Synthetic Swarm Mosquito Dataset for Acoustic Classification: A Proof of Concept

1st Thai Duy Dinh
Falculty of Engineering
Vietnamese-German University
Ho Chi Minh City, Vietnam
10421014@student.vgu.edu.vn

2nd Minh Luan Vo
Falculty of Engineering
Vietnamese-German University
Ho Chi Minh City, Vietnam
10423074@student.vgu.edu.vn

Cuong Tuan Nguyen †
Falculty of Engineering
Vietnamese-German University
Ho Chi Minh City, Vietnam
cuong.nt2@vgu.edu.vn

Hien Bich Vo †
Falculty of Engineering
Vietnamese-German University
Ho Chi Minh City, Vietnam
hien.vb@vgu.edu.vn

***Abstract*—Mosquito-borne diseases pose a serious global health threat, causing over 700,000 deaths annually. This work introduces a proof-of-concept Synthetic Swarm Mosquito Dataset for Acoustic Classification, created to simulate realistic multi-species and noisy swarm conditions. Unlike conventional datasets that require labor-intensive recording of individual mosquitoes, the synthetic approach enables scalable data generation while reducing human resource demands. Synthetic swarm audio generation is the core novelty of this work. This approach facilitates the development of realistic, scalable multi-species datasets that would be impractical to collect through fieldwork. Using log-mel spectrograms, we evaluated lightweight deep learning architectures for the classification of mosquito species. Experiments show that these models can effectively identify six major mosquito vectors and are suitable for deployment on embedded low-power devices. The study demonstrates the potential of synthetic swarm audio datasets to accelerate acoustic mosquito research and enable scalable real-time surveillance solutions. The public dataset used in this study can be found here [1]**

***Index Terms*—Synthetic swarm audio, mosquito audio classification, log-mel spectrogram, convolutional neural networks, vector surveillance**

I. Introduction

Mosquito-borne diseases—including dengue, malaria, and Zika—infect hundreds of millions each year, with key vectors such as *Aedes aegypti*, *Anopheles*, and *Culex* driving transmission [1]. Their growing range, fueled by urbanization and climate change, underscores the need for scalable surveillance.

Conventional methods like trapping and lab identification are slow and labor-intensive. These limits were evident during the 2015–2016 Zika outbreak, where delayed vector detection worsened the spread [2].

Audio-based detection via wingbeat frequency offers a low-cost, non-invasive alternative [3], but most systems are built for clean environments and single-species detection—limiting real-world applicability on embedded hardware.

This work introduces a compact, real-time mosquito detection and classification system tailored for edge deployment. It combines low-power audio sensing with a lightweight dual-output neural network for both presence and species prediction (Figure 1).

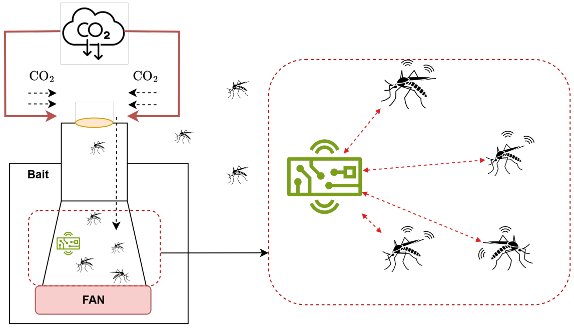


Fig. 1: Smart trap system with CO_2 bait, audio sensing, and onboard classification.

II. Related Work

A. Audio-Based Mosquito Monitoring

Traditional mosquito monitoring systems focused on extracting wingbeat frequencies under clean conditions, often limited to single-species detection. Kiskin et al. [4] pioneered the use of convolutional neural networks (CNNs) on spectrograms, demonstrating high accuracy in controlled settings.

† are the correspondence authors.

[1] https://github.com/duydinhthai27/synthetic-swarm-mosquito

The HumBug Project [5] introduced large-scale, real-world mosquito audio data collected via smartphones, enabling broader applicability. Fernandes et al. [6] achieved state-of-the-art binary classification using mel spectrograms and compact residual CNNs. More recently, Ramos et al. [7] developed a low-power audio recording system capable of real-time mosquito detection on embedded hardware.

