



SCHOOL of
GRADUATE STUDIES
EAST TENNESSEE STATE UNIVERSITY

East Tennessee State University
Digital Commons @ East Tennessee
State University

Electronic Theses and Dissertations

Student Works

8-2020

Assessing the Impact of Restored Wetlands on Bat Foraging Activity Over Nearby Farmland

Philip Allagas
East Tennessee State University

Follow this and additional works at: <https://dc.etsu.edu/etd>

 Part of the [Biology Commons](#)

Recommended Citation

Allagas, Philip, "Assessing the Impact of Restored Wetlands on Bat Foraging Activity Over Nearby Farmland" (2020). *Electronic Theses and Dissertations*. Paper 3804. <https://dc.etsu.edu/etd/3804>

This Thesis - embargo is brought to you for free and open access by the Student Works at Digital Commons @ East Tennessee State University. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of Digital Commons @ East Tennessee State University. For more information, please contact digilib@etsu.edu.

Assessing the Impact of Restored Wetlands on Bat Foraging Activity Over Nearby Farmland

A thesis

presented to

the faculty of the Department of Biology

East Tennessee State University

In partial fulfillment

of the requirements for the degree

Master of Science in Biology

by

Philip L. Allagas

August 2020

Dr. Richard Carter, Chair

Dr. T. Andrew Joyner

Dr. Thomas F. Laughlin

Keywords: Ecology, Conservation, Bat Activity, Wetlands, Machine Learning, Deep Neural
Networks, Transfer Learning, Bioacoustics, GIS, Species Distribution Modeling

ABSTRACT

Assessing the Impact of Restored Wetlands on Bat Foraging Activity Over Nearby Farmland

by

Philip L. Allagas

Up to 87% of the world's wetlands have been destroyed, considerably reducing ecosystem services these wetlands once provided. More recently, many wetlands are being restored in an attempt to regain their ecosystem service. This study seeks to determine the effects of restored wetlands on local bat habitat use. Bat activity was found to be significantly higher around the wetlands when compared to distant grassy fields; however, no significant difference was found among the restored wetlands and a remote cattle farm containing multiple water features.

Geospatial models of bat distribution and bat foraging were produced using machine learning that showed higher habitat suitability and foraging activity around restored wetlands than around distant grassy fields, suggesting that wetlands provide vital habitat for insectivorous bats. This study demonstrates that restored wetlands promote bat activity and bat foraging, and restoring wetlands may be a useful means of increasing natural pest control over nearby farmlands.

Copyright 2020 by Philip L. Allagas
All Rights Reserved

DEDICATION

I would like to dedicate my research to my amazing wife, Shawna Bearden-Allagas. Without her endless encouragement and support, I would never have had the inspiration to study the lovely little bats of Shady Valley. Although we never got to hold a bat, she still always listened when I was thinking out loud about whatever problem I was working through, and she never doubted that I would figure it out. After far too many nights of locking myself away with a computer, or sitting in the garage thinking, I finally finished, and I would never have without her support.

I would also like to dedicate this thesis to our puppies, Milan, the energetic Great Dane, and Dude, the most laid-back Labrador. They were instrumental in helping me decompress when I was frustrated to the point of quitting and always reluctantly gave me the time to run yet another model when they would have preferred playing or snuggling on the couch while watching cartoons. Despite their strange looks when I tried to explain a feeding buzz, I feel like they were every bit as invested as me.

Finally, I would like to thank all of the professors of ETSU who answered my many questions. First, I would thank my advisor, Dr. Carter, for taking me on as an aimless graduate student with a vague idea for a research project and guiding me to a study that I am proud of. Even when I came to him several times, asking him to spend more and more of his research budget on equipment, he never said no, only “what do you need?” He provided the instruments and sage advice that helped me grow exponentially as a student, and as a biologist. I would also like to thank my committee members, Dr. Joyner and Dr. Laughlin, for their insight into my research project. My thesis is undoubtedly more refined because of their feedback.

TABLE OF CONTENTS

ABSTRACT.....	2
DEDICATION.....	4
LIST OF TABLES.....	7
LIST OF FIGURES.....	8
LIST OF ABBREVIATIONS.....	9
CHAPTER 1. INTRODUCTION.....	10
Background.....	10
Project Overview.....	12
Hypotheses.....	14
An Overview of Machine Learning.....	14
Artificial Neural Networks.....	15
Deep Neural Networks.....	16
Transfer Learning.....	17
Machine Learning in Ecology.....	17
Species Distribution Modeling.....	19
CHAPTER 2. MATERIALS AND METHODS.....	21
Study Area and Site Selection.....	21
Passive Acoustic Monitor Setup and Data Collection.....	22
Data Processing.....	24
Statistical Analysis.....	24
Deep Neural Network: Training and Predicting.....	25
Modeling the Distribution of Bats.....	26
Modeling Bat Foraging Activity.....	27
CHAPTER 3. RESULTS.....	29
Passive Acoustic Monitoring.....	29
Statistical Analysis of Bat Activity.....	30
Google’s InceptionV3 Model Retrained.....	31
Species Distribution Model.....	32
Foraging Activity Model.....	34
CHAPTER 4. DISCUSSION.....	37
Bat Activity Among the Wetland Properties.....	37
Bat Activity of Near-Wetland Fields Versus Distant Fields.....	37
Efficacy of Identifying Feeding Buzzes with Machine Learning.....	38
Foraging Activity over Near-Wetland Fields Versus Distant Fields.....	40
Conclusions.....	41
REFERENCES.....	43
APPENDICES.....	50

