MOSWING: A NOISE-ROBUST MOSQUITO WINGBEAT DETECTION MODEL

มอส-วิง: โมเดลในการตรวจจับเสียงการกระพือปีกของยุงให้ต้านทานต่อเสียงรบกวน

BY

MR.	PHURIWAT	ANGKOONDITTAPHONG	6388003
MS.	NAPAHATAI	SITIRIT	6388102
MR.	DANAIDECH	ARDSAMAI	6388118

ADVISOR
DR. AKARA SUPATAK

**A Senior Project Submitted in Partial Fulfillment of
the Requirement for**

**THE DEGREE OF BACHELOR OF SCIENCE
(INFORMATION AND COMMUNICATION TECHNOLOGY)**

**Faculty of Information and Communication Technology
Mahidol University
2023**

ACKNOWLEDGEMENTS

This paper cannot be completed without the help and devotion from many people involved. The authors would like to express gratitude for the following people, whose efforts and assistance have made this project possible:

First, Asst. Prof. Dr. Akara Supratak, the advisor of this project, for his time and efforts in giving practical comments and guiding the direction of this paper. The authors also wish a happy childhood for his newborn son.

Second, MIRU lab and its members, Prof. Dr. Peter Haddawy and Dr. Myat Su Yin, for providing valuable datasets and giving insightful comments in every meeting.

Third, the faculty of ICT, for providing necessary computing resources, workspace, funds, and thoughtful feedbacks from the members of faculty.

Next, the family and friends of the authors, whose support and encouragement have allowed the project to go smoothly.

Special thanks are given to all nearby tea shops around the university, who provided sufficient caffeine intake throughout the entirety of this project.

Mr. Phuriwat Angkoondittaphong
Ms. Napahatai Sitirit
Mr. Danaidech Ardsamai

MOSWING: A NOISE-ROBUST MOSQUITO WINGBEAT DETECTION MODEL

MR. PHURIWAT	ANGKOONDITTAPHONG	6388003 ITCS/B
MS. NAPAHATAI	SITIRIT	6388102 ITCS/B
MR. DANAIDECH	ARDSAMAI	6388118 ITCS/B

B.Sc. (INFORMATION AND COMMUNICATION TECHNOLOGY)

PROJECT ADVISOR: DR. AKARA SUPATAK

PROJECT CO-ADVISOR:

ABSTRACT

Information on the mosquito population in each area is needed to combat life-threatening diseases from mosquito vectors. However, the traditional approach of manual counting is labor-intensive and a slow process to assert the population manually. There are alternative approaches in various ways, such as detecting mosquitoes from Wingbeat's fundamental frequency or even deep learning classifier. Nevertheless, most of them use only lab-recorded mosquito wingbeat sounds, which is less applicable to realistic operations.

In this paper, the authors developed noise-robust sound event detection models for mosquito species and sex, which can be used for automated counting. To create the environmental with mosquito present dataset, the authors overlay mosquito wingbeat sound on the recorded environmental noise with a difference gain factor. The dataset is then trained with a 1DCNN model with RNN to represent time-series significance. The result demonstrated that the proposed model significantly exceeded the baseline detection performance, achieving a 0.877 F1 score on habitat A and 0.936 on habitat B. Regarding classification, the proposed model overcame the baseline in habitat B while having lower performance on habitat A.

KEYWORDS: SOUND EVENT DETECTION /MOSQUITO CLASSIFICATION

37 P.

มอส-วิง: โมเดลในการตรวจจับเสียงการกระพือปีกของยุงให้ต้านทานต่อเสียงรบกวน

นาย ภูริวัจน์	อังกูรดิษฐพงศ์	6388003 ITCS/B
นางสาว นภหทัย	สิทธิฤทธิ์	6388102 ITCS/B
นาย ดนัยเดช	อาจสมัย	6388118 ITCS/B

วท.บ. (เทคโนโลยีสารสนเทศและการสื่อสาร)

อาจารย์ที่ปรึกษาโครงการ: ดร. อัคร สุประทักษ์
อาจารย์ที่ปรึกษาร่วมโครงการ:

บทคัดย่อ

ข้อมูลประชากรของยุงในพื้นที่เป็นสิ่งสำคัญมากในการหาแนวทางการป้องกันโรคระบาดร้ายแรงที่มียุงเป็นพาหะนำโรค แต่วิธีดั้งเดิมในการนับจำนวนยุงนั้นใช้เวลาและทรัพยากรมาก ยังมีวิธีอื่นอีกที่ช่วยประมาณประชากรยุงได้ดีกว่า เช่น การตรวจจับยุงด้วยคลื่นความถี่เสียง หรือการใช้เครื่องมือจำแนกด้วยการเรียนรู้เชิงลึกในการแยกประเภทยุง อย่างไรก็ตาม วิธีการส่วนมากนั้นใช้เสียงยุงจากสิ่งแวดล้อมควบคุมที่ไม่มีเสียงรบกวน เมื่อนำไปใช้จริงแล้วจึงใช้การได้ไม่ดีนัก

ในวิทยานิพนธ์นี้ ผู้เขียนได้พัฒนา sound event detection model ที่สามารถจำแนกพันธุ์และเพศของยุงได้ดีบนเสียงรบกวน โดยสามารถนำไปใช้ในการประมาณประชากรยุงในพื้นที่ได้ ผู้เขียนใช้ dataset ที่สร้างขึ้นมาโดยการรวมเอาเสียงรบกวนและเสียงยุงเข้าด้วยกัน dataset นี้มีการแบ่งเป็น 2 ส่วนได้แก่ ถิ่นที่อยู่ ก และถิ่นที่อยู่ ข เพื่อจำลองยุงแต่ละสปีชีส์ที่มักอาศัยอยู่ร่วมกันในสภาพแวดล้อมจริง หลังจากนั้นจึงนำข้อมูลไปใช้ในการสร้างเครื่องมือจำแนกด้วยการเรียนรู้เชิงลึก โดยใช้เครื่องมือ 1DCRNN และมีโครงสร้าง RNN ในการประมวลผลข้อมูลแบบเวลา ผลลัพธ์แสดงให้เห็นว่าเครื่องมือนี้ทำงานได้ดีกว่าเครื่องมือเดิมมากเมื่อใช้ในการตรวจจับเสียงยุง โดยได้รับ F1 score มากถึง 0.877 บนถิ่นที่อยู่ ก และ 0.936 บนถิ่นที่อยู่ ข ส่วนผลลัพธ์ด้านการจำแนกแต่ละเพศและสปีชีส์ของยุงแสดงให้เห็นว่า เครื่องมือเดิมทำงานได้ดีกว่าเมื่ออยู่บนถิ่นที่อยู่ ก แต่เครื่องมือที่พัฒนาใหม่สามารถทำงานได้ดีกว่าในถิ่นที่อยู่ ข

37 หน้า

CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Motivation

Vector-borne diseases such as Dengue, Chikungunya, and Yellow fever are all fatal diseases transmitted by mosquitoes. It poses an essential threat to public health in tropical countries [2], especially in Thailand, where the mosquito is highly populated. It is an estimated trend that the impact of these diseases will increase with time from the changing climate and urbanization [3]. Public health interventions targeting these diseases are implemented within each country to monitor the efficacy of vector control initiatives. Accurate information about mosquito vector population density is necessary.

To lessen the impact of mosquito-borne diseases, public health regulators have been trying to gather more information about the mosquito population and find ways to control it efficiently. The most traditional method includes installing traps in various locations and then manually counting and identifying the species and sex of the caught mosquitoes. The common traps used to catch mosquitoes often use light, heat, odour, and blood to attract mosquitoes into the deployed area [4]. However, this method is highly costly and labour-intensive. Each instalment of the trap costs more money to buy equipment, more people to manually install it, and more experts to classify the mosquito types. This process also has scalability limitations that hinder large-scale monitoring efforts and can be a bottleneck in timely and effective vector control. Therefore, other methods have been developed to identify mosquito types more efficiently.

