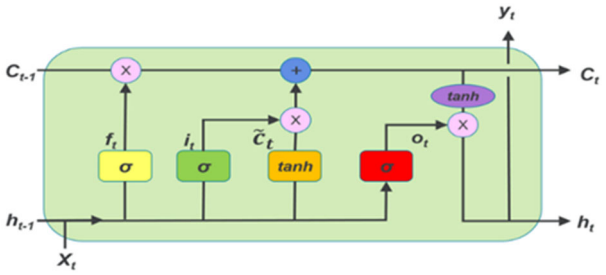


Figure 6. A repeating cell, which contains four interacting layers of an LSTM [39].

In what follows, W terms are the weight matrices with the indices indicating the output and input in that order, b terms are the bias vectors, σ is the logistic sigmoid function, and i, f, o and c are the input, forget gate, output gates and cell state vectors, respectively, x is the input vector, h is the cell output activation vector, $\odot$ is the Hadamard product of the vectors, g is a tanh activation function and ϕ is the output activation function, which is the sigmoid function in this paper [28], and each of the b terms is an appropriate constant bias term for that formula.

In the forget gate layer, a sigmoid function is used to determine the information to be thrown away from the cell state. The forget gate uses information from $\boldsymbol{h_{t-1}}$ (the previous time hidden state cell activation vector) and $\boldsymbol{x_t}$ (input vector at time t) to output a number between 0 (gate closed) and 1 (gate open) for each value in the cell state with a zero denoting "completely discard" and a 1 denoting "completely retain" by using the following formula:

$$\boldsymbol{f_t} = \sigma(W_{fx}\boldsymbol{x_t} + W_{fh}\boldsymbol{h_{t-1}} + b_f). \tag{14}$$

In the input gate layer, another sigmoid function is used to decide whether the values are to be updated. Like the forget gate, the input gate also uses information from $\boldsymbol{h_{t-1}}$ and $\boldsymbol{x_t}$ to output a number between 0 and 1. It uses the equation below to filter input information with values of zero denoting irrelevant information such as noise blocked from entering the cell and values of 1 denoting input deemed wholly relevant and thus are allowed passage into the cell.

$$\boldsymbol{i_t} = \sigma(W_{ix}\boldsymbol{x_t} + W_{ih}\boldsymbol{h_{t-1}} + b_i) \tag{15}$$

In a hyperbolic tangent layer, a vector of new possible values that could be added to the state is created using information from $\boldsymbol{h_{t-1}}$ and $\boldsymbol{x_t}$:

$$\tilde{\boldsymbol{c}}_t = g(W_{\tilde{c}x}\boldsymbol{x_t} + W_{\tilde{c}h}\boldsymbol{h_{t-1}} + b_{\tilde{c}}) \tag{16}$$

Lastly, the output layer uses a sigmoid function to determine the cell state output values.

$$\boldsymbol{o_t} = \sigma(W_{ox}\boldsymbol{x_t} + W_{oh}\boldsymbol{h_{t-1}} + b_o) \tag{17}$$

The cell state c_t, the hidden state h_t and the output y_t at time t are updated using the following formulae:

$$c_t = f_t \bigodot c_{t-1} + i_t \bigodot \tilde{c}_t \tag{18}$$

$$h_t = o_t \bigodot g(c_t) \tag{19}$$

$$y_t = \phi(W_{ya} h_t + b_y) \tag{20}$$

(iv) Convolutional Neural Networks

A convolutional neural network (CNN) is a type of artificial neural network widely used to analyse images. Its name derives from the fact that it uses a specialised mathematical operation called a convolution in place of general matrix multiplication used in traditional neural networks [23]. For a 2D image input I, a kernel K of size m × n, the convolution operation is defined as

$$M(i,j) = (I*K)(i,j) = \sum_m \sum_n I(n,m)K(i-m, j-n) \tag{21}$$

where $*$ represents the convolution operator.

The 2D output M is usually referred to as the feature map. In a CNN network, all the units in a feature map share the same weights and biases, and hence they detect the same features at all possible locations in the input. This reduces the number of free parameters a feature which makes them less prone to overfitting. Another important CNN feature is that they use sparse connections which makes them easier and faster to train compared to other networks of comparable size [37]. The CNN architecture has three main distinct types of layers: Convolutional, Pooling and a Fully Connected layers; see Figure 7.