B. Feature Representations

Early systems used fast Fourier transform (FFT) and mel-frequency cepstral coefficients (MFCCs), but these were sensitive to noise. Mel spectrograms proved more robust and better suited for CNNs. Wavelet-based methods such as the continuous wavelet transform (CWT) improved time-frequency localization. Yang et al. [8] showed that CWT enhanced detection in noisy environments. However, many approaches still rely on hand-crafted features rather than end-to-end learning.

C. Deep Learning Architectures

CNNs remain the most common choice for mosquito audio classification. Kiskin et al. [4] and Fernandes et al. [6] used CNNs to classify mosquito species with high precision. Wang et al. [9] introduced a convolutional recurrent neural network (CRNN) for modeling temporal dynamics.

To support edge deployment, lightweight models have been proposed. Toledo et al. [10] achieved 96% accuracy with an LSTM model containing fewer than 70k parameters. Fernandes et al. [6] also optimized residual CNNs for TinyML platforms.

D. Noise Robustness

Environmental noise remains a major challenge. Supratak et al. [11] addressed this with MosquitoSong+, incorporating noise augmentation and attention mechanisms to maintain high performance under real-world interference. Domain adaptation techniques have also been proposed, though they often require extensive annotated data.

E. Mosquito Audio Datasets

Several datasets support mosquito audio research:

- **HumBugDB** [5]: Over 20 species recorded in real-world conditions with background noise and variable quality.
- **Abuzz** [12]: Community-sourced smartphone recordings; large-scale but noisily labeled.
- **Curated sets** [4], [6]: Focused on 2–3 species in clean lab settings.

These datasets often lack recordings of overlapping species or swarms. Synthetic datasets have emerged to simulate multi-species scenarios and support mosquito counting tasks under controlled conditions.

F. Limitations in Existing Work

Despite progress, current systems face several limitations:

- Limited support for multi-species detection and mosquito counting;
- Few models optimized for embedded devices;
- Scarcity of datasets with realistic noise and swarming conditions.

G. Contribution of This Work

This study addresses the above gaps by proposing a dual-headed CNN model capable of:

- Scalable training with a synthetic dataset simulating swarms;
- Multi-species detection;
- Efficient execution on resource-constrained embedded devices;
- Robustness to various noise conditions via augmentation.

Our design builds upon prior work [4], [6], [11], while introducing novel contributions in architecture design, data synthesis, and deployment strategy.

III. Problem Statement

There is an urgent need for surveillance tools that can estimate mosquito density and distinguish vector species in real time under field conditions. While audio-based detection systems show promise, they are often fragile in noisy environments or when encountering multiple overlapping mosquito sounds.

Additionally, deep learning models used for acoustic insect classification are typically too resource-intensive for low-power embedded devices. Most existing systems are designed for presence detection only, lacking the capability to handle species-level classification from real-world, noisy swarm recordings.

IV. Methodology

This section outlines the methodology for mosquito sound classification and counting. It includes audio preprocessing, synthetic swarm generation, model architectures, training, and evaluation.

A. Audio Preprocessing and Feature Extraction

Raw audio signals are sampled at 16 kHz and normalized. To convert 1D waveforms into deep learning-compatible formats, multiple signal transforms are considered:

1) Fast Fourier Transform (FFT): The FFT is used for global spectral analysis:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}, \quad k = 0, 1, ..., N-1 \tag{1}$$

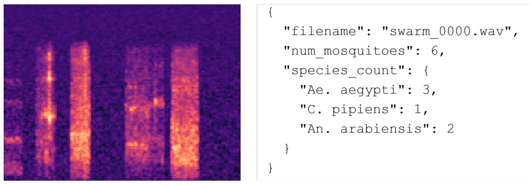
While useful for identifying dominant frequencies, FFT lacks temporal localization.

2) Short-Time Fourier Transform (STFT): To handle non-stationary wingbeat signals, STFT provides time-frequency representations:

$$X(t,\omega) = \int_{-\infty}^{\infty} x(\tau)w(\tau - t)e^{-j\omega\tau}d\tau \qquad (2)$$

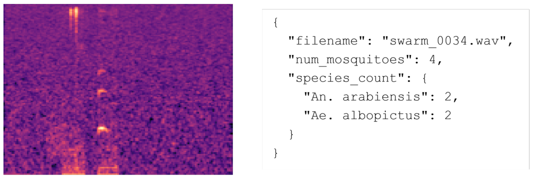
where $w(\cdot)$ is a window function. STFT forms the basis for spectrogram computation.

