Swarming Detection in Smart Beehives Using Auto Encoders for Audio Data

Pascal Janetzky, Melanie Schaller, Anna Krause, and Andreas Hotho

Computer Science Department, University of Würzburg, Würzburg, Germany

{janetzky,schaller,anna.krause,hotho}@informatik.uni-wuerzburg.de

***Abstract*—Swarming is the natural mechanism by which bee colonies reproduce, but for beekeepers it is a challenge. Precision beekeeping can aid their work through early notifications about impending swarms. In this work, we focus on identifying swarms and their early indicators in audio data captured from a smart beehive. The challenge with such domain-specific data is the low availability of labelled samples, the strong label imbalance, and the recording of undesired sources. We approach this challenge through a two-step setup: First, we use an auto encoder network to detect sounds from mechanical sources and then use it to clean data. Secondly, on the cleaned data we then employ a second network to identify event-related bee sounds. Using spectrogram features, our networks are able to reach a balanced accuracy score of more than $99\,\%$ in the detection of special bee events. The findings of this initial study can serve as the starting point for further research on handling imbalanced data collections from smart, remote sensor environments that also contain undesired signals.**

***Index Terms*—audio processing, machine learning, auto encoding networks, precision beekeeping**

I. Introduction

Swarming is a natural mechanism by which bee colonies reproduce and takes place in late spring and early summer. During a swarm, the old queen and $50\,\%$ to $70\,\%$ of the worker bees leave the hive and found a new colony, while a young queen takes over the old colony. In standard European beekeeping, apiarists try to prevent swarming altogether, which requires regular and intensive interactions with the colonies. One way to reduce interaction is to monitor these colonies using precision beekeeping systems to detect early swarm indicators that can assist beekeepers in their decision making. Due to the widespread availability of affordable electronic sensors, the setup of the required precision beekeeping systems is much easier and allows the automated observation of the bee colonies. In the we4bee project[1], such smart beehives equipped with many sensors have been distributed to mainly educational institutions in Germany. These smart bee hives are continuously collecting data and form the base of the analysis.

However, a challenge is the availability of *labelled* data. While the installed sensors monitor the beehives around the clock, labelling the data incurs massive manual effort, as domain experts are necessary to reach high label accuracy and high inter-annotator agreement. In addition, special bee events such as swarming are exceedingly rare and tend to occur in fixed parts of the year only. This leads to a high label imbalance, where the normal behaviour is dominantly represented in the data and the events of interest are very sparse. The analysis of such data requires methods that can handle highly imbalanced and sparsely labeled data sets. In this paper, we present an approach based on auto encoders, visualized in fig. 1, that allows us to detect samples from multiple classes in audio data recorded from one exemplary beehive. Our contributions are:

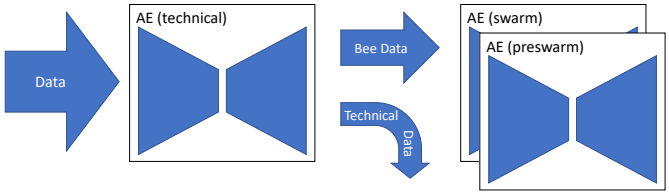

Fig. 1. Our proposed two-staged approach. On the we4bee audio dataset, the first stage uses auto encoders to detect and filter mechanical sounds (left), and the second stage uses auto encoders to identify the swarming event and pre-swarming phase of bees in the filtered data (right).

- introducing a mid-sized ($\sim$6000 data points) audio dataset with 5 classes
- modelling the problem via a two-step anomaly detection approach
- identifying swarms with high precision in audio data using auto encoders

II. Related work

The "To bee or not to bee" dataset [1] contains audio recordings from the OpenSource bee hive (OSBH) [2] and NU-Hive [2] projects. The dataset provides 12 hours of audio data, with segments labelled as containing bee sounds constituting $25\,\%$ of the total recordings; the remaining segments are labelled as not containing bee sounds. On this dataset, Nolasco and Benetos [3] evaluate the use of a support vector machine classifier (SVM) and a convolutional neural network (CNN). Their study shows that an SVM classifier is mostly superior to a CNN, but also indicates that a larger context (*i.e.*, a longer