Appendix A: Python Code for Data Processing.....	50
WavChop.py	50
Generate_spectrograms.py.....	51
Appendix B: Tensorflow Code for Retraining Google’s InceptionV3 Model.....	52
retrain.py	52
predict.py	56
Appendix C: Environmental Variables for Spatial Modeling.....	59
Digital Elevation Model.....	59
Land Cover	59
Aspect	60
Topographic wetness index	60
Euclidean distances to buildings.....	60
Euclidean distances to roads	60
Euclidean distances to streams	60
Euclidean distances to the forest edge	61
Appendix D: R Code for Ensemble Species Distribution Modeling in Biomod2	62
VITA.....	65

LIST OF TABLES

Table 1: File counts after each step of data processing. Note: Noise files were not scrubbed from the preliminary dataset in order to preserve noise files for training the neural network.	29
Table 2: Individual CNN model performance statistics. The training and validation performances were evaluated on the split training dataset. The testing performance was evaluated base on 10,000 randomly selected files from the study-period dataset...32	32
Table 3: Performance metrics of original GLM models. The area under the receiver operator curve (AUC) and True skill statistic (TSS)	33
Table 4: Importance of variables in determining the estimated foraging activity	35

LIST OF FIGURES

Figure 1: Basic patterns contained in bat echolocation calls.	18
Figure 2: (A) An elevation map of the study area with properties sampled indicated by different colored hatching. (B) A map of the detector sites (green) used for the SDM and foraging abundance models, and pseudoabsences (pink) used for the foraging abundance model.	22
Figure 3: Example of passive acoustic monitor deployment site	23
Figure 4: <i>A comparison of the $\text{Log}_{10}(\text{Bat Activity})$ by property.</i> Error bars represent standard error. Differences in means (Tukey, $\alpha = 0.05$) are represented by different letter-groups on the bars.....	31
Figure 5: Distribution of bats in Shady Valley, TN using generalized linear modeling. Cells colored red indicate areas of high habitat suitability, yellow cells indicate moderate habitat suitability, and green cells represent areas of low habitat suitability.	34
Figure 6: Foraging activity in Shady Valley, TN using Random Forest modeling. Cells colored red indicate areas of high foraging activity, yellow cells indicate moderate foraging activity, and green cells represent areas of low foraging activity.....	36
Figure 7: Environmental variables used to model species distribution and bat foraging activity	59

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
BB	John R. Dickey Birch Branch Sanctuary
CNN	Convolutional Neural Network
DEM	Digital Elevation Model
DNN	Deep Neural Network
FB	Feeding Buzz(es)
FBR	Forest-based Regression
GIS	Geographic Information Systems
GLM	Generalized Linear Model
ILSVRC	ImageNet Large Scale Visual Recognition Competition
MF	McQueen Farm
MRLC	The Multi-Resolution Land Characteristics consortium
MSE	Mean Squared Error
NLCD 2011	National Land Cover Database 2011
OB	Orchard Bog
QB	Quarry Bog
RF	Random Forest
RMSE	Root Mean Squared Error
SDM	Species Distribution Model
SS	Schoolyard Springs
TL 2019	Tiger/Line 2019
TNGIC	Tennessee Geographic Information Council

CHAPTER 1. INTRODUCTION

Background

Historically, wetland habitat was considered a nuisance¹. Standing water and saturated soil prevented agriculture on the nutrient-rich soil². The bogs and swamps were drained to access the fertile, arable land, which destroyed the ecosystem services that the wetlands were providing². Now, wetlands are known to play a vital role in the hydrologic cycle by absorbing floodwaters, filtering and removing excess nutrients and waste, such as fertilizers, and supplying water to streams during droughts³.

Before the middle of the 20th century, Shady Valley in northeast Tennessee was dominated by wild cranberry bogs that are characterized by slow-moving, acidic waters, and deep beds of sphagnum moss^{4,5}. In the 1960s, Shady Valley's wetlands were filled in, and the waterways were relocated to make way for pastures and agriculture⁶. More recently, the Nature Conservancy has begun restoring the lost wetlands in Shady Valley, and currently has three wetland sites that have been expanded to cover over 230 acres⁷. Of the wetland preserves in Shady Valley, all three contain at least one pond, either human-made or constructed by North American beavers (*Castor canadensis*), in addition to large areas of standing water⁷. The habitat is generally open and suitable for a large variety of animals and plants.