In this study, 64-bin log-mel spectrograms computed via STFT are used. Spectrograms are saved as RGB images for compatibility with visual backbones. Noise augmentation improves robustness. To prepare the acoustic features for classification, all mosquito swarm recordings were resampled to a target sampling rate of 16 kHz and then transformed into two alternative time–frequency representations. In the short-time Fourier transform branch, mel spectrograms were computed using a 512-point FFT with a 25 ms Hann window and a 10 ms hop size. Each spectrogram contained 128 mel bands, expressed in decibel scale relative to the signal maximum, and subsequently rendered as an image of 224 × 224 pixels.



```
{
  "filename": "swarm_0000.wav",
  "num_mosquitoes": 6,
  "species_count": {
    "Ae. aegypti": 3,
    "C. pipiens": 1,
    "An. arabiensis": 2
  }
}
```

Fig. 2: Mel-spectrogram of a synthetic mosquito swarm. Vertical lines indicate wingbeat frequencies.



```
{
  "filename": "swarm_0034.wav",
  "num_mosquitoes": 4,
  "species_count": {
    "An. arabiensis": 2,
    "Ae. albopictus": 2
  }
}
```

Fig. 3: Example input with corresponding species label metadata. Used for generating multi-label targets.

B. Synthetic Swarm Generation

Let $\mathcal{S} = s_1, ..., s_K$ denote mosquito species. For each synthetic sample, $n \sim \mathcal{U}(1, 10)$ mosquitoes are selected:

$$X(t) = \sum_{i=1}^{n} g_i \cdot x_i(t - \tau_i) \cdot \mathbf{1}[0,T](t - \tau_i) \qquad (3)$$

where $g_i \sim \mathcal{U}(0.2, 1.0)$ is gain and $\tau_i \sim \mathcal{U}(0, 3.0)$ is time offset. Segments $x_i(t)$ are randomly chunked ($\Delta t \in [0.3, 0.6]$s). White Gaussian noise is added:

$$Xnoisy(t) = X(t) + \epsilon(t), \quad \text{SNR} \in [20, 40] \text{ dB} \qquad (4)$$

Multi-label ground truth vectors $y \in 0, 1^K$ indicate species presence.

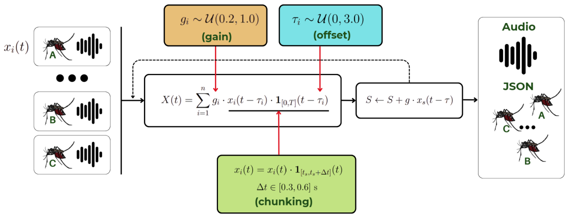

$x_i(t)$
A
B
C
$g_i \sim \mathcal{U}(0.2, 1.0)$
(gain)
$\tau_i \sim \mathcal{U}(0, 3.0)$
(offset)
$X(t) = \sum_{i=1}^{n} g_i \cdot x_i(t - \tau_i) \cdot \mathbf{1}_{[0,T]}(t - \tau_i)$
$S \leftarrow S + g \cdot x_s(t - \tau)$
$x_i(t) = x_i(t) \cdot \mathbf{1}_{[t_s, t_s+\Delta t]}(t)$
$\Delta t \in [0.3, 0.6]$ s
(chunking)
Audio
JSON
A
C
B

Fig. 4: Synthetic swarm generation pipeline.

As shown in Figure 4, swarm synthesis models density, overlap, and timing to create realistic mixtures that support multi-label learning and better generalization. To create the synthetic mosquito swarm dataset, individual mosquito recordings were combined using a carefully controlled strategy that mimicked real-world swarm conditions. Key steps included randomly selecting 3–7 mosquito audio chunks (each 0.3–0.6s) from up to 3 species per sample, applying gain variation (0.2–1.0) to simulate distance, and introducing random time offsets (0–3s) to spread audio events across a 5-second window. This approach ensured biological plausibility and reduced signal artifacts like clipping or redundancy. Additionally, white Gaussian noise was added at varying SNR levels (20–40 dB) to simulate realistic trap environments. Each sample was annotated with species counts and total mosquito numbers for multi-label training. These refinements significantly improved the realism, diversity, and balance of the dataset, enabling more effective and generalizable model training.