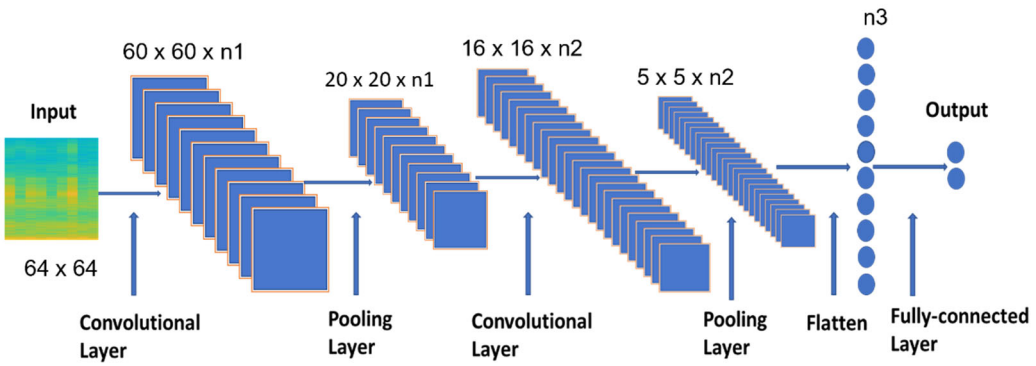


Figure 7. A basic CNN architecture with two Convolutional layers, two Pooling layers, a Fully connected layer, and an output layer with two possible outcomes, where $n1$ and $n2$ are the number of channels for layer 1 and layer 2 respectively, and $n3$ is the number of inputs to the Fully connected layer. (Adapted from [41].) The network takes a STFT spectrogram (as a 64 × 64 image) as input, and outputs one of two classes–Queen Present or Queen Absent.

A Convolutional Layer computes the output of neurons that are connected to local regions in the input, each computing a dot product between their weights and a small region they are connected to in the input volume using Equation (1) above. After the convolution operation, a rectified linear unit (ReLU) activation function, defined as $f(x) = max\ (0, x)$, is used to increase nonlinearity in the feature map. The convolutional layer has four hyperparameters, the number of filters c, their spatial extent f, the stride s, and the amount of zero padding p on the input image. For input volume of size $w_i \times h_i \times c$, filter of size $f \times f$ and stride s, the layer outputs a "volume" of size $w_o \times h_o \times c$, where

$$w_o = \left[\frac{w_i - f + 2p}{s}\right] + 1 \tag{22}$$

$$h_o = \left[\frac{h_i - f + 2p}{s}\right] + 1 \tag{23}$$

where [x] is the "floor" function of x, the greatest integer less than or equal to x.

A Pooling Layer summarizes the outputs of adjacent feature map values in the same kernel map. This progressively reduces the size of feature representation and the number of parameters resulting in faster computation in the network. Commonly used methods are max pooling, which takes the maximum within a rectangular neighbourhood defined by the kernel map, and average pooling, which reports the average output within a rectangular neighbourhood defined by the kernel map; see Figure 8. No training of weights occurs in a pooling layer. It has two hyperparameters, F and the "stride" s. It takes the output "volume" size of the preceding convolutional layer, $\boldsymbol{w}_{\mathrm{o}} \times \boldsymbol{h}_{\mathrm{o}} \times \mathrm{c}$, as input volume and outputs a "volume" of size $w_p \times h_p \times \mathrm{c}$, where

$$w_p = \left[\frac{w_o - F}{s}\right] + 1 \tag{24}$$

$$h_p = \left[\frac{h_o - F}{s}\right] + 1 \tag{25}$$

where c is the number of filters; F is the pooling layer spatial extent; and s is the stride used in the pooling layer.

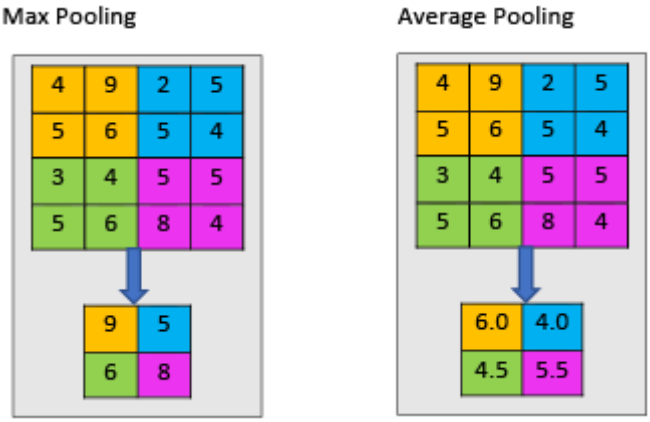


Figure 8. An illustration of the two pooling methods with a 2 × 2 filter and a stride of 2 applied.

The Fully Connected Layer is just a regular perceptron artificial neural network as described in section B2 (i) above. It takes the unrolled or flattened "volume" from the last convolutional or pooling layer as its input.

4. Results

4.1. Preliminary Analysis-Time Domain

We began by graphically exploring the original audio signals using plots of the bee audio signals from the Queen Present and Queen Absent hives; see Figure 9. In general, the plots reveal clear differences between the signals from these different hives in the time domain. The Queen Present signals showed less time variability and a smaller standard deviation in amplitude than the signals from the Queen Absent hives.