It is known that mosquitoes of different species and sexes often differ in their wingbeat audio signatures [5] such as frequency. Therefore, many methods of differentiating mosquito sex and species by their wingbeat sounds have been developed widely. For example, [6] use the wingbeat frequency, [7] use MFCC, and [8] use mosquito antenna characteristics to extract features of each mosquito types. Nevertheless, following the suggestion that differentiating via frequency may not be compelling enough [9] [10], recent attempts often extract features from the spectrogram of wingbeat sounds, then use deep learning models to utilize the image classification framework for spectrogram processing [1] [11]. This process of utilizing spectrogram in image classification is also widely used in classifying other animal species, such as birds and insects. However, [12] and [13] recently suggested that spectrograms may not be able to convey some critical differential features, so new methods of using raw wingbeat sound are developed to overcome this problem [14] [10].

However, accurately estimating mosquito populations and individual species in the field is compounded by extraneous environmental noise, such as traffic, human activity, and other insect sounds, which can obfuscate the distinct wingbeat frequencies of different mosquito species. [15] shows that using frequencies to differentiate mosquito types could not perform well given the noisy environment. This creates a substantial challenge for current surveillance techniques, which must be capable of isolating the unique acoustic signatures of mosquitoes from background noises. Some studies have developed a model to address this noisy environment problem in mosquito sounds classification noise [1] [16]. However, the models must perform better or can only do detection tasks. Other research focuses on using optical sensors to synthesize wingbeat sounds from light fluctuations [17], but this method works well only on sex classification, not species classification.

In order to detect and classify mosquito species and sex, including background noises into training data for the model to learn is one promising solution. It also has been proven to give impressive results in bird classification [18]. Even so, retrieving noisy mosquito sounds from natural environments is challenging. The traps must be deployed longer, and trained staff must manually listen to the long recordings to classify them, increasing the cost financially and prolonging the time. Another approach is synthesizing noisy mosquito sounds by overlaying mosquito sounds in a noisy environment, making the data collection process easier and faster. Several types of research show how data augmentation can reduce overfitting when the retrieved dataset is small and imbalanced [19], especially in a bioacoustic classification setting [20].

It is essential to obtain comprehensive estimates of mosquito population density and estimates involving individual species. Addressing this challenge is critical, as the advent of robust, noise-tolerant detection and classification systems would significantly advance entomological surveillance and, consequently, disease prevention.

Deep learning models addressing this problem are also developed separately between detectors and classifiers. Many researchers create a model to detect mosquito presence, then create another model to classify.

1.2 Problem Statement

The current deep-learning model for mosquito detection and classification needs to be more accurate due to noise pollution from the surrounding environment, ambient sounds of urban activity, traffic, and other wildlife, mainly insects. This brings up a critical issue: the current method needs to consider noise interference or evaluate its robustness against noise interference. This project aims to fill this critical gap by improving noise robustness.

Moreover, the typical practice of separating the tasks of mosquito detection and classification further exacerbates these issues. This dichotomy leads to a disjointed approach, where crucial insights into mosquito behaviour and population dynamics are lost, reducing the effectiveness of vector control. The current project addresses these shortcomings by integrating mosquito detection and classification into a unified process. By overcoming the limitations of traditional detection and classification, this approach aims to enable a more accurate and efficient estimation of mosquito populations.

1.3 Objective

To develop a deep learning model that identifies mosquito presence periods and classifies its species and sex from wingbeat sounds. The model should also be robust to noisy environments.

1.4 Scope of the project

1. This project will use the male and female sexes of five species of mosquitoes that are often found in Thailand: *Aedes aegypti, Aedes albopictus, Anopheles dirus, Anopheles minimus, and Culex quinquefasciatus*

2. This project will focus on the dataset from MIRU (Mahidol-Bremen Medical In-

formation Research Unit).

3. All recordings come from only specific microphone types, including Behringer ECM8000 and Primo EM172.

CHAPTER 2
BACKGROUND

2.1 Mosquito Wingbeat

The distinctive frequencies of mosquito wingbeats result from various biological and environmental factors. Each species has unique physiological traits such as wing size, body mass, and shape that influence the rate and pattern of their wing flapping. Males typically exhibit higher wingbeat frequencies due to their generally smaller body sizes when compared to females [21]. Environmental conditions like temperature and humidity can also modulate these frequencies. As such, the acoustic footprint of a mosquito's wingbeat is not merely a byproduct of its flight but a complex signature shaped by an interplay of intrinsic species-specific characteristics and external variables.

Mosquito Species	Sex	Age (Days)	Frequency (Hz)
Ae. Aegypti	Male	1 - 10	557 - 600
Ae. Albopictus	Male	4	724
An. Arabiensis	Male	2	703 ± 8.72
Ae. Aegypti	Female	1 - 10	414 - 453
Ae. Albopictus	Female	4	544
An. Arabiensis	Female	2	435 ± 4.88

Table 2.1: An analysis of mosquito wing beat frequencies, Table 1 in Sinka et al. (2021)[1] presents a detailed comparison of species-specific data under various environmental conditions [page 6].

2.2 Mosquito data collection

There are two main approaches for collecting the wingbeat sound, including an acoustic sensor and an optical sensor.

2.2.1 Acoustic sensor

An acoustic sensor is used by simply putting the mosquito in the trap and using a microphone to capture the acoustic footprint of the mosquito when it flies. However, there are obvious downsides to this simple method. The noise of the trap is also included in the recordings and might affect the system one has been researching. This method has been adapted and developed in various ways to be fed as training data into the model, such as transforming the raw signal into spectrogram [16] [1] and applying wavelet transform [22].

Yin et al. [21] collect mosquito wingbeat sound by putting each mosquito into a small cylindrical container covered by a net, then record its sound with a Studio Behringer (Primo EM172) in mono channel, 24 bits depth with 96 kHz sampling rate.

[6] introduce the system named LOCOMOBIS (LOw-COst MOsquito BIoacoustic Sensor), which integrates a sensitive microphone and environmental sensors within its architecture. The recorded audio data, containing temperature and humidity readings, are processed to classify different species and genders of mosquitoes.

Mukundaraja et al.[23] demonstrates that even basic mobile phones can sensitively record acoustic data specific to mosquito wingbeat sound and critical metadata like time and location. The researchers highlight the ability of mobile phone microphones to record the wingbeat frequencies of a wide range of medically necessary mosquito species. This capability enables rapid, non-invasive species identification, substantially advancing field-based mosquito monitoring.

2.2.2 Optical sensor

Collecting mosquito wingbeat via optical sensor will retrieve data from the fluctuation of light when a mosquito fly passes, then create a pseudo wingbeat sound, so it is not affected by interrupting noises in the background. Chen et al.[9] using custom-built pseudo-acoustic optical sensors. This sensor applies the technique of a phototransistor array that is aligned with a laser line. When insects move using their wings, they detect light intensity changes. When mosquitoes flew through the laser beam, their wings partially blocked the light, creating fluctuations captured by the sensor.

This method breaks the limitations of traditional acoustic sensors, such as sensitivity to ambient noise and distance-related sound attenuation. Filtered and amplified electrical signals from the phototransistor array were recorded as six-hour MP3 files. Other optical field noises, such as reflected light, flying dust, or ambient, can still affect this method. This limitation makes the approach unreliable for classifying mosquito species [17].

2.3 Deep-learning

2.3.1 Deep-learning Layer

Convolution Layer

Inspired by image processing, it is a fundamental operation from image processing in which a kernel matrix ”convolute” on an image. The result from 2D convolution is versatile, ranging from blur sharpening to edge detection, depending on the kernel. It processes only the local feature of the image. The model might learn to set the kernel via a backpropagation algorithm in a deep-learning context. It is a central part of successful image-based deep-learning models such as VGG family [24], ResNet family [25], and EfficientNet family [26]. It also has usage in signal-related problems such as Polyphonic Sound Event Detection (SED) [27], [28], and [29].

It also has a one-dimensional counterpart, which is widely known in signal processing. The 1D convolution layer has many uses in speaker recognition [30], electrocardiogram classification [31], and weak labeled SED [10].

Recurrent Layer

A recurrent layer is a class of deep-learning layers specifically designed to handle sequential or time series data as it has the internal memory of the layer. Such layer has been used in many problems such as machine translation [32], and weather forecasting [33]. There are many types of recurrent layers, including Long Short Time Memory (LSTM) [34] and Gated recurrent unit (GRU) [35].