C. Dataset

We use the Abuzz dataset [12], which contains labeled mosquito wingbeat recordings across 20 species. For this study, we categorize them into four groups:

- **Aedes**
- **Anopheles**
- **Culex**
- **Noise** (non-mosquito)

In this study, we focus on Aedes aegypti, Aedes albopictus, Anopheles arabiensis, Anopheles gambiae, Culex quinquefasciatus, Culex pipiens.

D. Model Architectures

Three neural network architectures are evaluated for multi-label classification:

1) CNN (ResNet-18): Baseline architecture using ResNet-18 to process 224 × 224 RGB spectrograms:

$$\hat{y} = \sigma(W^\top f + b), \quad f = \text{GAP}(\text{CNN}(X)) \qquad (5)$$

Here, $f \in \mathbb{R}^{B \times C}$ is the global average pooled feature representation, and B is the batch size. Pros: efficient and edge-friendly. Cons: no temporal modeling.

$$\text{Input Spectrogram } (224 \times 224) \xrightarrow{\text{CNN (ResNet-18)}} \mathbb{R}^{B \times C} \xrightarrow{\text{FC}} \hat{y}$$

Fig. 5: CNN baseline architecture.

2) CNN + RNN: CNN features $F \in \mathbb{R}^{B \times C \times H \times W}$ are reshaped and passed to an RNN:

$$S = \text{reshape}(F) \in \mathbb{R}^{B \times T \times C}, \quad T = H \cdot W \tag{6}$$

$$R = \text{RNN}(S) \in \mathbb{R}^{B \times d} \tag{7}$$

$$\hat{y} = \sigma(W^\top R + b) \tag{8}$$

Here, S is the sequence embedding of spectrogram patches, and R is the hidden state at the final step. This hybrid improves temporal awareness, but can suffer from vanishing gradients.

$$\text{Input Spectrogram } (224 \times 224) \xrightarrow{\text{CNN (ResNet-18)}} F \in \mathbb{R}^{B \times C \times H \times W} \xrightarrow{\text{reshape}} \mathbb{R}^{B \times T \times C} \xrightarrow{\text{RNN}} h_T \xrightarrow{\text{FC}} \hat{y}$$

Fig. 6: CNN + RNN architecture.

3) CNN + LSTM: Replacing the RNN with an LSTM improves long-range dependency modeling:

$$(h_t, c_t) = \text{LSTM}(S) \tag{9}$$

$$\hat{y} = \sigma(W^\top h_T + b) \tag{10}$$

$$\text{Input Spectrogram } (224 \times 224) \xrightarrow{\text{CNN (ResNet-18)}} F \in \mathbb{R}^{B \times C \times H \times W} \xrightarrow{\text{reshape}} \mathbb{R}^{B \times T \times C} \xrightarrow{\text{LSTM}} h_T \xrightarrow{\text{FC}} \hat{y}$$

Fig. 7: CNN + LSTM architecture.

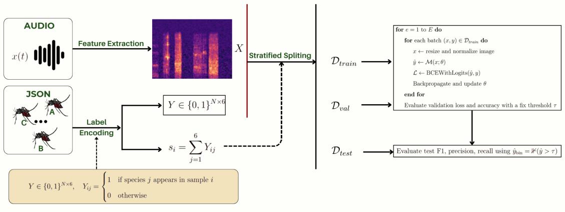
Here, h_T is the final hidden state capturing temporal dependencies across the sequence. This hybrid achieves stronger modeling of long-range wingbeat patterns compared to vanilla RNNs.

E. Training Pipeline

The training procedure includes spectrogram conversion, label encoding, stratified data splitting, and end-to-end model optimization:

- BCE loss with sigmoid activation
- Adam optimizer ($\alpha = 10^{-4}$)
- Early stopping based on validation loss

Predictions $\hat{y} \in [0, 1]^K$ are thresholded at different τ for evaluation.