[1] https://we4bee.org

[2] https://fablabbcn.org/projects/osbh-open-source-beehives

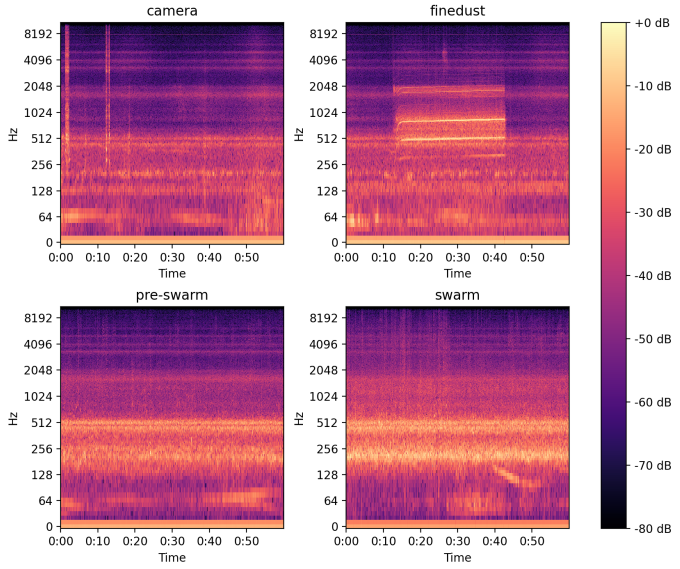

Fig. 2. Spectrograms of the classes contained in we4bee audio dataset. Row-wise, classes camera, finedust, pre-swarming and swarming are visualized. *Camera* samples contain two distinctive short spikes, and *finedust* samples have a noticeable block of increased energy between 512Hz and 2048Hz. The short-lived swarming of a bee colony produces clearly audible buzzing sounds, which can be seen in the *swarm* sample.

audio input) can improve the network's performance. We build upon these findings and use 15 s audio data as input.

Research by Žgank [4] used recordings from the OSBH project to train a one-state hidden Markov model (HMM) to detect bee swarms. Their work uses 65 min of audio data in total, of which 45 min are swarming data. After pre-processing the source signals to 16 kHz mono audio, [4] extract mel frequency cepstral coefficients and label the recordings as normal or swarm. On the annotated data, the HMM achieves an accuracy score of 80.89 %. In our work, we deviate from this approach by using 44 kHz stereo recordings and do not employ any re-sampling strategies. Kulyukin et al. [5] introduce two datasets, which are composed of audio recordings of bees, crickets, and ambient noise. The "Buzz1" dataset contains 10260 samples, and the "Buzz2" dataset consists of 12914 samples. Each sample is manually labelled into one of the three classes and 2 s in duration. The authors use a CNN to classify raw audio snippets and spectrogram representations. In contrast to this research, we use convolutional AEs on spectrograms covering 15 s.

Davidson et al. [6] use an AE on temperature data collected from four beehives. Their study focuses on the typical swarming period, May to September, and focuses on the general detection of anomalous signals, not just the swarming events. Their approach allows them to detect external influences on the monitored beehive, such as the opening of the lid to yield honey. Our research extends Davidson et al.'s usage of an AE to the detection of anomalous data in audio recordings.

Finally, Hadjur et al. [7] provide an overview of machine learning approaches in precision beekeeping. They show that researchers mainly use image, temperature, and audio data in their research. Further, their analysis uncovers that scholars mainly use spectral features (*e.g.*, short-time Fourier transformations) and their derivatives (*e.g.*, mel-scaled spectrograms). Following these findings, we also use spectrograms as features.

TABLE I
NUMBER OF 15 S SAMPLES PER CLASS IN THE TRAIN, VALIDATION, AND TEST SUBSET OF THE WE4BEE AUDIO DATASET.

Subset (total)	Class pre-swarm	swarm	camera	finedust	other
Train (2722)	45	32	73	353	2219
Validation (681)	11	8	18	89	555
Test (3096)	60	40	78	681	2237
Total (6499)	116	80	169	1123	5011

The datasets so far discussed and used by the related work are all labelled or annotated in some way. Our dataset, described in section III, differs from this and poses two challenges: first, it is largely unlabelled, and second, the few labels are distributed in a highly imbalanced way. Using this dataset, we build upon related research and use spectrograms to train AE networks on challenging audio recordings collected from a smart beehive.