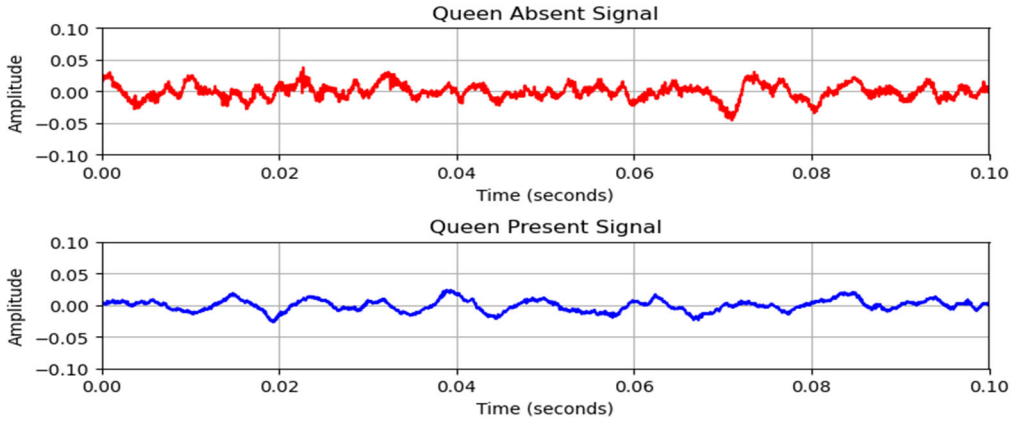


Figure 9. Plots of randomly selected 1 s clips for Queen Absent top (red) and Queen Present bottom (blue) audio signals.

4.2. Frequency Domain Analysis

The bee acoustic signals also appear to have distinctive spectral characteristics; see Figure 10. The Queen Absent signal displays higher energy levels overall. Moreover, the energy distributions over the frequency range are also different when comparing morning and afternoon signals. This suggests that hive status can be classified acoustically for hive monitoring. Our preliminary investigation indicated that, although the "fundamental" frequency of the bee sounds was of the order of 100 Hz, use of signals low pass filtered at 3000 Hz provided poorer classification performance than those filtered at 6000 Hz. The latter provided comparable performance to signals where no low pass filtering was applied. Hence, the classification models described below used the signals low pass filtered at 6000 Hz as input to the MFCC or STFT as appropriate. Example spectrograms of signals from Queen Present and Queen Absent hives are shown in Figure 11. Subtle differences can be observed between the spectrograms of the two hive types.

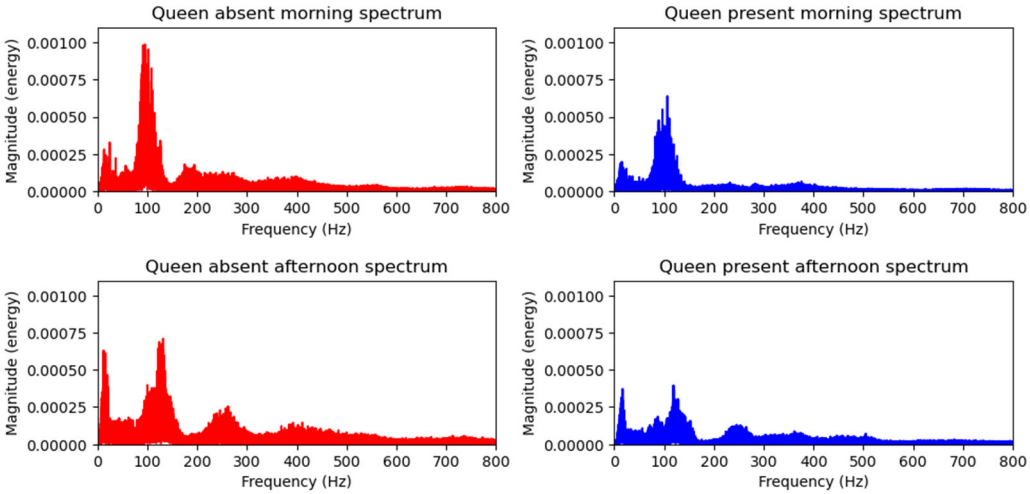


Figure 10. Acoustic Spectra for Queen Absent (red, **left**) and Queen Present (blue, **right**) hives. The upper graphs show data recorded in the morning, between 1 am and 2 am, whilst the lower ones display afternoon data recorded between 1 pm and 2 pm.