AUDIO
$x(t)$
Feature Extraction
X
Stratified Spliting
JSON
A
C
B
Label Encoding
$Y \in \{0,1\}^{N\times6}$
$s_i = \sum_{j=1}^{6} Y_{ij}$
$Y \in \{0,1\}^{N\times6}, \quad Y_{ij} = \begin{cases} 1 & \text{if species } j \text{ appears in sample } i \\ 0 & \text{otherwise} \end{cases}$
$\mathcal{D}_{train}$
$\mathcal{D}_{val}$
$\mathcal{D}_{test}$

```
for $e = 1$ to $E$ do
    for each batch $(x, y) \in \mathcal{D}_{train}$ do
        $x \leftarrow$ resize and normalize image
        $\hat{y} \leftarrow \mathcal{M}(x; \theta)$
        $\mathcal{L} \leftarrow \text{BCEWithLogits}(\hat{y}, y)$
        Backpropagate and update $\theta$
    end for
    Evaluate validation loss and accuracy with a fix threshold $\tau$
Evaluate test F1, precision, recall using $\hat{y}_{\text{bin}} = \nVdash(\hat{y} > \tau)$
```

Fig. 8: End-to-end learning pipeline for mosquito swarm classification.

Algorithm 1 An overview of the end-to-end training pipeline for multi-label mosquito species classification from swarm audio.

```
procedure TrainSwarmClassifier($\mathcal{D}$, Model $\mathcal{M}$, Epochs $E$)
    Convert swarm audio $x(t)$ to STFT spectrogram $X$
    Compute binary label matrix $Y \in \{0, 1\}^{N \times 6}$ from species counts
    Compute label cardinality $s_i \leftarrow \sum_j Y_{ij}$
    $\mathcal{D}_{train}, \mathcal{D}_{val}, \mathcal{D}_{test} \leftarrow$ stratified split on $s_i$
    Load backbone CNN $\phi$ (e.g., ResNet-18), attach classifier head, or RNN, or LSTM
    Initialize optimizer Adam($\theta$) and loss function BCE
    for $e = 1$ to $E$ do
        for each batch $(x, y) \in \mathcal{D}_{train}$ do
            $x \leftarrow$ resize and normalize image
            $\hat{y} \leftarrow \mathcal{M}(x; \theta)$
            $\mathcal{L} \leftarrow \text{BCEWithLogits}(\hat{y}, y)$
            Backpropagate and update $\theta$
        Evaluate validation loss and accuracy with a fix threshold $\tau$
    Evaluate test F1, precision, recall using $\hat{y}_{\text{bin}} = \nVdash(\hat{y} > \tau)$
```

F. Evaluation Metrics

We use the following multi-label metrics:

- **Multi-label Accuracy**: The average proportion of correctly predicted labels per sample, across all classes. For prediction $\hat{y} \in \{0, 1\}^C$ and true label $y \in \{0, 1\}^C$, accuracy per sample is: $\frac{1}{C} \sum_j \nVdash(y_j = \hat{y}_j)$
- **Macro Precision, Recall, F1-score**

These metrics highlight per-class performance and robustness across imbalanced data.

V. Experimental Results

This section presents the results of experiments conducted on the multi-label mosquito species classification task using CNN+LSTM, CNN+RNN, and CNN models trained on swarm audio spectrograms. We evaluate performance under different decision thresholds (0.3, 0.5, and 0.7), focusing on the trade-off between sensitivity (recall) and precision, particularly for embedded real-time applications.

A. Impact of Threshold on Model Performance

Figure 9 compares the performance of CNN+LSTM, CNN+RNN, and CNN across three thresholds using four metrics: Accuracy, Macro F1, Precision, and Recall. The CNN+LSTM consistently outperforms the others, particularly in F1 and recall, confirming its ability to capture long-term audio patterns.

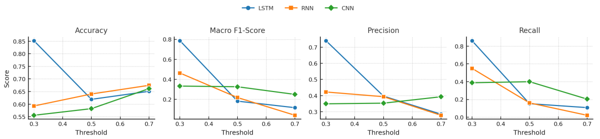
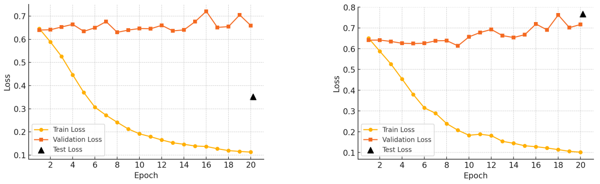


Fig. 9: Performance comparison of CNN+LSTM, CNN+RNN, and CNN models across thresholds 0.3, 0.5, and 0.7. Metrics include Accuracy, Macro F1-Score, Precision, and Recall. CNN+LSTM performs best overall, especially in recall and F1 at lower thresholds.