III. WE4BEE AUDIO DATASET

The we4bee project runs app. 100 smart beehives, which are all equipped with two microphones. Since almost three years, one of the hive collects the audio data which results in approximately 20 000 h of *unlabelled* audio recordings. As mentioned in section I, labelling a dataset of this size is expensive and relies on manual labelling by domain experts. Furthermore, the events of interest are very rare which makes it even more difficult to come up with a proper dataset. To construct an initial dataset, we restricted the data collection to a selection of twelve days, of which 30 min excerpts from eight days cover the bee's winter period, two days cover the swarming, and two days constitute recordings where no special bee events occurred.

The `we4bee audio dataset` consists of 1625 stereo audio recordings sampled at 44.4 kHz, captured on four different and non-consecutive days, with each 60 s recording being split into four non-overlapping 15 s snippets. The dataset is divided into distinct train and test sets, which were recorded on different days, prohibiting data leakage.

In our data, we identified four classes, visualized in fig. 2: two classes of particular interest to apiarists, *pre-swarm* and *swarm* (which denote samples in which the bees are in a pre-swarming and swarming state, respectively), and two classes of sounds from mechanical sources. These two classes, *finedust* and *camera*, identify samples on which the smart beehive's automated monitoring system could be heard. Lastly, samples that could not be positively assigned to one of the previous classes were collected in an additional fifth class called *other*.

The labels for the mechanical classes were derived by mapping the recording's timestamps to the times at which the automated monitoring systems were running. The labels for

TABLE II
RANGES FOR THE HYPERPARAMETER SEARCH. *Filters* DETERMINES THE SIZE OF THE CONVOLUTION FILTERS IN CONNECTION WITH THE NUMBER OF ENCODING LAYERS (NOT OPTIMIZED).

Parameter	Search range
Learning rate	[0.0001, 0.01], uniform
Epochs	[20, 300], step size of 10
Batch size	[8, 32], step size of 8
Latent size	[8, 256], step size of 8
Filters	[8, 64], step size of 4

swarms are derived from beekeeping logs and by checking the hive's weight for sudden drops that lead to consistently lower weight. For the pre-swarm class, we labelled a duration of $15\,\mathrm{min}$ prior to the actual swarming event as *pre-swarm*. From the training data, stratified $20\,\%$ of the samples are used as a validation subset, leading to the distribution shown in table I. Due to the challenge of appropriately labelling data samples, our amount of labelled data is small compared to the larger, but unlabelled, *other* class.

IV. METHODS

Anomaly detection is the task of separating well-defined data points from outliers which are called normal data points and anomalous data points respectively [8, 9]. We frame our swarming detection task as a two-stage anomaly detection task shown in fig. 1: the first stage learns to identify mechanical sounds, the second stage learns to identify swarms and pre-swarms against the *other* background. To this end, we employ auto encoders (AE) as anomaly detectors [9, 10, 11]. These AEs consist of two parts: the encoder and the decoder. The encoding stage learns a dimensionality-reduced version of the data, which is projected into a latent space smaller than the input size [12]. From this space, the decoder learns to reconstruct the original input data. In the context of anomaly detection, the AE is trained on normal, non-anomalous data only, using a cost function that computes a difference between the original input and its reconstruction, such as the mean-squared error function. To detect anomalous data points, various methods exist [9]. One common method is calculating a threshold [13]. In our work, this threshold was determined by maximizing the F1 score of the anomalous class on the validation data set.

The AE's encoder encodes the $64\,\mathrm{x}\,512\,\mathrm{x}\,2$ inputs via a stack of 5 convolution layers and one linear layer into the flattened representation, leading to the latent space of size N. All encoding layers use `relu` activation functions and a kernel size of $(2, 2)$ with a stride of 2. The kernel size was chosen to successively half the feature dimensions. The decoder stage consists of one linear layer, and 4 transposed convolution layers with a kernel of $(3, 3)$ and a stride of 2, where all layers except the final one use `relu` activations. The output has the shape $64\,\mathrm{x}\,512\,\mathrm{x}\,2$. The hyperparameters were determined via a parameter search.