The combination of the convolution layer and recurrent layer has been used on SED task [27], [29] as they can interpret temporal information.

2.4 Spectrogram

A spectrogram is a visual representation of the spectral density of a signal over time. The use of two-dimensional visualization characterizes this analysis tool. The frequency is displayed on the vertical axis. Moreover, time is displayed on the horizontal axis. The third dimension represents the amplitude or power of signals at different frequencies. It is presented visually using different intensities or shades. Spectrograms are used primarily to analyze audio signals. This makes it possible to monitor dynamic changes in the frequency spectrum. It provides valuable insights into the temporal characteristics of various frequency components. Included in the signal. To create a spectrogram, Two commonly used steps are used:

figures/chap2/ex_spectrogram.png

Figure 2.1: Example of Spectrogram from Environmental Noise

2.4.1 Fast Fourier Transform (FFT)

In signal processing, Fast Fourier Transform[36] is an advanced algorithm for calculating Discrete Fourier Transform (DFT) with high accuracy. Complex coefficient calculations allow the signal to be efficiently decomposed into its frequency components.

2.4.2 Short-Time Fourier Transform (STFT)

Short-Time Fourier Transform[37] divides a time-domain signal into shorter overlapping segments, and for each segment, it calculates the Fourier Transform. This process generates a series of spectra representing the signal's frequency content at different time intervals.

2.5 Literature review

Achieving efficient remote mosquito surveillance remains difficult, considering many studies on the detection and classification of mosquito wingbeats. Advancing the problem areas is the development of noise-robust models, which, in diverse and unpredictable environmental noise conditions, make these models unreliable.

2.5.1 Combination between Detection and classification

A common approach uses a pipeline starting with detection and following by classification[21]. This approach effectively eliminates extraneous noise and sounds that can help reduce the classification computation resource. However, it operates under the often unmet assumption of flawless detection, which is difficult to achieve in real-world situations. Another research introduces a unique combination of pre-processing and deep-learning model [38] to build a mosquito detector from mosquito wingbeat sound background noises collected under different environmental conditions. The researcher found that the issues with traditional audio pre-processing techniques and methods, such as log-mel front-end processing, also have limitations, such as sensitivity to variations in incoming audio signals and the dynamic range of filterbank energy. These issues can affect the model's ability to detect mosquito wingbeats reliably.

Vasconcelos et al.[6] recorded the wingbeat sound of three mosquito species: *Aedes aegypti, Culex, and Culiseta* and used the Fast Fourier Transform (FFT) to process the audio data and identify the fundamental frequency of the wingbeat, which is a crucial feature for species and gender classification[9][10].

While the fundamental frequency is widely used and also considered to be the is an important component when classification how to differentiate between the kinds of mosquitoes[39], several studies have shown that it may not be sufficient to separate between mosquito species accurately. The introduction of non-mosquito classification[9] is an important study that addresses this issue.

However, the issue of balancing the classifier's computational load with the detector's sensitivity and crucial features in the continuous section is lost because the technique does not account for temporal information. The loss of essential features limits the applicability of the classification model and rejects the potential benefits of continuous temporal data in enhancing detection and classification accuracy. On the other hand, a significant challenge that needs to be addressed is that insect flight sounds are often sparse and discontinuous in recording due to their insect nature.

This research gap highlights whether combining the detection and classification into an end-to-end model may be more useful and efficient. This model addresses the need to balance between detection sensitivity and the need to calculate and manipulate continuous temporal data for detection and classification purposes.

Consequently, our project aims to pioneer the development of an end-to-end deep-learning model involving spatial and temporal data extracted from raw sound. This technique is in fulfillment of the priciples of SED[29] [40] [27] [41] [42]. Therefore, applying SED to the model can remain unexplored. So, this project seeks to fill this gap by adapting the SED technique for mosquito wingbeat sound analysis challenges.

2.5.2 Robust to noisy environments

[38] introduces random augmentations on the audio samples to help the model learn to distinguish mosquito buzz from various sounds. This includes using external noise sources to train the model, which increases its robustness in real-world scenarios where background noise is present. Recently, work [43] study uses noise simulations for model evaluation, and the challenge remains in accurately simulating the varied and unpredictable real-world environmental noise conditions. Two primary approaches have merged to enhance the robustness of mosquito wingbeat detection and classification of mosquito wingbeat sounds in noisy environments. The first involves converting raw wingbeat audio to spectrograms by using feature extraction such as MFCC [6] and fundamental frequency [7] and Now recently, followed by employed deep-learning model such as 1DCNN [43], DenseNet121 [44]

Another approach is data augmentation, which adds more data to the model by adding noise from the surroundings during the model training process, and it worked well for classifying birds [12]. The researcher applied several data augmentation methods, such as temporal shifting and the addition of low-amplitude noise. The researcher mentioned the limitation of the Mel frequency scale and the neglect of phase or temporal fine structure, which can be crucial for specific bioacoustic tasks. Another existing work [38] used a different data augmentation technique, which involves randomly warping blocks of frequency channels and time steps in the audio recordings. This technique helps the model more robust in separating auditory elements and noise conditions.

However, the precision of these noise augmentation approaches for mosquito species identification still needs to be explored. To address this gap, this project aims to develop a method for training deep-learning models to detect and classify mosquito species and sex from raw wingbeat sound.

CHAPTER 3
METHODOLOGY

This chapter explained the details of the experiment's implementation, including data preparation, noise simulation, problem formulation, model architecture, and training.

Figure 3.1 illustrated the overall methodology of this project. Given the dataset from MIRU lab and Humbug, the authors simulated mosquito sounds on noises by overlaying wingbeat sounds on environment sounds. The resulting dataset is used to train the polyphonic sound event detection model.

figures/chap3/methodology.pdf

Figure 3.1: Overall of methodology

3.1 Problem Formulation

Consider the problem to be segment-based Polyphonic SED. An event means mosquito(es) fly near the microphone to make the sound output audible. The dataset $D = \{(x_i, y_i)\}_{i=0}^{N}$ is a collection of the overlayed environmental sounds $x_i \in \mathbb{X}$ with an event recording matrix $y_i \in \mathbb{Y}$, where $\mathbb{X} = [-1, 1]^{n \times c}$ is a sound with $n = sr \times t$ data points, sampling rate sr, duration of t seconds, and c number of audio channels. The labels are $\mathbb{Y} = \{0, 1\}^{s \times n_{classes}}$, a one-hot matrix encode events, where $s = \frac{t}{t_{seg}}$ number of segments, t_{seg} duration (in seconds) per segment and n_{cls} number of classes or type of events. If an event of class cls presents at a segment s, then $y[s, cls] = 1$ and set to 0 if an event is absent. This is also a multi-label problem, meaning the events from different classes may be present simultaneously.

Models will receive input $x \in \mathbb{X}$, representing the sound with or without the event present. It outputs predictions as a matrix $\hat{y} \in [0, 1]^{s \times n_{cls}}$ whose values are a probability of event presence in a specific class and time frame. Then, a threshold $thres$ separates the event presence and absence. If the probability is less than the $thres$, it is classified as absence. If the probability is more than the $thres$, it is classified as presence. Since the model may exhibit different outcomes across different classes, the $thres$ can be adjusted to maximize performance in each class.

3.2 Noise Simulation

Given an environmental sound $env \in [-1, 1]^{t_{env}}$, a mosquito wingbeat sound $mos \in [-1, 1]^{t_{mos}}$ of class cls at sampling rate sr, a time when the mosquito start on environmental sound t_{start}, a normalize function $\text{norm} : \mathbb{R}^n \to [-1, 1]^n$ and a factor to control the proportion of the wingbeat amplitude $G \in (0, 1]$. Assume $t_{mos} \leq t_{env}$, the overlayed environmental sound $x \in \mathbb{X}$ can be calculated using the equation 3.1.

$$x = \text{norm}(w) \tag{3.1}$$

$$w[t] = \begin{cases} env[t] + G \cdot \text{norm}(mos)[t - t_{start}], & t_{start} \leq t < t_{start} + t_{mos}. \\ env[t], & \text{otherwise.} \end{cases} \tag{3.2}$$

where j is ranging from 0 upto t_{mos}.