Restoration efforts in Shady Valley have been met with resistance from landowners in the region⁸. With the changes in the perceptions of wetlands in recent decades, biologists have begun researching the benefits of wetlands to farmland. This raises the question of how wetlands are directly impacting farmers in the area, particularly in quantifiable economic terms such as cleaner water, or a reduced need for pesticides. The Nature Conservancy is currently developing a protocol for quantifying the reduction in water pollution provided by their restored wetlands⁸;

however, no research is currently being conducted to determine how the wetlands impact the activity of bats.

The Shady Valley wetlands have been manipulated by human restoration practices, which places these systems somewhere between a natural wetland and a constructed wetland⁸. Farmers and municipalities often create wetlands on land that has not historically contained wetlands to filter runoff or process biological waste, and these artificial wetlands have also been found to support increased biodiversity in the area^{9,10}. Specifically, in a study conducted in England, constructed wetlands were found to significantly promote bat activity in the surrounding region¹⁰. By using acoustic monitoring and thermal imaging, Stahlschmidt et al. (2012) showed a marked increase in foraging over the retaining ponds when compared to the open area over the nearby vineyards. Notably, the study also assessed the correlation between the activity recorded on the acoustic monitors and the number of individual bats counted via thermal imaging in the area and found that in general, as the number of individuals in an area increases, the activity recorded on the acoustic monitors also increases in a relatively linear fashion¹⁰.

Large bodies of standing water seem to be a significant attractant for bats¹¹. In a previous study of Central Appalachian wetlands, researchers found that bats were only active around wetlands that contain a large pond^{11,12}. Francl et al. (2004) define a pond as a still body of acidic water with a maximum depth of at least one meter¹¹. In the wetlands at Big Run Bog in Tucker County, WV, USA, there is a 2-hectare pond, and the study found a significant increase in little brown bats (*Myotis lucifugus*) and big brown bats (*Eptesicus fuscus*), which are voracious insectivores that prey on many flying insects from earworm moths to mosquitoes¹¹. Bats have been estimated to provide billions of dollars in ecological services to the agricultural community alone¹³⁻¹⁵. Additionally, bats have been shown to reduce pest insect populations in their

foraging areas significantly and are an excellent means of naturally reducing crop pests¹⁶. As bats are natural predators of many pest insect species, increased foraging over wetlands in and around farmland would have distinct advantages to the farmers who rely on the land for their livelihood.

Project Overview

A primary aim of ecologists and conservationists is to preserve and maintain healthy ecosystems, such as wetlands, old-growth forests, and coral reefs. To that end, many ecologists study how ecosystems function and how the organisms within ecosystems interact with each other and the environment. With up to 87% of wetlands in the world lost¹⁷, many conservationists are actively researching the best way to restore wetlands and reactivate the vital ecosystem services that wetlands provide¹⁸. In Shady Valley, The Nature Conservancy has restored three large bogs by removing drainage canals installed by the U.S. Army Corps of Engineers and revitalizing the seed bank in an effort to restore the native habitat⁷. Researchers have conducted many studies that aim to explore the effects of ecological restoration projects in former wetland sites¹⁹⁻²¹; however, few studies examine the interactions between bats and newly restored wetlands^{22,23}. This study aims to determine if bats are using the wetlands in Shady Valley as foraging grounds more than the surrounding landscape.

Frequently, bat ecologists use passive acoustic monitoring to track bats and determine the level of bat activity at a given site²⁴⁻²⁷. While these types of studies provide vital information about what species of bats are present at a given site, and their relative abundance, bat activity studies using acoustic data do not elucidate how the bats are using the area. Bats could be using a given location as a roosting site, a hunting ground, or a flyway to commute between their roost and the primary foraging area. While the presence of resting bats can visually identify roosting

sites during the day, flyways and foraging areas are more challenging to recognize. Using bat activity alone (counts of echolocation passes), a flyway and a foraging area may appear very similar. Foraging bats emit search-phase calls and high-tempo feeding buzzes that indicate that the bat is approaching its prey^{28,29}. Since terminal feeding buzzes occur during the approach and attempted-capture of prey, counting feeding buzzes at a given location can allow researchers to distinguish between flyways and hunting grounds. Unfortunately, feeding buzzes make up a tiny fraction of the calls that are recorded (personal observation). Currently, manual counts are the standard technique for quantifying the level of foraging activity. Feeding buzzes are very distinct calls and can usually be identified in a spectrogram of a recorded bat call, but a single passive acoustic recorder can generate thousands of recordings per night. In this study, I retrained Google's InceptionV3 CNN³⁰ to identify call recordings containing at least one feeding buzz, thereby automating the counting process and significantly accelerating the analysis of collected data.