TABLE III
SELECTED HYPERPARAMETERS OF THE AUTO ENCODING MODELS. THE FIRST ROW INDICATES THE THREE EXPERIMENTS CONDUCTED AND THE SECOND ROW THE AEs TRAINED ON THE RESPECTIVE FEATURE SET, WHERE *Spec.* IS SHORT FOR SPECTROGRAMS.

Experiment	technical		pre-swarm		swarm	
Parameter	PCA	Spec.	PCA	Spec.	PCA	Spec.
Learning rate	0.0011	0.0004	0.0014	0.0011	0.0147	0.0024
Epochs	170	290	40	150	220	200
Batch size	24	8	16	24	8	24
Latent size	16	208	80	216	192	16
Filters	24	64	8	16	20	24

V. EXPERIMENTAL SETUP

A. Pre-processing

As part of the pre-processing, the $50\,\mathrm{Hz}$ grid frequency was filtered from each sample. Two feature sets were computed on the samples: spectrograms [14, 15, 16, 17], and PCA vectors [14, 16] retrieved from the spectrograms. The parameters for the spectrogram computation used an FFT window of size 127 and a hop length of 1294. The larger hop length was chosen to ensure an even distribution of frames computed from the audio signal and to capture information from events shorter than 15 seconds, while maintaining a manageable feature size. The resulting per-snippet feature has the shape $64\,\mathrm{x}\,512\,\mathrm{x}\,2$, where 64 is the number of frequency bins, 512 is the number of FFT frames extracted from the audio signal, and 2 is the number of channels, having one feature matrix per audio channel. Preliminary experiments with varying numbers of frames have shown, that more frames per input feature lead to models that generalize poorly, while fewer frames per input feature lead to insufficient model performance. For the PCA features, `sklearn` [18] was used to compute a PCA with 64 components on the spectrograms per audio channel. Finally, we applied a z-normalization per audio channel on the data.

B. Baselines

An Isolation Forest [19] with default parameters and its contamination value equal to the percentage of the anomalous samples present in the training data was used as a baseline. We also trained a supervised Random Forest [20] with default settings and a maximum tree depth of 2 to determine an upper bound for the performance. Both algorithms were trained on flattened spectrogram features.

C. Experiments

We conducted three experiments on the `we4bee audio dataset` (cf. section III), following the pipeline visualized in fig. 1: (1) The `technical` experiment trains the AE to differentiate mechanical samples from event-related bee samples. The mechanical sounds (*camera*, *finedust*) constitute the normal behaviour and the event-related bee sounds (*pre-swarm*, *swarm*) the anomalies. (2) The `pre-swarm` and (3) `swarm` experiment train models to identify pre-swarming and swarms respectively. Both models use the class *other* as normal data for training.

TABLE IV
RESULTS OF THE AEs ON THE RESPECTIVE FEATURE SETS. THE NUMBERS INDICATE THE BALANCED ACCURACY SCORE (*Acc.*) AND THE F1 SCORE OF THE ANOMALOUS DATA (*F1*) ON THE TEST DATASET. *IF* IS FOR ISOLATION FOREST, *RF* FOR RANDOM FOREST, AND *AE+X* IS THE AUTO ENCODER TRAINED ON THE PCA OR SPECTROGRAM FEATURES.

	IF		AE+PCA		AE+Spec.		RF	
Experiment	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
technical	58.72	28.00	68.90	39.00	65.02	35.00	98.41	95.00
pre-swarm	55.48	17.00	62.01	24.00	59.22	22.00	76.42	62.00
swarm	63.66	40.00	99.24	70.00	99.73	87.00	99.75	88.00

The AEs were trained with the mean squared error as the loss function using the Adam optimizer [21]. The evaluation metrics are balanced accuracy and the F1 score of the anomalous class. The balanced accuracy score ranges from 0 (worst) to 1 (best) and is the averaged recall of both classes (i.e., normal and anomalous behaviour) [22, 23]. The F1 score ranges from 0 (worst) to 1 (best) and is the harmonic mean of precision and recall. A held-out test set consisting of both normal and anomalous behaviour was used to evaluate the models' performance. For all experiments, we used `Optuna` [24] to conduct a separate hyperparameter study over 100 trials using the search details defined in table II. The finally selected values are given in table III.