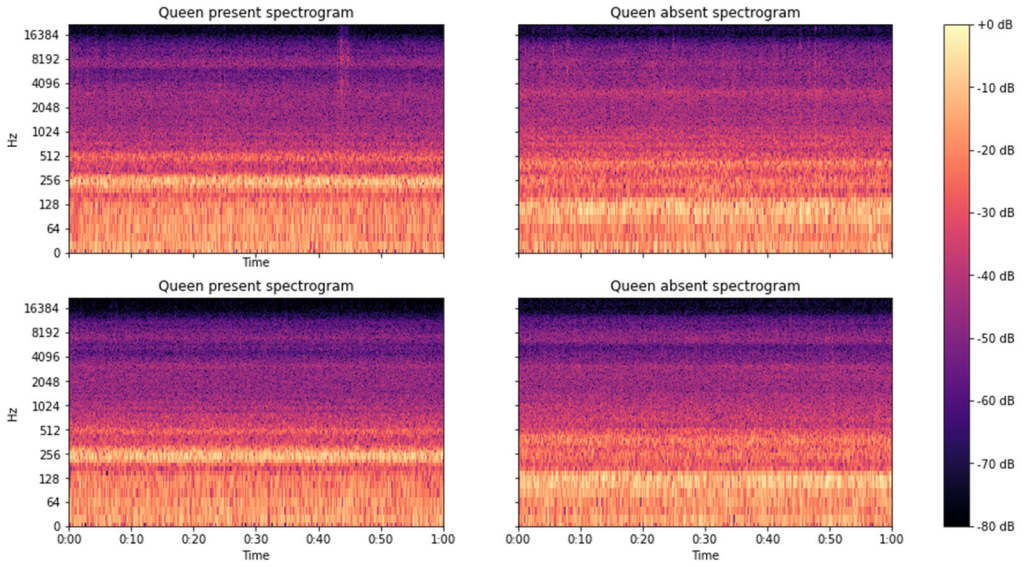


Figure 11. Example Acoustic Spectrograms for Queen Present (**left**) and "Queen Absent (**right**) hives. The brighter/lighter colours indicate more power in the signal in that frequency band. The Queen Present examples show a strong band centred around 250 Hz not observed in the Queen Absent ones, and whilst both sets show a band of moderate power around 500 Hz, the centre frequency of this band appears to be slightly higher for the Queen Present examples than for the Queen Absent ones.

4.3. Mel Frequency Domain

Firstly, in a similar manner to [20], we extracted 14 features, 13 MFCCs and the log energy, from the bee audio signals from the four hives using MATLAB [42]. The resultant MFCCs and log energy from control hives were combined to form the Queen Present or "healthy" hive class and those from the experimental hives to form the Queen Absent or "unhealthy" hive class. The mean for each Mel coefficient and the log energy was computed for the "healthy" and "unhealthy" hives classes. We then tested for equality of means between the two classes using an ANOVA test for each of the MFCCs and log energy averages. The test indicated that there is a statistically highly significant difference ($p < 0.001$) between the means of the "healthy" and "unhealthy" classes for all MFCCs and the log energy. This confirmed the initial observations are illustrated in Figures 9–11, suggesting that the two hive classes are readily distinguishable.

4.4. Evaluation Metrics

To evaluate the model performance the model accuracy, precision, recall and the F1 score was obtained for each classification task. The performance metrics are defined as

$$\text{Accuracy} = \frac{TP + TN}{N}, \tag{26}$$

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{27}$$

$$Recall = \frac{TP}{TP + FN}, \tag{28}$$

where TP = True positives, FP = False positives, FN = False negatives, TN = True negatives and N is the total number of hive status samples.

$$F1\ score = \frac{2 * precision * recall}{precision + recall}. \tag{29}$$

4.5. Machine Learning Experiments

In this section, we consider two approaches to hive classification using (1) MFCCs and (2) STFT spectrograms as features. To avoid unequal sample bias, only data from the 5th to the 9th of August 2012 were used to train and validate the models. Thus, each class had 240 one-minute-duration audio samples.

4.5.1. MFCCs as Features

From the 14 features extracted in Section 4.3, four combinations of 13 MFCCs and the log energy features were used as features to train three classifiers, a Logistic Regression, an MLP and an LSTM in our experiments. These are: dataset 1 with 13 MFCCs plus the log energy, dataset 2 with 13 MFCCs, dataset 3 with MFCCs minus the first coefficient but with the log energy included, and dataset 4 with MFCCs minus the first coefficient and omitting the log energy. These combinations were investigated due to their success in other related audio classification tasks [26]. For each of the datasets, a Logistic Regression, an MLP and an LSTM classifier were fitted. We used Scikit learn [42] for both the Logistic Regression [43] and the MLP classifiers [44]. For the LSTM network, we used a Keras framework [45], which includes functionality for optimizing model parameters automatically, with an architecture with 100 units, binary cross-entropy loss function and an ADAM optimizer. ADAM, whose name derives from Adaptive Memory Estimation, is an efficient stochastic gradient descent optimisation method [46]. The method computes adaptive learning rates for each parameter using first and second order moments of the gradients. It is commonly used in deep learning applications as it is straightforward to implement, is computationally efficient, and has relatively few memory requirements [46]. We also initially experimented with using other optimisation methods, such as RMSprop [47], but ADAM provided the best performance. For all the experiments, we used 80% of each class for training and 20% of each class for validation. The training set data were normalized so that each coefficient and the log energy had zero mean and unit variance to reduce the effects of undue influence of larger log energy values on the model training weights and to speed up learning. Results from the experiments are summarized in Table 2 below.

Table 2. Validation accuracy values (with perfect = 1.00) across the four datasets considered, as described in Section 4.5.1 above.