Then, the event is encoded in the one-hot matrix y using the equation 3.3.

$$y[t, cls] = \begin{cases} 1, & t_{start} \leq t < t_{start} + t_{mos}. \\ 0, & \text{otherwise.} \end{cases} \tag{3.3}$$

To create a dataset, The authors create a collection of the overlayed environmental sounds $D = \{(x_i, y_i)\}_{i=0}^{N}$. The x_i refers to the i-th overlayed environmental sound, and

y_i i-th is for the event recording matrix.

In figure 3.2, the mos represents a wingbeat sound, scale with the gain factor G and then added to env, an environmental sound at the time t_{start}. The label is also created from this information, as shown in figure 3.3. A section without mosquito presence will only consist of environment sounds, represented in a red rectangle in the image. This part is labelled as a matrix filled with 0s. On the other hand, the section with a mosquito presence is filled with 1s.

figures/chap3/env overlay.drawio.pdf

Figure 3.2: Example of overlaying a wingbeat sound on an environmental sound. The mosquito wingbeat sound starts on from t_{start} and ends at $t_{start} + t_{mos}$.

figures/chap3/label from overlay.pdf

Figure 3.3: Example of the label created from overlaying a wingbeat sound on an environmental sound. The red rectangle represents the matrix with all zeros, and The green rectangle represents the matrix with all one, starting from t_{start} and ending at $t_{start} + t_{mos}$.

3.3 Model Architecture

In this section, the architecture of the baseline model, along with its techniques and weaknesses, will be explained. Then, the authors will explain how the proposed model will replace parts of the architecture to overcome the baseline's disadvantages.

3.3.1 SEDNet

the authors selected state of the art for SED, SEDNet from [28], as a baseline. SEDNet utilizes log mel band energy (lmbe) of the sound as an input feature of the model, passes it through 3 of 2D CNN blocks, then 2 Bidirectional Gated Recurrent units (BiGRUs) [45][46] and two fully connected layers with a sigmoid activation function on the last layer to map the result of the model to range $[0, 1]$. The figure of the complete architecture is shown in figure 3.5.

A 2D CNN block consists of several layers in the following order: 2D convolution, Batch normalization [47], ReLU activation function, Max pooling 2D, and Dropout as illustrated in figure 3.4.

Given a sound with a duration of n seconds, namely $x \in [-1, 1]^{sr \cdot n}$, a log mel band energy of the sound, lmbe(x), can be calculated by using the following steps. First, calculate the Short-Time Fourier Transform (STFT) of the sound, denoted $\hat{x} =$ STFT(x), then calculate the energy of each frequency by applying element-wise absolute function en$(x) = |\hat{x}|$. Next, transform from linear scale to mel scale, mbe$(x) =$ mel $\cdot$ en(x), where mel scale is define as mel$(f) = 2595 \log_{10}(1 + \frac{f}{700})$ where f is frequency (Hz). Lastly, apply element-wise natural logarithmic function, lmbe$(x) =$ ln(mbe(x)). There are also numerous hyperparameters to adjust, such as nfft, hoplen, and mel-band, which were implied in the equations above.

figures/chap3/cnnblock.drawio.pdf

Figure 3.4: An architecture of 2D CNN block. Activation function omitted.

figures/chap3/baseline sed.drawio.pdf

Figure 3.5: An architecture of baseline CRNN. Activation function omitted.

Utilizing multiple 2DCNN layers, SEDNet can see all sounds and noises in the data in 2D form, making the frequency of each mosquito visible and easy to differentiate.

However, frequency is only a significant feature in differentiating mosquito sexes and species since many species have overlapping frequency ranges. Additionally, SEDNet requires an extraction of mel-band spectrogram (lmbe). As the sounds are originally in 1D format, converting them into 2D may leave out some subtle features and cause the loss of the ability to investigate small differences.

3.3.2 One-dimensional Convolution Recurrent Neural Network (1DCRNN)

Inspired by the previously mentioned SEDNet from [28], the authors designed a similar architecture but removed lmbe and replaced 2D CNN blocks with 1D CNN blocks. The 1D CNN block is similar to the 2D variation but changes from 2D convolution to 1D, and the same logic is applied for the max pooling layer.

By removing lmbe feature extraction, our model receives the input waveform directly without frequency extraction. Since the lmbe ignores the fine-grain detail of the input features, the authors believe the model might gain some insight from these fine-grain features. Also, the hyperparameters in the lmbe are removed from the equation.

figures/chap3/1dcrnn_newarch.pdf

Figure 3.6: An architecture of 1DCRNN. Activation function omitted.

CHAPTER 4
RESULT

4.1 Experimental Setup

4.1.1 Data Preparation

Mosquito Wingbeat Sound

MIRU lab provided mosquito wingbeat sounds from two major sources. The first source uses the same method of collection as [21], which recorded the wingbeat sound using a Studio Behringer (Primo EM172) in the mono channel, 24 bits depth with 96kHz sampling rate at a researcher's home (without noise proof wall). The second source was collected recently with the same microphone setting but recorded in the studio room (with a noise-proof wall). This marks these two sources with significant differences in sound quality and characteristics. The sounds come in multiple cut files, and some belong to the same recordings of mosquitos. The authors ensure that each source and recording are allocated to the training, testing, and validating set with balance.

The number of wingbeats cut files from the MIRU lab is shown in the table 4.1 and 4.2. The authors encountered highly imbalanced data on Habitat A, specifically on class A.Minimus.M and An.Minimus.F. While other classes can be allocated to the training set, testing set, and validating set with balance on the data source, these two classes have a low quantity, making it impossible. Therefore, the authors stratify the chance of each class being chosen in creating training, validating, and testing datasets.

Mosquito Species	Male(files)	Female(files)
An. Dirus	178	152
Cx. Quin	113	120
Ae. Albopictus	218	128
Ae. Aegypti	152	166
An. Minimus	50	37

Table 4.1: The quantity and duration of wingbeats cut files from MIRU first Source.

Mosquito Species	Male(files)	Female(files)
An. Dirus	220	286
Cx. Quin	370	385
Ae. albopictus	271	280
Ae. Aegypti	300	303
An. Minimus	30	33

Table 4.2: The quantity and duration of wingbeats cut files from MIRU second Source.

Environmental Sound

The authors obtain three types of environmental sounds in this project: MIRU, Humbug, and Silence.

1. Type 1 comes from MIRU lab, which provided 10 files of 1851 seconds, consisting of birds, motorcycles and talking noises. These represent urban noises found in houses and cities, where mosquitoes are likely to live among humans.

2. Type 2 comes from HumBugDB [1]. To supply additional environmental sound, the authors also use data from HumbugDB, a public dataset consisting of sounds from mosquito and their environment collected from multiple locations worldwide. HumbugDB offers mosquito sounds of various origins, as well as environmental sounds. We decided to use only their environmental sounds and exclude the mosquitos, as their data collection method differs largely from ours. The authors kept the environment sounds longer than 10 seconds and used a sampling rate of 8000. There are 649 files with a duration of 38,884.6 seconds. The background noise includes walking, talking, breathing, periodic, and static noise.

3. Type 3 is silence or just simply no noise. Aiming to let the model learn pure mosquito sounds and noise, the authors added silence to the environmental sound dataset. As a result, the model would be able to understand the characteristics of each mosquito class without any interrupting noises. In practice, the author uses random Gaussian noise with low mean and low std. to simulate the low amplitude white noise since some algorithms cannot work with an array of all zeros.

4.1.2 Simulating the Habitats of Mosquito

To simulate the actual habitat of mosquitos, the authors prepare two separate settings to represent how mosquitos are found in real locations. The settings are called Habitat A and Habitat B, representing species commonly found together in one location.

1. Habitat A (Anopheles + Culex): This setting simulates a mixed-species environment where Anopheles and Culex mosquitoes coexist. The habitat consists of 6 classes: Culex quinquefasciatus Male and Female, Anopheles Dirus Male and Female, Anopheles Minimus Male and Female.