The massive amount of data generated by passive acoustic monitoring, and the expensive equipment required to sample a site limits the number of sites that can be sampled and analyzed. Furthermore, data collected by passive acoustic monitoring only provides information for the specific site being sampled. To infer bat activity or foraging activity at unsampled sites, species distribution models and species abundance models use statistical techniques and machine learning to find an association between the sample data and the local environment³¹. Expanding on this concept, automated feeding buzz count data collected at many sites were used to model the foraging abundance of bats in Shady Valley at a 100-meter resolution to test the following hypotheses.

Hypotheses

1. Bat activity will be higher over wetland sites containing large bodies of standing water (maximum depth > one meter, surface area > 200 square meters).
2. Bat activity over pastures and fields near wetland sites will be higher than over distant pastures and fields.
3. A convolutional neural network can be trained to identify bat feeding buzzes in bioacoustic recordings with human-level accuracy.
4. Feeding buzzes will be detected most often over the bogs, and less frequently in dry fields and forested areas.

An Overview of Machine Learning

The idea of machine learning was conceived by Arthur Samuel in 1952 when he was developing a computer system that could play chess; however, it was not until 1959 when he first used the term in the literature³². Machine learning uses computer algorithms to predict outcomes based on known inputs³³. Machine learning can be implemented in a variety of ways. Supervised learning, unsupervised learning, and reinforcement learning are the most common methods of applying machine learning³³. The best method depends upon the problem or question that needs to be addressed.

Supervised learning works best when the problem involves predicting an outcome or condition from a known set of observations or data. To solve classification problems, the researcher creates a known dataset of inputs with the matching desired outputs³³. The desired outputs are called labels. This dataset is usually divided into two parts; the first part of the dataset is used to train the computer, and the second part is used to validate that the computer is "learning." During the training phase, the computer runs the algorithm on each input and

"learns" what the output should be. After all the training data has been processed, the coefficients in the algorithm and the mathematical functions are stored and are now collectively referred to as a model.

The researcher uses the second portion of the dataset that was withheld during training to confirm the performance of a model during training. This portion of the dataset is known as the validation data. The computer then processes the inputs from the validation data through the model and compares the predicted output with the desired output to determine the model's accuracy. After training and validation have been completed, the researcher prepares another dataset with matched inputs and desired outputs that can be used to test the model. In this instance, researchers provide the computer with the input data but withhold the correct output labels. Researchers then ask the computer to predict the right outputs based on the given input data. Model validation ensures that the model can be applied to data that the model has not processed before, which is known as generalization.

A model that generalizes well can accurately make predictions or classifications based on previously unseen input data. Models that are very accurate in the training and validation phases, but fail to perform well in the testing phase, are said to be over-fit. In this case, the model has "memorized" the input-output pairs, instead of "learning" why each input matches its corresponding output. With enough training data, researchers can teach computers to analyze complex data and produce human-like predictions from the data.

Artificial Neural Networks

Researchers use many different machine learning algorithms to train computers to solve a wide variety of problems^{34,35}. In one type of problem called computer vision, researchers attempt to teach computers to interpret visual data, such as pictures or videos. In computer

vision, artificial neural networks (ANNs) are the most common category of algorithms used to process image data to accomplish tasks like image classification and image segmentation. ANNs were initially conceived to emulate the neural pathways of animal vision. Like amacrine cells, bipolar cells, horizontal cells, and ganglion cells in animal vision, ANNs consist of layers of mathematical functions called nodes that are connected, and the outputs of initial nodes are fed to subsequent nodes influencing the final prediction produced by the output layer. A simple ANN may consist of the input node, a group of intermediate nodes called a hidden layer, and the last output node that makes the predictions.

Deep Neural Networks

The width and depth of machine learning algorithms vary widely depending on the accuracy researchers require from the model and the complexity of the problem³⁶. More sophisticated models known as Deep Neural Networks (DNNs) utilize many hidden layers and are generally more capable of parsing through more complex input data than traditional ANNs³⁷. A simple model can easily classify red squares and green squares, but distinguishing cats from dogs, or faces from photographs, is much more challenging. In the first case, a model with one or two hidden layers could likely achieve near 100% accuracy very quickly. In the second case, a simple model may never perform well. To solve more complex problems, researchers add more hidden layers to the model to extract more features and details from the input data. Deep neural networks (DNNs) are models that use many layers to analyze input data before making predictions.

As DNNs increase in complexity, more processing power is required to train the models. While the average user does not possess a workstation with enough processing power to efficiently train large DNNs from scratch, technology firms and research labs like Google and

Microsoft have stepped in to compete in building the most accurate networks in image classification³⁸. Google and Microsoft regularly release new DNNs that are open-sourced and available to researchers^{30,39-41}. Many artificial intelligence teams compete to build DNN image classifiers that are progressively more capable of classifying images in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC)³⁸. These models are trained to classify images contained in the ImageNet dataset, which contains over 20,000 classes and over 14 million images³⁸.