VI. RESULTS AND DISCUSSION

The results in table IV show that the audio signals contain sufficient information to allow accurate classification into their respective classes. In all three experiments, our AE models outperform the baseline Isolation Forests. Furthermore, in one experiment the AEs match the performance of the supervised Random Forest model.

In the first experiment `technical`, the Isolation Forest reaches a baseline balanced accuracy of $58.72\,\%$ in the detection of event-related bee sounds. This baseline is surpassed by both AEs, with the AE trained on the PCA features reaching $68.90\,\%$. The Random Forest model reaches both the highest balanced accuracy score, $98.41\,\%$ and the highest F1 score on the anomalous class, $95.00\,\%$. In this experiment, the winter samples present in our dataset might provide an additional challenge for the auto encoding models. While the automated systems are running year-round, bee colonies change their behaviour in the winter months [25]. In listening tests, we found that while bees can still be heard humming, they do so with strongly decreased intensity. We suspect that the limited amount of winter data is not sufficient to let the AEs learn this as normal behaviour. Conversely, collecting more samples from the winter and the transition period should improve the models' performance.

The second experiment, `pre-swarm` is the most challenging of the three experiments, as the results (c.f. table IV) show. Here, the baseline Isolation Forest model is not able to discern the bee's everyday behaviour from the pre-swarming one. The AE models fare better, both reaching a balanced accuracy around $60\,\%$, but come with a low recall of just over $20\,\%$. The Random Forest model reaches a higher balanced accuracy of $76.42\,\%$. The challenge in this experiment can be attributed to two factors: First, in contrast to sudden swarming events, there is only a minor difference between constant bee-humming and pre-swarming behavior, though sound signals indicative of an upcoming swarm are known in the literature [26, 27]. Second, labeling a 15-minute duration preceding a known swarming event as *pre-swarm* data may not be adequate since the pre-swarming phase can commence much earlier. This could result in a situation where numerous samples from the pre-swarm phase are included in the class *other*. Increasing the amount of pre-swarming data while adapting the labelling process seem to be promising approaches to improve the performance.

In the third experiment, `swarm`, the baseline Isolation Forest reaches a balanced accuracy score of $63.66\,\%$. The model is clearly surpassed by the AEs and the Random Forest model, with both approaches reaching a balanced accuracy score of over $99\,\%$. While the AE trained on the spectrogram data and the PCA-trained AE reach similar accuracy scores, their F1 scores of the anomalous class differ, which is markedly higher for the spectrogram-only model. The calculation of the PCA may result in reduced discriminability between samples from the *swarm* and *other* classes.

VII. CONCLUSION

Smart beehives provide vast amounts of unlabeled data. In this work, we created the `we4bee audio dataset`, which is a collection of more than 1500 stereo audio recordings from a smart beehive equipped with microphones. For this dataset, we constructed a labelling pipeline that labels bee-related events and sounds from mechanical sources. The challenges in this dataset are twofold: First, the recordings contain both event-related bee sounds and mechanical sounds. Second, the label distribution is highly imbalanced, as the bee events constitute only a minority of the total data samples. To identify event-related bee sounds, we used a two-stage approach: first, we learn to identify the mechanical sounds, then we learn to detect the swarming and pre-swarming samples. Our studies show that convolutional AEs can identify a swarming bee colony, but cannot reach the performance of a Random Forest model in the challenging detection of the pre-swarming phase. Analyzing this further is a promising direction for future research and involves gathering a larger quantity of labelled data samples and deploying the models directly on smart beehives.

ACKNOWLEDGMENT

The authors would like to thank Claudia Leikam, Hans Neumayr, and Padraig Davidson, all from the we4bee project, for maintaining the systems and data used in this research.

REFERENCES

[1] I. Nolasco and E. Benetos, "To bee or not to bee: An annotated dataset for beehive sound recognition," Jul. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1321278

[2] S. Cecchi, A. Terenzi, S. Orcioni, P. Riolo, S. Ruschioni, and N. Isidoro, "A preliminary study of sounds emitted by honey bees in a beehive," in *Audio Engineering Society Convention 144*. Audio Engineering Society, 2018.