2. Habitat B (Aedes + Culex): Represents a scenario where Aedes and Culex genera are present. The habitat consists of 6 classes: Culex quinquefasciatus Male and Female, Aedes Aegypti Male and Female, Aedes Albopictus Male and Female.

These settings are created to evaluate the model under species compositions.

4.1.3 Dataset Creation

The authors created the dataset by randomly overlaying mosquito sounds onto the environment noises to simulate the natural setting. The gain factor G is uniformly random in range $[0.01, 0.1]$ for each wingbeat overlay to simulate the various ranges of noisy environments. The recordings are mono channel and were downsampled to 8 kHz as shown in [21] that 8 kHz performs better on deep learning model than 96 kHz, so the $sr = 8000$. A synthesized file is 10 seconds long ($t = 10$), with each data point being 1 second apart ($t_{seg} = 1$) and limited to having only 1 or 2 mosquitoes with no overlapping between mosquitoes.

When choosing environmental files for dataset creation, the authors ensure that each environmental sound is allocated with balance in the training, testing, and validating set. As for mosquito sounds, the authors also ensure that mosquito sounds from the same recording are used in the same dataset. The cut files from the same recording (the same mosquito) might be too similar, so they do not reflect real-world practice. Therefore, one mosquito recording will not be used in multiple datasets for training, testing, or validation.

In the overlaying of mosquito sounds and environmental sounds, the files are randomly chosen based on their quantity in the data pool. The mosquito classes that are most populated are more likely to be included in the dataset. The problem is imbalanced data is handled by incorporating class weights while training the model. Class weights are calculated by the inverse of a class's quantity among total files, as shown in the equation 4.1.

$$cw_{cls} = \frac{sc_{cls}}{\sum_{j \in classes} sc_j} \tag{4.1}$$

$$sc_{cls} = \frac{1}{\text{number of cut files in class } cls} \tag{4.2}$$

While habitat B is trained regarding class weights to handle imbalanced data, the authors used another method for habitat A. The files in low-populated classes are forced to have more probability of being chosen while generating a dataset using stratified sampling between each mosquito class. This is to account for the highly imbalanced data, especially An.Minimus.F and An.Minimus.M.

Training, validating, and testing datasets are prepared in each habitat separately. The training set lasts 20,000 seconds. The authors constructed the dataset with 25% overlayed Type 1 environmental sound, 25% overlayed Type 2 environmental sound and 50% overlayed 25% overlayed Type 3 environmental sound.

The validating and testing sets last 3,600 seconds and have the same settings as the training sets. They are fixed throughout all training loops and evaluations.

4.2 Model Training

Every model was trained with Adam optimizer [48] at a learning rate of 0.001 and binary cross entropy loss as a loss function with batch size 32. They were trained for 1,000 epochs with early stopping if the validation error rate did not improve further for 50 epochs. The model is trained separately in each habitat. For habitat B, the model is trained with corresponding class weights.

4.2.1 Data Augmentation

The dataset is augmented by regenerating the training dataset every 30 epochs of training loops. This newly generated dataset employs the same characteristics as the original one, using mosquito and noise data from the same pool of sources as described in the section Data Preparation. The differences are apparent in the randomized choice of noises and mosquito sounds. As one is generated, the algorithm uses the following techniques:

- **Random combination of noises and mosquitos:** a noise and 1-2 mosquito sounds are chosen randomly from the original training pool.
- **Amplitude variation:** Each mosquito sound is applied with a random gain factor ranging from 0.01 to 0.1.
- **Overlay sounds:** Mosquito sounds are overlayed onto noise with randomized timestamps.

One dataset is used for only 30 epochs, and a new one is generated. This creates a wider learning material for the model, allowing it to learn the mosquito sounds under various noises and amplitudes.

4.2.2 Threshold Optimization

As the output of model prediction is in the form of probability, a threshold is needed to convert it into a prediction of the presence and absence of mosquitoes. A general threshold is 0.5, where values less than 0.5 are interpreted as absence and vice versa. In this project, the authors determine the optimal threshold that yields the highest detection F1 score on the validation set as shown in equation 4.3. The $\hat{y}$ is a model prediction on the validation set. This optimal threshold is then used on the testing set. Each model's threshold is different as they are chosen by this optimization method.

$$thres_{cls} = \underset{thres \in [0,1]}{\text{argmax}} \; F1(y[t, cls], \hat{y}[t, cls] > thres) \tag{4.3}$$

4.3 Evaluation Metrics

Unlike monophonic SED, polyphonic SED has no widely accepted metrics [49], Since polyphonic SED allows the prediction to be correct in one class and wrong in another class at the same time. The authors use a segment-based method, namely evenly separating the whole signal duration into a small segment.

The authors experimented with 2 metrics proposed in [28] and [49], Error rate (ER), and F1-score (F1). The equation 4.4 can calculate the error rate.

$$ER = \frac{\sum_{t=1}^{N} s_t + \sum_{t=1}^{N} i_t + \sum_{t=1}^{N} d_t}{\sum_{t=1}^{R} a_t} \tag{4.4}$$

Where s_t is the substitution error of segment t, i_t is the insertion error of segment t, d_t is the deletion error of segment t, a_t is, and a_t is the number of events in segment t. Notice that the ER is in the range $[0, \infty)$ as the number of errors can be as much as the number of data points in the label. If the ER is high (especially if it goes over 1), it also means that the model predicts more than it should. But if the ER is low (around 0), the model prediction barely has an error.

Since the model prediction is a binary classification of each class, the authors also use F1-Score to measure the class-wise classification performance and overall, which can be calculated as follows.

$$P = \frac{TP}{TP + FP} \tag{4.5}$$

$$R = \frac{TP}{TP + FN} \tag{4.6}$$

$$F1 = \frac{2 \cdot P \cdot R}{P + R} \tag{4.7}$$

Where TP is true positive (predict correctly), FP is false positive (predict where it should not), and FN is false negative (no prediction where it should).

However, the authors also interpret an event as mosquito wingbeat detection (not classification). Assume that if any of the classes have a score over the threshold $thres$, then the authors interpret that the model has detected an event using the equation 4.9 and create a detection label using equation 4.8.

$$y_{det}[t] = \max_{cls \in classes} y[t, cls] \tag{4.8}$$

$$\hat{y}_{det}[t] = (\max_{cls \in classes} \hat{y}[t, cls]) > thres \tag{4.9}$$

Then, The authors can calculate the F1 score on y_{det} and $\hat{y}_{det}$ for detection performance.

4.4 Results

After developing the 1DCRNN model and outlining its processing capabilities and architecture, it is essential to compare its performance to the state-of-the-art SEDNet model presented in Section 3.3.1 of Chapter 3 (Model Architecture). This comparison aims to identify the performance of 1DCRNN relative to the SEDNet to provide insights into the advancements made in sound event detection.

To facilitate unbiased experiments, the following conditions were adhered to:

- **Settings Parameters:** The configuration of both models, including the hyperparameters such as learning rate, batch size, the number of epochs, and the early stopping condition, was kept constant.

Differences in the computational environment are duly noted:

- **SEDNet Training and Evaluation:**SEDNet was trained and evaluated on a desktop computer equipped with an NVIDIA GeForce RTX 3090 GPU.
- **1DCRNN Training and Evaluation:**In contrast, 1DCRNN utilized GPU V4 of Google Colab for its computations.

4.4.1 Find the best settings

The authors need to find the optimal configuration parameters for the SEDNet and 1DCRNN models to maximize their performance in sound event detection. It comprehensively describes the experimental setup, emphasizing the significance of the number of mel coefficients (N mels), Data augmentation, and Threshold Optimization.

The tables below present the results of the experiments for the SEDNet model in two distinct habitats, illustrating how varying the number of mel coefficients (N mels) affects the models' performance.