Transfer Learning

Sometimes researchers may find themselves looking to solve many similar problems. Transfer learning allows researchers to use a previously trained model as the foundation for solving a new, comparable problem⁴². For example, if a researcher has already prepared a model to identify doors in a picture, that model can easily be retrained to identify windows instead. While learning to recognize a door, the algorithm has already learned to differentiate edges from surfaces and to look for square patterns in images. By fixing the weights in the initial layers of the algorithm, and only retraining the last couple layers, the model only needs to learn the fine details that make windows different from doors. This process is called fine-tuning a model and can significantly enhance the predictive performance on new related data when compared to training a new model from scratch⁴³.

Machine Learning in Ecology

Collecting and analyzing data for ecological studies can be labor-intensive and costly. Machine learning has been employed in a variety of ecological tasks to reduce the staff-hours required to process and analyze vast collections of data. Convolutional neural networks (CNNs)

have been used extensively. Kellenberger et al. (2018) implemented a CNN to conduct a census of large mammals of the Kuzikus wildlife reserve in Namibia⁴⁴, and Norouzzadeh et al. (2018) trained an ensemble of DNNs, including CNNs, to perform classification of images collected with camera traps from the Snapshot Serengeti dataset containing 3.2 million images⁴⁵. Using a multi-stage approach, the images from the Snapshot Serengeti dataset were classified with human-level accuracy and provided details on the presence, species identifications, and behaviors captured by camera traps⁴⁵.

CNNs are not limited to analyzing visual data. When audio data are converted in spectrograms (e.g., Figure 1), CNNs can learn from these visual representations of sound. Ecologists often seek to study animals that are difficult to track visually or are highly mobile. By studying animal vocalizations, ecologists can gain insight into the presence and behavior of elusive species. Researchers from the University of Quebec at Rimouski used a multi-layer perceptron model, the precursor to CNNs, to identify and classify calls emitted by blue whales in the Saguenay-St. Lawrence Marine Park using short-time Fourier transform spectrograms⁴⁶. Many other marine ecologists are now using machine learning to process their bioacoustic data and identify marine mammals^{47,48}.

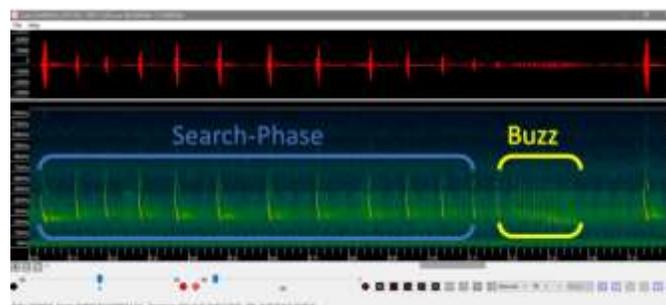


Figure 1: Basic patterns contained in bat echolocation calls

While marine ecologists search beneath the ocean waves, terrestrial ecologists are presented with similar challenges on land and in the air. Flying animals often move quickly and

occupy large ranges that make visually tracking and monitoring individuals difficult.

Ornithologists have recognized the power of machine learning and now use a variety of machine learning techniques to passively monitor birds over large swaths of habitat by recording bird songs and calls^{49,50}.

In bat research, chiroptologists rely heavily on machine learning and bioacoustic data as well. Bats, being nocturnal, flying mammals that echolocate with ultrasonic vocalizations, are especially elusive to human eyes and ears. Technology has risen to the task. Ultrasonic microphones allow researchers to “listen in” on bats at night, but the substantial amount of raw data produced can quickly become overwhelming to process manually. Computers can automatically sort recordings of bats’ high-frequency vocalizations based on the type of call using machine learning. According to Griffin et al. (1961), insectivorous bats produce different patterns of high-frequency echolocation calls when they are commuting, and when they are actively foraging²⁸. Commuting calls are typically referred to as search-phase calls, and active foraging calls are terminal buzzes or feeding buzzes²⁹. Many researchers have used machine learning to find search-phase calls in large batches of recordings quickly^{51,52}, and according to Jennings et al. (2008), computers are as accurate as experienced humans, and more accurate than humans with less than one year of experience, when deciding whether a recording contains a search-phase call or not⁵³.

Species Distribution Modeling

Species distribution modeling is a machine learning technique used by conservationists and ecologists to determine the possible current, future, or past range of a species⁵⁴. Researchers use geographic information science (GIS) software to map known occurrences or absences of a species and compare them to the environmental variables at those locations^{54,55}. Various machine

learning algorithms can then be used to predict species occurrences in new locations based on environmental variables⁵⁴. Current applications of species distribution modeling include locating new populations of rare species^{56,57}, analyzing habitat suitability to assess the impact of invasive species^{58,59}, and possible disease vectors^{60,61}, predicting the impact of climate change on species distributions^{62,63}, and many other valuable analyses.