At threshold 0.3, the CNN+LSTM achieves its highest F1 and recall, indicating strong sensitivity to mosquito species presence. However, this comes at the cost of reduced precision due to more liberal predictions. As the threshold increases to 0.5 and 0.7, all models show a decrease in recall and F1 with minimal gains in precision—highlighting the classic trade-off in multilabel classification between detection sensitivity and false positives.

B. Learning Curve Analysis

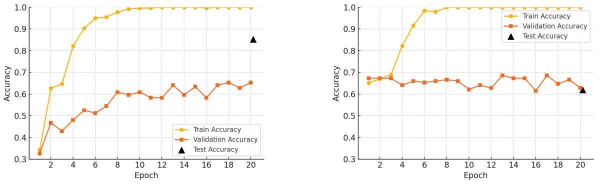


(a) Loss curves at threshold 0.3

(b) Loss curves at threshold 0.5

Fig. 10: Training and validation loss of CNN+LSTM model under different thresholds. Triangle markers denote final test loss.

Threshold 0.3 – Stable Generalization: Figure 10(a) shows smooth and consistent convergence in both training and validation losses, suggesting good generalization. The model avoids overfitting, and its final test performance (triangle marker) confirms strong robustness.



(a) Accuracy at threshold 0.3

(b) Accuracy at threshold 0.5

Fig. 11: Training and validation accuracy of CNN+LSTM under different thresholds. Triangle markers indicate test accuracy.

In Figure 11(a), validation accuracy continues to rise after training accuracy saturates—evidence of well-balanced learning. This supports using threshold 0.3 for applications where missing a mosquito is more costly than a false positive, such as in early-warning traps.

Threshold 0.5 – Overconfidence and Reduced Sensitivity: In contrast, Figure 10(b) shows early convergence in training but unstable validation loss, indicating reduced generalization. Figure 11(b) further shows plateaued validation accuracy. The final test performance drops compared to the 0.3 case.

This suggests that threshold 0.5 leads to a model that is overconfident yet less sensitive to minority classes—an undesired outcome for imbalanced multilabel detection tasks.

C. Field Validation in Vung Tau, Vietnam

TABLE I: Summary of mosquito species detections in Vung Tau, Vietnam.

Species	**# Positive Detections (out of 12)**
Aedes aegypti	10
Anopheles arabiensis	8
Aedes albopictus	6
Culex quinquefasciatus	1
Anopheles gambiae	0
Culex pipiens	1

To evaluate real-world performance, we conducted a field validation study in Vung Tau, Vietnam, where swarm audio data was collected using a baited mosquito trap equipped with CO$_2$ as an attractant to simulate human presence. The trap was designed to encourage natural mosquito swarming behavior while minimizing environmental noise. A high-sensitivity microphone was positioned near the trap to capture continuous flight-tone recordings of approaching and hovering mosquitoes. These real swarm audio samples were then processed using our trained CNN+LSTM detection model, demonstrating the best performance in prior evaluations. During inference, a conservative detection threshold of 0.3 was applied to the model's output probabilities to balance sensitivity and specificity, particularly under field conditions. Based on this setup, the following results summarize the model's performance across six target mosquito species.

Table I summarizes detection outcomes from field recordings. There are 12 swarm audios in total. The model produced strong detections for *Ae. aegypti* (10/12) and *An. arabiensis* (8/12). *Ae. albopictus* was detected in half of the samples (6/12) with moderate confidence (35%). By contrast, *C. quinquefasciatus* and *C. pipiens* rarely exceeded the detection threshold (1/12 each), while *An. gambiae* was never detected (0/12).

Considering the context, the dominant detections of *Ae. aegypti* and moderate signals of *Ae. albopictus* align with their known distribution in southern Vietnam, including Vung Tau. By contrast, the frequent positive detections of *An. arabiensis* are false, as this African vector does not occur in Vietnam. Meanwhile, *An. gambiae* was consistently absent, which is geographically correct, although the discrepancy with its sister species suggests wingbeat misclassification. *C. quinquefasciatus* is common in Vietnam, and its scant recognition indicates model under-sensitivity. Finally, *C. pipiens* holds little public health significance in the region, making its low detection rate acceptable.

VI. Conclusion

This paper proposed an end-to-end audio-based system for real-time mosquito species detection and classification, designed for deployment in resource-constrained smart traps. The pipeline integrates synthetic swarm audio generation, spectrogram-based feature extraction, deep learning models, and thresholded multi-label classification to achieve robust species-level prediction in noisy, multi-mosquito environments.