Habitat A Results								
Model	N mels	F1	ER	Precision classify	Recall classify	F1 detection	Precision detection	Recall detection
SEDNet	40	0.525	0.623	**0.594**	0.471	0.788	0.888	0.708
	128	**0.536**	**0.592**	0.593	**0.489**	**0.823**	**0.908**	**0.752**

(a) Experiments result in Habitat A

Habitat B Results								
Model	N mels	F1	ER	Precision classify	Recall classify	F1 detection	Precision detection	Recall detection
SEDNet	40	0.506	0.666	**0.558**	0.463	0.776	0.852	0.713
	128	**0.506**	**0.651**	0.546	**0.472**	**0.805**	**0.864**	**0.754**

(b) Experiments result in Habitat B

Table 4.3: Classification and Detection Performance of SEDNet in different habitats on varying the number of mel coefficients (N mels).

With comparing the results within Habitat A, the model configured with 128 mel bands shows improvement in several key performance metrics over the configuration with 40 metal bands. For instance, the F1 score for classification improved from 0.525 to 0.536, and the Error Rate (ER) decreased from 0.623 to 0.592. This trend is also consistent with the detection metrics, where both Precision and Recall scores show notable increases.

A similar pattern is observed in Habitat B, where the model with 128 mel bands again outperforms the 40 mel bands configuration. The F1 score remains constant at 0.506, but the Error Rate decreases from 0.666 to 0.651. More importantly, the Precision in detection tasks jumps from 0.852 to 0.864 and Recall from 0.713 to 0.754, underscoring an overall improvement in model accuracy.

Table 4.4 shows the result between the threshold settings and the performance metrics of the SEDNet in different habitats. In Habitat A, the model's performance with a threshold of 0.374 showcases an enhanced F1 score of 0.559 compared to 0.536 at a threshold of 0.5, suggesting that a lower threshold may be more optimal in scenarios characterized by this specific habitat's acoustic properties. Notably, the Recall in detection at the lower threshold improves remarkably to 0.842 from 0.752, underscoring a superior capability to correctly identify sound events without increasing the number of false positives, as evidenced by the Precision in detection slightly decreasing from 0.908 to 0.878.

Similarly, in Habitat B, adjusting the threshold to 0.400 leads to an improvement in both the F1 score and Recall in detection, moving from 0.506 to 0.519 and from 0.754 to a significant 0.908, respectively. This adjustment indicates that the model becomes more sensitive to detecting true positives without overly compromising on precision, though the Error Rate does increase from 0.651 to 0.736, which may indicate a trade-off between sensitivity and error tolerance in more challenging acoustic environments.

Habitat A Results								
Model	Threshold	F1	ER	Precision classify	Recall classify	F1 detection	Precision detection	Recall detection
SEDNet	0.5	0.536	**0.592**	**0.593**	0.489	0.823	**0.908**	0.752
	0.374	**0.559**	0.660	0.540	**0.579**	**0.860**	0.878	**0.842**

(a) Experiments result in in Habitat A

Habitat B Results								
Model	Threshold	F1	ER	Precision classify	Recall classify	F1 detection	Precision detection	Recall detection
SEDNet	0.5	0.506	**0.651**	**0.546**	0.472	0.805	**0.864**	0.754
	0.400	**0.519**	0.736	0.475	**0.571**	**0.876**	0.847	**0.908**

(b) Experiments result in Habitat B

Table 4.4: Classification and Detection Performance of SEDNet in different habitats on Threshold optimization.

These experiments suggest that careful optimization of the detection threshold, as further discussed in Section 4.2.2, is important for the improvement of the SEDNet model to perform optimally under varying environmental conditions.

Table 4.5 contrasts the performance metrics of the SEDNet model between the original training model and a modified model that involves data augmentation every 30 epochs. In Habitat A, the original model achieved a significantly higher F1 score for classification at 0.559 compared to 0.467 post-augmentation. Interestingly, the augmented model in Habitat A shows an improved Error Rate (ER) at 0.467, down from 0.660, and a higher Recall in detection at 0.900, up from 0.842. These improvements indicate that overall classification accuracy decreased.

Similar trends are observable in Habitat B. The augmented model's F1 score decreased slightly from 0.519 to 0.502, and the ER worsened from 0.736 to 0.811. However, similar to Habitat A, the Recall in detection increased from 0.908 to 0.920.

Habitat A Results								
Model	Thres hold	F1	ER	Precision classify	Recall classify	F1 detection	Precision detection	Recall detection
Original	0.374	**0.559**	0.660	**0.540**	**0.579**	**0.860**	**0.878**	0.842
Augmented	0.451	0.467	**0.467**	0.436	0.503	0.859	0.821	**0.900**

(a) Experiments result in in Habitat A

Habitat B Results								
Model	Thres hold	F1	ER	Precision classify	Recall classify	F1 detection	Precision detection	Recall detection
Original	0.400	**0.519**	**0.736**	**0.475**	0.571	**0.876**	**0.847**	0.908
Augmented	0.397	0.502	0.811	0.445	**0.576**	0.867	0.820	**0.920**

(b) Experiments result in in Habitat B

Table 4.5: Classification and Detection Performance of SEDNet in different habitats on Data augmentation.

Evaluations are completed in two settings, one for classification performance and another for detection performance. The classification evaluations are also computed separately between classes, identifying the models' performance in different species and sexes. The evaluation methods are conducted separately between habitats A and B to adhere to the mosquito species found in real environments.

The proposed 1DCRNN model uses the optimal threshold of 0.133 on habitat A, and 0.148 on habitat B. SEDNet uses a threshold of 0.374 on habitat A and 0.400 on habitat B.

4.4.2 Classification Performance

Table 4.6 shows the classification performance across all datasets. The proposed model achieved an F1 score of 0.537 on the classification task across all classes for habitat A and 0.603 for habitat B. In habitat B, our proposed model achieved a higher F1 score than SEDNet. On the other hand, SEDNet outperformed 1DCRNN in habitat A. Even though each model overcame the others in different habitat settings, the ER in SEDNet is noticeably higher than 1DCRNN for habitat B.

Model	F1	ER
SEDNet	**0.559**	**0.660**
1DCRNN	0.537	0.666

(a) Habitat A (An + Cx)

Model	F1	ER
SEDNet	0.529	1.036
1DCRNN	**0.603**	**0.648**

(b) Habitat B (Ae + Cx)

Table 4.6: Classification performance

	Per-Class F1					
Model	An.Mini.M	An.Diru.M	Cx.Quin.M	An.Mini.F	An.Diru.F	Cx.Quin.F
SEDNet	0.000	**0.591**	0.458	0.070	0.520	**0.772**
1DCRNN	0.000	0.423	**0.465**	**0.353**	**0.566**	0.671

(a) Classification performance per class in habitat A

	Per-Class F1					
Model	Ae.Aegy.M	Ae.Albo.M	Cx.Quin.M	Ae.Aegy.F	Ae.Albo.F	Cx.Quin.F
SEDNet	0.181	0.360	0.453	0.572	0.514	0.756
1DCRNN	**0.391**	**0.449**	**0.532**	**0.591**	**0.759**	**0.773**

(b) Classification performance per class in habitat B.

Table 4.7: Classification performance per class. The **bold** font represents the best number in the category.

Observing the performance from each class in habitat A separately (see figure 4.7a), Our model mostly outperformed SEDNet. Our model has a higher F1 score on the following classes: Cx.Quin.M, An.Mini.F, and An.Diru.F However, for classes An.Minimus.F and An.Minimus.M, both models showed low performance, especially on An.Minimus.M, where both are unable to be classified. The authors suggest that this class's imbalanced data cause this problem. Since the testing set of this class is in high shortage with only a few samples to be evaluated, the performance drops to zero. It is worth noting that in class An.Minimus.F, our model was able to achieve a 0.353 F1 score, compared to 0.070 from

As for the per-class evaluation results in habitat B, our 1DCRNN model showed an outstanding classification performance compared to SEDNet (see figure 4.7b). Model 1DCRNN outperformed SEDNet in every class, For Class Ae.Albopictus.F is where the 1DCRNN model achieved a much higher F1 score than SEDNet, with a difference of

0.183.

4.4.3 Detection Performance

The following figure 4.1 compares the detection performance of SEDNet and 1DCRNN on habitats A and B.