		Model Validation Accuracy		
Dataset	**Number of Features**	**Logistic Regression**	**MLP**	**LSTM**
Dataset 1	14	0.8743	0.9008	0.9178
Dataset 2	13	0.8554	0.8990	0.9038
Dataset 3	13	0.8704	0.8979	0.9108
Dataset 4	12	0.8141	0.8479	0.8744

Whatever the combination of features used, the LSTM model performed either as well as or better than each of the other models, achieving the best accuracy of 0.9178 on dataset 1; see Table 2. Whilst the MLP delivered a best accuracy of 0.9008, the Logistic Regression model could only reach an accuracy of 0.8743. One should note that the best accuracy results were generally achieved when all 14 features were used, while the worst accuracies correspond to the use of only 12 features.

4.5.2. Spectrograms as Features

Each of the 480 bee audio samples were divided into 5 s clips, and for each an STFT spectrogram was plotted in Matlab using the spectrogram() function, producing a total of 5760 spectrograms. The resultant spectrograms from the control hives were put in the "healthy" hive class and those from the experimental hives were put in the "unhealthy" hive class. The data were used to train and validate a CNN model with 80% of the data

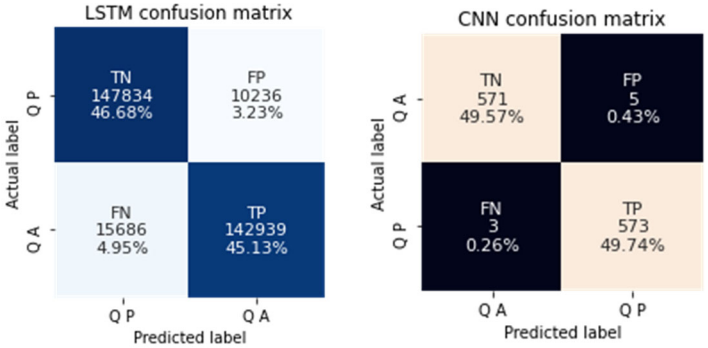
used for training and 20% used for validation. For the CNN network, we used a Keras framework [48] using automated parameter optimization and with an architecture using 64 channels, a 3×3 kernel, a single convolutional layer, a pooling layer with a 2×2 filter, binary cross-entropy loss function and an ADAM optimizer. As for the LSTM model, we determined that ADAM performed better than alternatives such as RMSprop [47]. We then compared the best model using MFCCs, the LSTM using 14 features, to a CNN model using spectrograms. The two models were also tested using data from a completely different hive, namely the one in Surrey, U.K. recorded in Summer 2020, providing impressive accuracy results.

The CNN model performed much better than the LSTM model, with a close to perfect classification accuracy. For our problem, false positives indicate that the queen is present when in actual fact the hive is queenless. This is the worst-case scenario, as queen absence, which seriously affects the health of the colony, needs prompt identification; ideally, one would prefer to have the possibility of this occurring as low as possible–see Table 3.

Table 3. Performance metrics for the best performing model (LSTM) applied to the 14 MFCC features, and the CNN models applied to the STFT spectrograms. Please note the Precision, Recall and F1-scores given are for the models applied to the validation dataset.

Model	Precision	Recall	F1-Score	Training Accuracy	Validation Accuracy	Test Accuracy
LSTM	0.92	0.92	0.92	0.9180	0.9178	0.9181
CNN	0.9931	0.9931	0.9931	0.9912	0.9931	0.9900

False negatives, on the other hand, are indications that the queen is absent when in fact she is still there (i.e., a false alarm). Although this is not as bad as the first scenario, high numbers of false alarms result in excessive inspections, resource wastage and hive disruptions, which ideally should be kept at a minimum. Thus, we want a classifier with both a high precision and a high recall, and for that, the F1 score is a good measure. High F1 scores for both models suggest that their respective accuracies have not been unduly influenced by a large number of true negatives. It also indicates that there is an even distribution among the correct predictions for the two classes and that there are low numbers of false positives and false negatives. This is informative, showing that the models are correctly detecting the presence or absence of a queen in the hive very well, with the CNN model delivering particularly highly accurate predictions. Confusion matrices for the LSTM and CNN models are shown in Figure 12 below.



(**a**) (**b**)

Figure 12. Confusion matrices for (**a**) the LSTM applied to the 14 MFCC features; (**b**) the CNN applied to the STFT spectrograms. QA and QP stands for Queen Absent and Queen Present, respectively.

Expanding on our earlier work [10], we tested our LSTM and CNN models using data supplied by a beekeeper from a different hive located in rural Surrey, U.K. recorded during Summer 2020. We obtained very good results for both models, particularly for

the CNN model which achieved a test mean accuracy of 98.61% on the unseen data, see Table 4, surprisingly almost 4% higher than the mean accuracy on the training data. For the LSTM model, we obtained a mean accuracy of 90.8% compared to a mean accuracy of 91.79% obtained on the training data. This indicates that both models have very good generalisation, although more data from different hives and locations might be needed to fully substantiate this before practical deployment.