[3] I. Nolasco and E. Benetos, "To bee or not to bee: Investigating machine learning approaches for beehive sound recognition," in *Workshop on Detection and Classification of Acoustic Scenes and Events*, 2018.

[4] A. Žgank, "Acoustic monitoring and classification of bee swarm activity using mfcc feature extraction and hmm acoustic modeling," in *2018 ELEKTRO*. IEEE, 2018, pp. 1–4.

[5] V. Kulyukin, S. Mukherjee, and P. Amlathe, "Toward audio beehive monitoring: Deep learning vs. standard machine learning in classifying beehive audio samples," *Applied Sciences*, vol. 8, no. 9, p. 1573, 2018.

[6] P. Davidson, M. Steininger, F. Lautenschlager, K. Kobs, A. Krause, and A. Hotho, "Anomaly detection in beehives using deep recurrent autoencoders," in *Proceedings of the 9th International Conference on Sensor Networks (SENSORNETS 2020)*, no. 9. SCITEPRESS – Science and Technology Publications, Lda., 2020, pp. 142–149.

[7] H. Hadjur, D. Ammar, and L. Lefèvre, "Toward an intelligent and efficient beehive: A survey of precision beekeeping systems and services," *Computers and Electronics in Agriculture*, vol. 192, p. 106604, 2022.

[8] V. Chandola, A. Banerjee, and V. Kumar, "Outlier detection: A survey," *ACM Computing Surveys*, vol. 14, p. 15, 2007.

[9] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *CoRR*, vol. abs/1901.03407, 2019. [Online]. Available: http://arxiv.org/abs/1901.03407

[10] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE journal*, vol. 37, no. 2, pp. 233–243, 1991.

[11] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Artificial Neural Networks and Machine Learning–ICANN 2011: 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I 21*. Springer, 2011, pp. 52–59.

[12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[13] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal processing*, vol. 99, pp. 215–249, 2014.

[14] M.-T. Ramsey, M. Bencsik, M. I. Newton, M. Reyes, M. Pioz, D. Crauser, N. S. Delso, and Y. Le Conte, "The prediction of swarming in honeybee colonies using vibrational spectra," *Scientific reports*, vol. 10, no. 1, p. 9798, 2020.

[15] N. Pérez, F. Jesús, C. Pérez, S. Niell, A. Draper, N. Obrusnik, P. Zinemanas, Y. M. Spina, L. C. Letelier, and P. Monzón, "Continuous monitoring of beehives' sound for environmental pollution control," *Ecological engineering*, vol. 90, pp. 326–330, 2016.

[16] M. Ramsey, M. Bencsik, and M. I. Newton, "Long-term trends in the honeybee 'whooping signal' revealed by automated detection," *PloS one*, vol. 12, no. 2, p. e0171162, 2017.

[17] S. Ferrari, M. Silva, M. Guarino, and D. Berckmans, "Monitoring of swarming sounds in bee hives for early detection of the swarming period," *Computers and electronics in agriculture*, vol. 64, no. 1, pp. 72–77, 2008.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[19] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth ieee international conference on data mining*. IEEE, 2008, pp. 413–422.

[20] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[22] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The balanced accuracy and its posterior distribution," in *2010 20th international conference on pattern recognition*. IEEE, 2010, pp. 3121–3124.

[23] J. D. Kelleher, B. Mac Namee, and A. D'arcy, *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies*. MIT press, 2020.

[24] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[25] M. Calovi, C. M. Grozinger, D. A. Miller, and S. C. Goslee, "Summer weather conditions influence winter survival of honey bees (apis mellifera) in the northeastern united states," *Scientific reports*, vol. 11, no. 1, p. 1553, 2021.

[26] A. Qandour, I. Ahmad, D. Habibi, and M. Leppard, "Remote beehive monitoring using acoustic signals," *Acoustics Australia / Australian Acoustical Society*, vol. 42, pp. 204–209, 12 2014.

[27] A. Terenzi, S. Cecchi, and S. Spinsante, "On the importance of the sound emitted by honey bee hives," *Veterinary Sciences*, vol. 7, no. 4, p. 168, 2020.