Model	F1	Precision	Recall
SEDNet	0.860	0.878	**0.842**
1DCRNN	**0.877**	**0.940**	0.822

(a) Habitat A (An + Cx)

Model	F1	Precision	Recall
SEDNet	0.876	0.847	**0.908**
1DCRNN	**0.936**	**0.967**	0.906

(b) Habitat B (Ae + Cx)

Figure 4.1: Detection performance

In terms of detection performance, our proposed model demonstrates its great ability to detect classes, exceeding baseline on both habitat settings. On habitat A, 1DCRNN overcome SEDNet slightly in terms of F1 score. Considering precision and recall, our model proved to possess a much more precise capability to identify the correct class with a precision of 0.940, compared to 0.878 from SEDNet. This is also true in habitat B, where the precision of our proposed model is much higher than the baseline. Although the recall scores of the 1DCRNN model are slightly lower than SEDNet, the high precision compensates for it, resulting in a higher F1 score.

4.5 Discussion

The comparative results between 1DCRNN and SEDNet in the previous section provide insight into the performance of each model across the two habitats from detection and classification points of view.

Regarding detecting mosquito sounds, it is clear that the proposed 1DCRNN model outperforms SEDNet in habitats A and B. The higher F1 and precision scores show that the 1DCRNN model could better recognize mosquito sounds among the interrupting noises than the baseline. Even though the recall scores for the 1DCRNN model were lower, the difference is too subtle to affect the overall performance. The result shows that 1DCRNN is preferable for mosquito detection tasks.

For the classification task, the results are competitive between SEDNet and 1DCRNN models. SEDNet showed better results in habitat A, while 1DCRNN outperformed it in habitat B — the classification result for species Cx.Quin, present in both habitats, does not show a high difference between the proposed model and the baseline — the class of An.Dirus.M is where SEDNet overcomes 1DCRNN by a high F1 score. On the other hand, 1DCRNN also highly outperformed the baseline on Ae.Albopictus.F class, as well.

This clear separation of performance on each habitat suggests that each model can be deployed in different locations based on the species of mosquitos in the area. Using each model on its specialized habitats will allow them to maximize performance and give the best result on classification.

One interesting part is the species An.Minimus, where the imbalance data issue damages the performance of both models. Despite this issue, the 1DCRNN model was able to classify An.Minimus.F quite well on limited data, achieving an F1 score of 0.353, compared to SEDNet, which cannot classify it. This result indicates that 1DCRNN may be more suited to use in imbalance settings, where the data is scarce.

REFERENCES

[1] Sinka ME., Zilli D., Li Y., Kiskin I., Msaky D., Kihonda J., et al., “HumBug – An Acoustic Mosquito Monitoring Tool for use on budget smartphones”, Methods in Ecology and Evolution. Oct. 2021;12(10):1848–1859, [Online]. Available: https://besjournals.onlinelibrary.wiley.com/doi/10.1111/2041-210X.13663.

[2] WHO, “Vector borne diseases”. 3 2020;.

[3] Franklinos LHV., Jones KE., Redding DW., Abubakar I., “The effect of global change on mosquito-borne disease”, The Lancet Infectious Diseases. Sep. 2019;19(9):e302–e312, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1473309919301616.

[4] Dormont L., Mulatier M., Carrasco D., Cohuet A., “Mosquito Attractants”, Journal of Chemical Ecology. May 2021;47(4):351–393, [Online]. Available: https://doi.org/10.1007/s10886-021-01261-2.

[5] Offenhauser WH., Kahn MC., “The sounds of disease-carrying mosquitoes”, The Journal of the Acoustical Society of America. 1949;21(3):259–263.

[6] Vasconcelos D., Nunes N., Ribeiro M., Prandi C., Rogers A., “LOCOMOBIS: a low-cost acoustic-based sensing system to monitor and classify mosquitoes”, In: 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC). Las Vegas, NV, USA: IEEE; Jan. 2019. p. 1–6, [Online]. Available: https://ieeexplore.ieee.org/document/8651767/.

[7] Lukman A., Harjoko A., Yang CK., “Classification MFCC feature from Culex and Aedes aegypti Mosquitoes Noise using Support Vector Machine”, In: 2017 International Conference on Soft Computing, Intelligent System and Information Technology (ICSIIT); Sep. 2017. p. 17–20, [Online]. Available: https://ieeexplore.ieee.org/document/8262536.

[8] Ziemer T., Wetjen F., Herbst A., "The Antenna Base Plays a Crucial Role in Mosquito Courtship Behavior", Frontiers in Tropical Diseases. 2022;3, [Online]. Available: https://www.frontiersin.org/articles/10.3389/fitd.2022.803611.

[9] Chen Y., Why A., Batista G., Mafra-Neto A., Keogh E., "Flying Insect Classification with Inexpensive Sensors", Journal of Insect Behavior. Sep. 2014;27(5):657–677, [Online]. Available: http://link.springer.com/10.1007/s10905-014-9454-4.

[10] Yin MS., Haddawy P., Nirandmongkol B., Kongthaworn T., Chaisumritchoke C., Supratak A., et al., "A Lightweight Deep Learning Approach to Mosquito Classification from Wingbeat Sounds", In: Proceedings of the Conference on Information Technology for Social Good. Roma Italy: ACM; Sep. 2021. p. 37–42, [Online]. Available: https://dl.acm.org/doi/10.1145/3462203.3475908.

[11] Steinfath E., Palacios-Muñoz A., Rottschäfer JR., Yuezak D., Clemens J., "Fast and accurate annotation of acoustic signals with deep neural networks", eLife;10:e68837, [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8560090/.

[12] Stowell D., "Computational bioacoustics with deep learning: a review and roadmap", PeerJ. Mar. 2022;10:e13152, [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8944344/.

[13] Morfi V., Lachlan RF., Stowell D., "Deep perceptual embeddings for unlabelled animal sound eventsa)", The Journal of the Acoustical Society of America. Jul. 2021;150(1):2–11, [Online]. Available: https://doi.org/10.1121/10.0005475.

[14] Varma ALSVS., Bateshwar V., Rathi A., Singh A., "Acoustic Classification of Insects using Signal Processing and Deep Learning Approaches", In: 2021 8th International Conference on Signal Processing and Integrated Networks (SPIN); Aug. 2021. p. 1048–1052, ISSN: 2688-769X, [Online]. Available: https://ieeexplore.ieee.org/document/9566121.

[15] Vasconcelos D., Yin MS., Wetjen F., Herbst A., Ziemer T., Förster A., et al., "Counting Mosquitoes in the Wild: An Internet of Things Approach", In: Pro-

ceedings of the Conference on Information Technology for Social Good. GoodIT '21. New York, NY, USA: Association for Computing Machinery; Sep. 2021. p. 43–48, [Online]. Available: https://dl.acm.org/doi/10.1145/3462203.3475914.

[16] Li Y., Kiskin I., Sinka M., Zilli D., Chan H., Herreros-Moya E., et al., "Fast mosquito acoustic detection with field cup recordings: an initial investigation.", In: DCASE; 2018. p. 153–157.

[17] Genoud AP., Basistyy R., Williams GM., Thomas BP., "Optical remote sensing for monitoring flying mosquitoes, gender identification and discussion on species identification", Applied Physics B. Feb. 2018;124(3):46, [Online]. Available: https://doi.org/10.1007/s00340-018-6917-x.

[18] Stowell D., Petrusková T., Šálek M., Linhart P., "Automatic acoustic identification of individuals in multiple species: improving identification across recording conditions", Journal of the Royal Society Interface. Apr. 2019;16(153):20180940, [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6505557/.

[19] Shorten C., Khoshgoftaar TM., "A survey on Image Data Augmentation for Deep Learning", Journal of Big Data. Jul. 2019;6(1):60, [Online]. Available: https://doi.org/10.1186/s40537-019-0197-0.

[20] Nanni L., Maguolo G., Paci M., "Data augmentation approaches for improving animal audio classification", Ecological Informatics. May 2020;57:101084, [Online]. Available: https:// www.sciencedirect.com/ science/ article/ pii/ S1574954120300340.