Table 4. Summary Accuracy statistics for the LSTM and CNN models of the original dataset and the Cobham, Surrey dataset we tested our models on. For the original Arnia data, we used 10-fold cross-validation statistics, and 30 samples from the Surrey dataset were used to test our model.

Model + Dataset	LSTM Model, Arnia Data	LSTM Surrey, UK Data	CNN Model, Arnia Data	CNN, Surrey, UK Data
Mean	0.9179	0.9080	0.9465	0.9861
Standard deviation	0.001	0.0034	0.0500	0.0116

5. Discussion

Our work adds evidence supporting both beekeepers' claims that the sounds emitted by bees in a hive can indicate important information regarding the queen and the value of acoustics in automated monitoring of beehives. Our best model, the CNN model using spectrograms, achieved a very high accuracy of 99% discrimination between hives where the queen was absent and hives with the queen present with close to perfect precision and recall. This shows that acoustic monitoring could be a very useful tool for beekeepers to remotely monitor the status of their hives. Although audio spectrograms have been used to train a CNN model which produced high accuracies for various acoustic classification tasks, to our knowledge, no work to date has specifically used them to detect the presence or absence of a queen bee in a hive. Most previous work on honeybee audio analysis has focused on the detection or prediction of swarming.

Another interesting feature from our results is that for both the LSTM and the CNNs, the high accuracies were achieved with relatively simple networks rather than very deep networks as used in most previous studies. The fact that such small networks achieved such high accuracies indicates that the architectures used here are appropriate for the task.

6. Conclusions and Future Work

In this work, we addressed the problem of hive queen bee status using a variety of approaches. Results from standard classifiers, namely Logistic Regression and MLP, were compared to those from an LSTM classifier using different combinations of MFCCs. The best classifier–feature combination, namely the LSTM using 14 MFCCs, was then compared for performance to a CNN using spectrograms as input. The CNN model performed substantially better than the other models with a close-to-perfect predictive accuracy.

One limitation of this current work is the limited range of data on which the models have been trained and tested. In particular, the first dataset was acquired in highly controlled conditions (four hives, all in very close proximity to each other, with the queens being removed from two of the hives at a precisely known time) which is unlikely to be the case in a general scenario. For future work, we plan to further test our models on independently acquired data, for example, from the Open-Source Beehives (OSBH) project [49,50] or the NU-Hive project [51,52]. We also wish to investigate differences between African and European honeybees using acoustic signals through a collaboration with a university in Ghana, West Africa and to use machine learning with acoustic data to predict swarms.

It would also be desirable to be able to predict the death (or other loss) of the queen in advance or deduce that the queen is in poor health from remotely acquired signals. However, this would require evidence from a considerably larger body of data so that longer-term patterns in the signals leading up to the loss of a queen could be studied.

Author Contributions: Conceptualization, S.R. and G.H.; methodology, G.H., S.R. and J.-C.N.; software, S.R.; validation, S.R., G.H. and J.-C.N.; writing—original draft preparation, S.R and G.H.; writing—review and editing, S.R., G.H., O.D. and J.-C.N.; supervision, G.H., O.D. and J.-C.N.; project administration, G.H.; funding acquisition, G.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partly funded by Innovate UK as part of the Bee Smart project, grant number 45568.

Data Availability Statement: The data provided by Arnia Ltd. can be requested from the company. The data recorded from the hive in Surrey, U.K. can be obtained on request from the authors.

Acknowledgments: Stenford Ruvinga is grateful to the Graduate School of Kingston University for awarding him a Postgraduate Research Studentship enabling him to work on this project. We would all like to thank Arnia Ltd. for making their data available for us to use, and to beekeepers John Futcher, Colm Treacy and Stewart Westsmith for providing valuable insights into the life of honeybees.

Conflicts of Interest: The authors declare no conflict of interest with regard to this work.