CHAPTER 2. MATERIALS AND METHODS

Study Area and Site Selection

The study area includes Shady Valley, TN, a small community in Johnson County, TN, USA. Located between Holston Mountain and the Iron Mountains in the Southern Appalachian Mountains of East Tennessee (36.5193°, -81.9279°), Shady Valley is a high-elevation community that encompasses 140 square kilometers with elevation ranging from approximately 600 meters to 1300 meters above sea-level. While Shady Valley previously contained a massive, forested cranberry bog⁴, today, the valley contains mostly cow pastures, farm equipment storage, and residential dwellings. Five properties were selected (Figure 2A) for passive acoustic sampling. Orchard Bog, Schoolyard Springs, and Quarry Bog are sites that have been ecologically restored by The Nature Conservancy to natural wetlands. The McQueen Farm and John R. Dickey Birch Branch Sanctuary served as distant dry pasture sites. The McQueen Farm is an active cattle farm owned by the McQueen family, and the John R. Dickey Birch Branch Sanctuary is a historical mountain homestead maintained by The Nature Conservancy. Sample sites were selected by stratified random sampling. One hundred random points were plotted in ArcGIS Pro 2.5.1 (Esri) on each property using the random point tool. Ten points were then selected from the list at each property, ensuring that each point was at least 100 meters away from other selected points and 30 meters away from open bodies of water, such as ponds (Figure 2B).

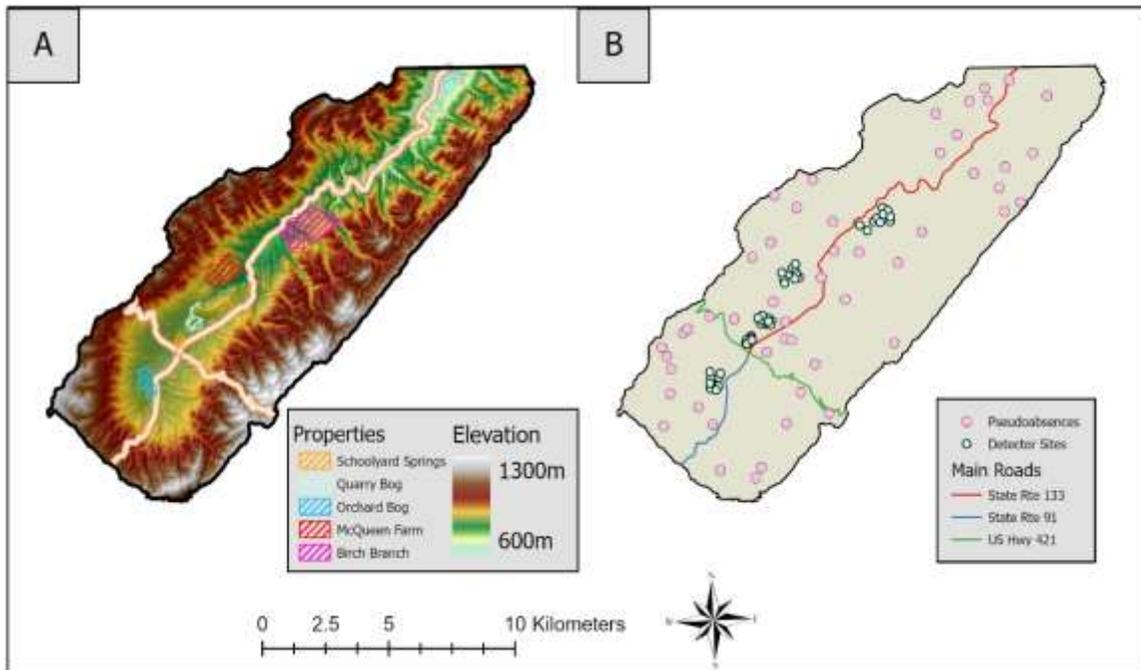


Figure 2: (A) An elevation map of the study area with properties sampled indicated by different colored hatching. (B) A map of the detector sites (green) used for the SDM and foraging abundance models, and pseudoabsences (pink) used for the foraging abundance model.

Passive Acoustic Monitor Setup and Data Collection

This study employed passive acoustic monitoring of bat species in Shady Valley, TN, from June 10th until October 16th, 2019. Ten Wildlife Acoustics SM4-Bat FS acoustic monitors with SMM-U2 microphones (Sample Rate: 256kHz, Min Trigger frequency: 20 kHz, Schedule: Sunset-30 to Sunrise+30, all other settings left at defaults) were deployed at one-week intervals with two passive acoustic monitors located at each property. Microphone towers were built using 3-meter lengths of 1/2" PVC, and three nylon rope guy lines to stabilize each microphone, as seen in Figure 3. The towers were oriented such that the microphones were aimed directly toward the open sky. When passive acoustic monitors were deployed in forested areas, the microphones were positioned and aimed toward the highest point in the canopy. Every effort was

made to ensure that passive acoustic monitors were deployed in open areas within the forest to provide the clearest audio recordings. During May, preliminary recordings were collected from 10 randomly selected sites within the study area to obtain training data for the neural network that was used to identify feeding buzzes. During the study period, two sites were selected from each of the five properties in the study area each week. A total of 50 sampling sites were monitored over five sampling weeks, and then during the second half of the study, the same sites were resampled in the same order as the first period. This process provided the opportunity to sample each site during various weather conditions, from the best weather in July and August (warm nights above 15° C) to the marginal weather in June, late September, and early October (cold nights below 15° C). Recorded bat calls were stored on 16GB SD cards and transferred to a desktop PC for analysis.