[21] Yin MS., Haddawy P., Ziemer T., Wetjen F., Supratak A., Chiamsakul K., et al., "A deep learning-based pipeline for mosquito detection and classification from wingbeat sounds", Multimedia Tools and Applications. Feb. 2023;82(4):5189–5205, [Online]. Available: https://link.springer.com/10.1007/s11042-022-13367-0.

[22] Kiskin I., Zilli D., Li Y., Sinka M., Willis K., Roberts S., "Bioacoustic detection with wavelet-conditioned convolutional neural networks", Neural Computing

and Applications. Feb. 2020;32(4):915–927, [Online]. Available: https://doi.org/10.1007/s00521-018-3626-7.

[23] Mukundarajan H., Hol FJH., Castillo EA., Newby C., Prakash M., “Using mobile phones as acoustic sensors for high-throughput mosquito surveillance”, eLife. Oct. 2017;6:e27854, [Online]. Available: https://elifesciences.org/articles/27854.

[24] Simonyan K., Zisserman A.. “Very Deep Convolutional Networks for Large-Scale Image Recognition”, arXiv; Apr. 2015, ArXiv:1409.1556 [cs], [Online]. Available: http://arxiv.org/abs/1409.1556.

[25] He K., Zhang X., Ren S., Sun J., “Deep Residual Learning for Image Recognition”, In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA: IEEE; Jun. 2016. p. 770–778, [Online]. Available: http://ieeexplore.ieee.org/document/7780459/.

[26] Tan M., Le QV.. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”, arXiv; Sep. 2020, ArXiv:1905.11946 [cs, stat], [Online]. Available: http://arxiv.org/abs/1905.11946.

[27] Mesaros A., Heittola T., Virtanen T., Plumbley MD., “Sound Event Detection: A tutorial”, IEEE Signal Processing Magazine. Sep. 2021;38(5):67–83, Conference Name: IEEE Signal Processing Magazine, [Online]. Available: https://ieeexplore.ieee.org/document/9524590.

[28] Cakır E., Parascandolo G., Heittola T., Huttunen H., Virtanen T., “Convolutional recurrent neural networks for polyphonic sound event detection”, IEEE/ACM Transactions on Audio, Speech, and Language Processing. 2017;25(6):1291–1303.

[29] Adavanne S., Pertila P., Virtanen T., “Sound event detection using spatial features and convolutional recurrent neural network”, In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). New Orleans, LA: IEEE; Mar. 2017. p. 771–775, [Online]. Available: http://ieeexplore.ieee.org/document/7952260/.

[30] Ravanelli M., Bengio Y.. "Speaker Recognition from Raw Waveform with SincNet", arXiv; Aug. 2019, ArXiv:1808.00158 [cs, eess], [Online]. Available: http://arxiv.org/abs/1808.00158.

[31] Kiranyaz S., Ince T., Gabbouj M., "Real-Time Patient-Specific ECG Classification by 1-D Convolutional Neural Networks", IEEE Transactions on Biomedical Engineering. 2016;63(3):664–675.

[32] Kalchbrenner N., Blunsom P., "Recurrent Continuous Translation Models", In: Yarowsky D., Baldwin T., Korhonen A., Livescu K., Bethard S., editors. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Seattle, Washington, USA: Association for Computational Linguistics; Oct. 2013. p. 1700–1709, [Online]. Available: https://aclanthology.org/D13-1176.

[33] Zaytar MA., El Amrani C., "Sequence to sequence weather forecasting with long short-term memory recurrent neural networks", International Journal of Computer Applications. 2016;143(11):7–11.

[34] Hochreiter S., Schmidhuber J., "Long short-term memory", Neural computation. 1997;9(8):1735–1780.

[35] Cho K., Van Merriënboer B., Gulcehre C., Bahdanau D., Bougares F., Schwenk H., et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation", arXiv preprint arXiv:14061078. 2014;.

[36] Wanhammar L., "Digital Signal Processing", In: DSP Integrated Circuits. Elsevier; 1999. p. 59–114, [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/B9780127345307500039.

[37] Kehtarnavaz N., "CHAPTER 7 - Frequency Domain Processing", In: Kehtarnavaz N., editor. Digital Signal Processing System Design (Second Edition). Burlington: Academic Press; Jan. 2008. p. 175–196, [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123744906000076.

[38] Campos S., Khandelwal D., Nagaraj SC., Nugen F., Todeschini A., "Deep Learning-Based Acoustic Mosquito Detection in Noisy Conditions Using Trainable Kernels and Augmentations", In: Proceedings of the 30th ACM International Conference on Multimedia. Lisboa Portugal: ACM; Oct. 2022. p. 7094–7098, [Online]. Available: https://dl.acm.org/doi/10.1145/3503161.3551586.

[39] Mueen A., Keogh E., Zhu Q., Cash SS., Westover MB., Bigdely-Shamlo N., "A disk-aware algorithm for time series motif discovery", Data Mining and Knowledge Discovery. Jan. 2011;22(1-2):73–105, [Online]. Available: http://link.springer.com/10.1007/s10618-010-0176-8.

[40] Guirguis K., Schorn C., Guntoro A., Abdulatif S., Yang B., "SELD-TCN: Sound Event Localization & Detection via Temporal Convolutional Networks", In: 2020 28th European Signal Processing Conference (EUSIPCO); Jan. 2021. p. 16–20, ArXiv:2003.01609 [cs, eess, stat], [Online]. Available: http://arxiv.org/abs/2003.01609.

[41] Miyazaki K., Komatsu T., Hayashi T., Watanabe S., Toda T., Takeda K., "Weakly-Supervised Sound Event Detection with Self-Attention", In: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Barcelona, Spain: IEEE; May 2020. p. 66–70, [Online]. Available: https://ieeexplore.ieee.org/document/9053609/.

[42] Gemmeke JF., Ellis DPW., Freedman D., Jansen A., Lawrence W., Moore RC., et al., "Audio Set: An ontology and human-labeled dataset for audio events", In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); Mar. 2017. p. 776–780, ISSN: 2379-190X, [Online]. Available: https://ieeexplore.ieee.org/document/7952261.

[43] Supratak A., Haddawy P., Yin MS., Ziemer T., Chiamsakul K., Chantanalertvilai T., et al., "MosquitoSong+: A noise-robust deep learning model for mosquito classification from wingbeat sounds". 2023;.

[44] Fanioudakis E., Geismar M., Potamitis I., “Mosquito wingbeat analysis and classification using deep learning”, In: 2018 26th European Signal Processing Conference (EUSIPCO); Sep. 2018. p. 2410–2414, ISSN: 2076-1465, [Online]. Available: https://ieeexplore.ieee.org/document/8553542.

[45] Cho K., Van Merriënboer B., Bahdanau D., Bengio Y., “On the properties of neural machine translation: Encoder-decoder approaches”, arXiv preprint arXiv:14091259. 2014;.

[46] Schuster M., Paliwal KK., “Bidirectional recurrent neural networks”, IEEE transactions on Signal Processing. 1997;45(11):2673–2681.

[47] Ioffe S., Szegedy C., “Batch normalization: Accelerating deep network training by reducing internal covariate shift”, In: International conference on machine learning. pmlr; 2015. p. 448–456.

[48] Kingma DP., Ba J., “Adam: A method for stochastic optimization”, arXiv preprint arXiv:14126980. 2014;.

[49] Mesaros A., Heittola T., Virtanen T., “Metrics for Polyphonic Sound Event Detection”, Applied Sciences. 2016;6:162, [Online]. Available: https://api.semanticscholar.org/CorpusID:9101533.

BIOGRAPHIES

NAME	Mr. Phuriwat Angkoondittaphong
INSTITUTIONS ATTENDED	Thai-German Pre-Engineering School , 2019:
	High School Diploma
	Mahidol University, 2024:
	Bachelor of Science (ICT)
NAME	Ms. Napahatai Sitirit
INSTITUTIONS ATTENDED	The Prince Royal's College , 2019:
	High School Diploma
	Mahidol University, 2024:
	Bachelor of Science (ICT)
NAME	Mr. Danaidech Ardsamai
INSTITUTIONS ATTENDED	Assumption Convent Lamnarai , 2019:
	High School Diploma
	Mahidol University, 2024:
	Bachelor of Science (ICT)