References

1. Neumann, P.; Blacquière, T. The Darwin cure for apiculture? Natural selection and managed honeybee health. *Evol. Appl.* **2016**, *10*, 226–230. [CrossRef] [PubMed]
2. The World Wide Fund for Nature. Available online: https://www.wwf.org.uk/sites/default/files/2019-05/EofE%20bee%20report%202019%20FINAL_17MAY2019.pdf (accessed on 12 June 2020).
3. Sharma, V.P.; Kumar, N.R. Changes in honey bee behaviour and biology under the influence of cell phone radiations. *Curr. Sci.* **2010**, *98*, 1376–1378.
4. Boys, R. Listen to the Bees. Available online: https://beedata.com.mirror.hiveeyes.org/data2/listen/listenbees.htm (accessed on 4 January 2023).
5. Terenzi, A.; Cecchi, S.; Orcioni, S.; Piazza, F. Features Extraction Applied to the Analysis of the Sounds Emitted by Honeybees in a Beehive. In Proceedings of the 11th International Symposium on Image and Signal Processing and Analysis (ISPA), Dubrovnik, Croatia, 23–25 September 2019; pp. 3–8. [CrossRef]
6. Kirchner, W.H. Acoustical Communication in Honeybees. *Apidologie* **1993**, *24*, 297–307. [CrossRef]
7. Howard, D.; Duran, O.; Hunter, G. Signal Processing the Acoustics of Honeybees (*Apis mellifera*) to Identify the 'Queenless' State in Hives. In Proceedings of the Institute of Acoustics, Nottingham, UK, 13 May 2013.
8. Seeley, T.D.; Tautz, J. Worker Piping in Honeybee Swarms and its Role in Preparing for Liftoff. *J. Comp. Physiol.* **2001**, *187*, 667–676. [CrossRef] [PubMed]
9. Ferrari, S.; Silva, M.; Guarino, M.; Berckmans, D. Monitoring of swarming sounds in beehives for early detection of the swarming period. *Comput. Electron. Agric.* **2008**, *64*, 72–77. [CrossRef]
10. Ruvinga, S.; Hunter, G.J.A.; Duran, O.; Nebel, J.C. Use of LSTM Networks to Identify "Queenlessness" in Honeybee Hives from Audio Signals. In Proceedings of the 17th International Conference on Intelligent Environments (IE2021), Dubai, United Arab Emirates, 21–24 June 2021. [CrossRef]
11. Scheiner, R.; Abramson, C.I. Standard methods for behavioral studies of *Apis mellifera*. *J. Apic. Res.* **2013**, *52*, 1–58. [CrossRef]
12. Shaw, J.; Nugent, P. Long-wave infrared imaging for non-invasive beehive population assessment. *Opt. Express* **2011**, *19*, 399. [CrossRef] [PubMed]
13. Murphy, F.E.; Magno, M.; O'Leary, L. Big brother for bees (3B)—Energy neutral platform for remote monitoring of beehive imagery and sound. In Proceedings of the 6th IEEE International Workshop on Advances in Sensors and Interfaces (IWASI), Gallipoli, Italy, 18–19 June 2015. [CrossRef]
14. Campbell, J.; Mummert, L.; Sukthankar, R. Video monitoring of honey bee colonies at the hive entrance. In Proceedings of the Visual Observation and Analysis of Animal and Insect Behavior, ICPR 2008, Tampa, FL, USA, 8–11 December 2008; pp. 1–4.
15. Kachole, S.; Hunter, G.; Duran, O. A Computer Vision Approach to Monitoring the Activity and Well-Being of Honeybees. In Proceedings of the IE 2020: 16th International Conference on Intelligent Environments, Madrid, Spain, 20–23 July 2020. [CrossRef]
16. Crawford, E.; Leidenberger, S.; Norrström, N.; Niklasson, M. Using Video Footage for Observing Honeybee Behavior at Hive Entrances. *Bee World* **2022**, *99*, 139–142. [CrossRef]
17. Wenner, A.M. Sound Communication in Honeybees. *Sci. Am.* **1964**, *210*, 116–124. [CrossRef]
18. Eren, H.; Whiffler, L.; Manning, R. Electronic sensing and identification of queen bees in honeybee colonies. In Proceedings of the Instrumentation and Measurement Technology Conference, Ottawa, ON, Canada, 19–21 May 1997. [CrossRef]
19. Žgank, A. Acoustic Monitoring and Classification of Bee Swarm Activity using MFCC Feature Extraction and HMM Acoustic Modelling. In Proceedings of the ELEKTRO 2018, Mikulov, Czech Republic, 21–23 May 2018. [CrossRef]