Figure 3: Example of passive acoustic monitor deployment site

Data Processing

All data were separated by sample-site before processing. As passive acoustic monitoring generates many false triggers, the scrubber tool in Kaleidoscope Pro 5.1.0 (Wildlife Acoustics) was used, which was set to include calls with a minimum of three pulses and minimum call duration of two milliseconds. The software then filtered out recordings that did not include bat-like call characteristics. A bat pass was counted for each recording that contained bat-like call characteristics, and the number of bat passes was used to assess the intensity of bat activity at each site. All recordings containing bat-like characteristics were processed with a custom Python script (python 3.7) using the Pydub module (an audio processing extension to the Python programming language) to split each recording into multiple one-second segments. Then, a second custom Python script using the Librosa module was used (an audio analysis extension to the Python programming language) to produce magnitude spectrograms of each one-second recording segment. Segments shorter than one second were not included to ensure that the resulting spectrograms were all the same 299x299 pixel dimensions without requiring temporal distortion of the spectrograms. The python code used for data processing is included in Appendix A.

Statistical Analysis

The bat activity among sites was compared using PROC ANOVA in SAS 9.4 (SAS Institute). I used an alpha level of 0.05 for all tests. First, the normality of the dataset and the bat activity at each site was assessed using the Shapiro-Wilks test of the residuals. When the data failed to pass the Shapiro-Wilks test, a log transformation was performed to normalize the distribution by taking the \log_{10} of the number of bat passes at each sample site, and the Shapiro-Wilks test was rerun. The normality of the transformed data was confirmed with Q-Q plots.

Levene's Test for Homogeneity was used to test for equal variances of the bat activity at each property. A one-way ANOVA was used to analyze the main effects of passive acoustic monitor location for the five properties in the study area (McQueen Farm, John R. Dickey Birch Branch Sanctuary, Orchard Bog, Quarry Bog, and Schoolyard Springs). The response variable was the bat activity. When the F-test indicated significant differences in bat activity among the properties, a Tukey's post-hoc analysis was performed.

Deep Neural Network: Training and Predicting

Five thousand spectrograms were randomly selected from the preliminary dataset to build an initial training dataset for the neural network. These spectrograms were manually sorted. Spectrograms containing any portion of a feeding buzz were placed into a directory named "Buzz," and all other spectrograms were placed into a directory named "Other." The sorted portion of the preliminary dataset was then loaded into TensorFlow 2.1 (Google) using flow-from-directory to generate labels for each recording. An initial CNN model was generated using transfer learning by loading Google's InceptionV3 model³⁰ from TensorFlow Hub. The Nadam optimizer⁶⁴ was used to guide the learning process by calculating the updates to the model weights after each batch of data. The early-stopping callback was used to reduce the risk of overfitting and overtraining the model. If the model's validation loss failed to improve (decrease) for five consecutive epochs, training was halted. To reduce the initial training volatility, a learning rate schedule with a 10-epoch warm-up period (learning rate = $1e-4$) was used, followed by a period of faster learning (learning rate = $5e-4$) to decrease the training time, and ending with a low learning rate ($1e-4$) to fine-tune the model weights. Next, the initial model was used to make predictions on the remaining preliminary data. All recordings classified as feeding buzzes by the initial model were manually vetted, and true positives and false positives were added to

the “buzz” folder and the “other” folder, respectively.

The InceptionV3 model was then restarted and retrained on the supplemented training data. The resulting model was used to make predictions on the remaining recordings in the preliminary dataset again, and the vetting process was repeated. This cycle was repeated from retraining InceptionV3 to vetting the results for a total of five cycles. The “other” predictions were then vetted to ensure that recordings containing clear feeding buzzes were not being misclassified. The Weights and Biases software was used to optimize the model’s hyperparameters (dropout rate, L2 regularization, and batch size) using the Bayesian optimization technique. A total of 218 models were trained, and three models were chosen based on performance. Models that achieved higher MCC scores on both the training and validation data were prioritized. Selected models were combined using the soft-voting ensemble approach in which the predictions of all three models were averaged together to generate ensemble predictions. This method emphasizes models that are more confident in their predictions. Finally, the ensemble of the selected models, hereafter referred to as BuzzFinder, was used to classify the bat recordings from the study-period as either containing a feeding buzz or not. To avoid counting feeding buzzes from the study-period dataset multiple times (in cases where the feeding buzz spanned multiple recording segments), the filename suffixes (e.g. “SEG_1”) were removed, and unique base filenames were counted when generating the final tally of recordings containing at least one feeding buzz. The TensorFlow code used to produce the neural networks is included in Appendix B.