In contrast to simple augmentation methods such as time-shifting and pitch-scaling, swarm synthesis functions at a higher level, simulating overlapping organism interactions as well as changing swarm density. This supports multi-label learning, which helps the model adapt better to actual swarm behaviours. Moreover, the synthetic swarm dataset is a key innovation, as it offers a scalable and reproducible resource for surveillance research, thus minimizing the reliance on expensive field data collection.

We explored multiple architectures, including a lightweight CNN, a CNN+RNN hybrid, and a CNN+LSTM model. Among these, the CNN+LSTM consistently achieved the best classification performance, particularly in terms of macro F1-score and recall, highlighting its strength in capturing temporal dynamics.

Thresholding played a key role in balancing recall and precision. At $\tau = 0.3$, high recall was achieved, suitable for surveillance scenarios where missing a species is costlier than a false alarm. Higher thresholds (e.g., $\tau = 0.7$) yielded better precision but degraded recall, reinforcing the need for application-specific tuning.

We also introduced a stratified label cardinality-based data split to improve training balance. The system's performance across six mosquito species showed both strengths and trade-offs.

Future Work

Future research directions include:

- Integrating attention mechanisms to improve spectral focus.
- Applying pruning and quantization for CNN+LSTM deployment on ultra-low-power hardware.
- Enhancing the swarm generator to simulate complex acoustic conditions.
- Exploring contrastive/self-supervised learning to reduce dependence on labeled data.
- Implementing adaptive, context-aware thresholding strategies.

This work demonstrates that lightweight, real-time mosquito classification via audio is both feasible and practical, offering a scalable foundation for vector surveillance in low-resource environments.

Acknowledgment

I would like to express my sincere gratitude to Dr. Nguyen Tuan Cuong and Dr. Vo Bich Hien for their invaluable guidance, insightful feedback, and unwavering support throughout the course of this thesis. Their expertise has been instrumental in shaping the research direction, addressing technical challenges, and refining the overall quality of this work. I am deeply appreciative of their mentorship, encouragement, and dedication to my academic development.

References

[1] B. Breedlove, "Deadly, dangerous, and decorative creatures," *Emerging Infectious Diseases*, vol. 28, no. 2, pp. 495–496, 2022.

[2] S. Boyer, E. Calvez, T. Chouin-Carneiro, D. Diallo, and A.-B. Failloux, "An overview of mosquito vectors of zika virus," *Microbes and Infection*, vol. 20, no. 11-12, pp. 646–660, 2018. [Online]. Available: www.sciencedirect.com/science/article/pii/S128645791830039X

[3] T. Bale, S. Belew, N. Williams, B. Johnson, T. Hall, N. Buckerfield, A. Kaul, M. H. Imtiaz, and B. Taylor, "Acoustic-based classification of mosquitoes using flight tones: A survey," *Acoustics Australia*, vol. 47, no. 2, pp. 191–199, 2019.

[4] I. Kiskin, V. Kindratenko, and S. J. Cox, "Bioacoustic detection with deep learning," *Interspeech*, 2019.

[5] MindFoundry, "The humbug project," 2022, https://www.mindfoundry.ai/blog/humbug-2022.

[6] H. Fernandes and P. Batista, "Mosquito detection using cnn on smartphone audio," *arXiv preprint arXiv:2008.09024*, 2020.

[7] K. Ramos, M. L. C. Guico, and J. K. A. Galicia, "Acoustic sensor module for mosquito detection and classification," in *2023 9th International Conference on Computer and Communication Engineering (ICCCE)*. IEEE, 2023, pp. 126–131.

[8] M. Yang and L. Zhang, "Insect wingbeat classification using cwt," *Sensors*, vol. 19, no. 5, p. 1123, 2019.

[9] Y. Wang, T. Li, and X. Chen, "Crnn for mosquito species identification," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 900–904, 2020.

[10] R. Toledo, R. da Silva, and J. Souza, "Lstm-based mosquito genus classification using their wingbeat sound," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 3, pp. 1–20, 2021.

[11] A. Supratak, P. Rattanatamrong, and S. Chomphan, "Mosquitosong+: Noise-robust mosquito classification in real-world environments," *PLOS ONE*, vol. 19, no. 2, p. e0310121, 2024.

[12] A. Project, "Citizen science to track mosquitoes using smartphones," 2018, https://abuzz.stanford.edu/.