20. Robles-Guerrero, A.; Saucedo-Anaya, T.; González-Ramírez, E.; De la Rosa-Vargas, J.I. Analysis of a multiclass classification problem by Lasso Logistic Regression and Singular Value Decomposition to identify sound patterns in queen-less bee colonies. *Comput. Electron. Agric.* **2019**, *159*, 69–74. [CrossRef]
21. Peng, R.; Ardekani, L.; Sharifzadeh, H. An Acoustic Signal Processing System for Identification of Queen-less Beehives. In Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Auckland, New Zealand, 7–10 December 2020; Available online: https://ieeexplore.ieee.org/document/9306388 (accessed on 14 January 2023).
22. Hochreiter, S.; Schmidhuber, J. Long Short-term Memory. *Neural Comput.* **1997**, *8*, 1735–1780. [CrossRef] [PubMed]
23. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
24. Pheng, F.; Song, Z. Comparison of Different Implementations of MFCC. *J. Comput. Sci. Technol.* **2001**, *16*, 582–589.
25. Davis, S.; Mermelstein, P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. Acoust. Speech Signal Process* **1980**, *28*, 357–366. [CrossRef]
26. Ganchev, T.; Fakotakis, N.; Kokkinakis, G. Comparative evaluation of various MFCC implementations on the speaker verification task. In Proceedings of the 10th International Conference on Speech and Computer, Patras, Greece, 17–19 October 2005.
27. Beritelli, F.; Grasso, R. A pattern recognition system for environmental sound classification based on MFCCs and neural networks. In Proceedings of the 2nd International Conference on Signal Processing and Communication Systems, Gold Coast, Australia, 15–17 December 2008. [CrossRef]
28. Kour, G.; Mehan, N. Music Genre Classification using MFCC, SVM and BPNN. *Int. J. Comput. Appl.* **2015**, *112*, 6.
29. Deng, M.; Meng, T.; Cao, J.; Wang, S.; Zhang, J.; Fan, H. Heart sound classification based on improved MFCC features and convolutional recurrent neural networks. *Neural Netw.* **2020**, *130*, 22–32. [CrossRef] [PubMed]
30. Mohamed, A. Deep Neural Network Acoustic Models for ASR. Ph.D. Thesis, University of Toronto, Toronto, ON, Canada, 2014.
31. Shimodaira, H.; Rennals, S. Speech Signal Analysis. 2013. Available online: https://www.inf.ed.ac.uk/teaching/courses/asr/2012-13/asr02-signal-4up.pdf (accessed on 4 January 2023).
32. Paliwal, K.; Lyons, J.; Wojcicki, K. Preference for 20–40 ms window duration in speech analysis. In Proceedings of the 4th International Conference on Signal Processing and Communication Systems, Gold Coast, Australia, 13–15 December 2010. [CrossRef]
33. Wyse, L. Audio spectrogram representations for processing with convolutional neural networks. *arXiv* **2017**, arXiv:1706.09559.
34. Bishop, C.M. *Neural Networks for Pattern Recognition*; Oxford University Press: Oxford, UK, 1995.
35. Carling, A. *Introduction to Neural Networks*; Sigma Press: Cheshire, UK, 1992.
36. Goodfellow, I.; Bengio, Y.; Courville, A. Deep Learning. Available online: http://www.deeplearningbook.org (accessed on 14 January 2023).
37. Gibaru, O. Neural Network. Available online: https://www.oliviergibaru.org/courses/ML_NeuralNetwork.html (accessed on 4 May 2022).
38. Ng, A.; Katanforoosh, K.; Bensouda Mourri, Y. Sequence Models. Available online: https://www.coursera.org/learn/nlp-sequence-models (accessed on 12 January 2022).
39. Olah, C. Understanding LSTM Networks. Available online: http://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed on 18 January 2023).
40. Sak, H.; Senior, A.; Beaufays, F. Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling. In Proceedings of the INTERSPEECH 2014 (15th Annual Conference of the International Speech Communication Association), Singapore, 14–18 September 2014; pp. 338–342.
41. Ratan, P. What Is the Convolutional Neural Network Architecture? 2021. Available online: https://www.analyticsvidhya.com/blog/2020/10/what-is-the-convolutional-neural-network-architecture/ (accessed on 14 November 2022).
42. Mathworks.com. MFCC. Available online: https://uk.mathworks.com/help/audio/ref/mfcc.html (accessed on 12 January 2022).
43. Scikit-learn. LogisticRegression. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html (accessed on 15 January 2023).
44. Scikit-Learn. MLPClassifier. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html (accessed on 15 January 2022).
45. Keras. LSTM layer. Available online: https://keras.io/api/layers/recurrent_layers/lstm/ (accessed on 13 January 2022).
46. Kingma, D.P.; Ba, J.L. ADAM: A Method for Stochastic Optimization. *arXiv* **2015**, arXiv:1412.6980.
47. Hinton, G.; Srivastava, S.; Swersky, K. Neural Networks for Machine Learning—Lecture 6e—Rmsprop: Divide the Gradient by a Running Average of Its Recent Magnitude. 2012. Available online: http://www.cs.toronto.edu/~hinton/coursera/lecture6/lec6.pdf (accessed on 17 February 2023).
48. Keras. Convolution Layers. Available online: https://keras.io/api/layers/convolution_layers/ (accessed on 18 January 2022).
49. Open-Source Beehives Project. Available online: https://zenodo.org/communities/opensourcebeehives/?page=1&size=20 (accessed on 17 January 2023).
50. Nolasco, I.; Benetos, E. To be or not to bee: Investigating machine learning approaches for beehive sound recognition. *arXiv* **2018**, arXiv:1811.06016. [CrossRef]

51. Cecchi, S.; Terenzi, A.; Orcioni, S.; Riolo, P.; Ruschioni, S.; Isidoro, N. A preliminary study of sounds emitted by honeybees in a beehive. In Proceedings of the 144th Convention of the Audio Engineering Society, Paper 9981, Milan, Italy, 23–26 May 2018; Available online: http://www.aes.org/e-lib/browse.cfm?elib=19498 (accessed on 24 February 2023).
52. Terenzi, A.; Cecchi, S.; Spinsante, S. On the Importance of the Sound Emitted by Honey Bee Hives. *Vet. Sci.* **2020**, *7*, 168. [CrossRef]