Modeling the Distribution of Bats

A species distribution model (SDM) showing the distribution of bats was created for Shady Valley, TN, based on the site locations where any species of bat was recorded during the

study. Based on passive acoustic sampling, bats were present at each site. With a small, mountainous study area, macroclimatic data would be ineffective in modeling the actual distribution of bats at high resolution, because of the high variability of weather patterns over short distances in the mountains⁶⁵. Since the most detailed climatic data currently available for Shady Valley is only approximately one-kilometer resolution, topographic features (aspect, topographic wetness index), Euclidean distances to crucial habitat features (buildings, roads, streams, forest edge), and land cover data were used as recommended by Jaberg and Guisan^{66,67}. Details about the creation of the individual environmental variables can be found in Appendix C.

Using the Biomod2 package⁶⁸ in R, a generalized linear model (GLM) of the distribution of bats in Shady Valley was produced. A total of five GLM models were trained. Each model was projected over the study area, and then a consensus model was constructed by combining the outputs of all five models into one ensemble by averaging the predictions. Finally, the ensemble model was exported to ArcGIS Pro in ASCII format for use in modeling the potential foraging activity of bats in Shady Valley.

Modeling Bat Foraging Activity

The Forest-Based Regression (FBR) tool in ArcGIS Pro was used to create a foraging abundance model to predict the bat foraging activity in Shady Valley. The FBR tool is based on Leo Breiman's Random Forests algorithm and performs well when environmental variables are highly correlated⁶⁹. When training a regression model, the number of pseudoabsences should match the number of data points⁷⁰. Since bat foraging activity data were collected at 50 sample sites, 50 random pseudoabsences were generated by the random point generator tool in ArcGIS Pro using a buffer of 400 meters around detector sites to prevent pseudoabsences from being generated near sites known to support foraging activity. The feeding buzz counts given by the

BuzzFinder model were used to calculate the average foraging activity (feeding buzzes per night) observed at each site by dividing the number of recordings containing feeding buzzes by the number of nights that the passive acoustic monitor was deployed at each site. FBR was then used to analyze the relationship between the aspect, topographic wetness index (TWI), Euclidean distances (to roads, streams, forest edge, and buildings), the likelihood of bat presence, and the observed foraging activity at my study sites. I employed the grid-search technique to tune the FBR model hyperparameters. The best model was produced with 250 trees, a maximum depth of five, and minimum leaf size of 20. The number of randomly sampled environmental variables included in each tree was set to two. Based on these relationships, the model predicted the average number of recordings containing feeding buzzes per night within each 100-meter cell in the study area. The mean squared error was calculated by the FBR and converted to root mean squared error (RMSE) for model evaluation.

CHAPTER 3. RESULTS

Passive Acoustic Monitoring

During preliminary data collection, 19,613 audio recordings were collected on the passive acoustic monitors. The noise files were not scrubbed from the preliminary dataset to preserve some files that did not contain any bat-like call characteristics for use in training the BuzzFinder model. After the preliminary recordings were processed with the WavChop script (Appendix A), 149,496 recordings were converted into magnitude spectrograms using the custom generate_spectrograms script.

The number of files in each dataset after each processing step can be seen in Table 1. Over ten weeks, and 679 sample nights, the passive acoustic monitors were triggered 149,522 times. After removing the probable noise files with the Kaleidoscope Pro scrubber, 100,917 recordings contained bat-like call characteristics. These files were then segmented into 889,716, one-second audio segments after discarding short clips. One audio file from the study-period dataset was corrupted and could not be converted into a readable spectrogram.

Table 1: File counts after each step of data processing. Note: Noise files were not scrubbed from the preliminary dataset in order to preserve noise files for training the neural network.

<u>Dataset Status</u>	<u>Preliminary Dataset</u>	<u>Study-Period Dataset</u>	<u>Total</u>
Raw Audio	19,613	149,522	169,135
Scrubbed with Kaleidoscope Pro	19,613	100,917	120,530
Processed with WavChop script	149,496	889,716	1,039,212
Spectrograms	149,496	889,715	1,039,211

During the study period, three microphones were disturbed or lost power and failed to produce data. During week two of the study, site BB4 was knocked down during a storm, and site OB8, in Orchard Bog, was taken down by cattle. At the McQueen Farm, site MF1 failed to

record data during week four because of a battery failure. Also, sites BB1, MF8, QB1, QB2, and SS1 suffered battery failures and lost power resulting in a shortened sampling period during week 6.

Apart from difficulties in collecting data at some sites, other sites produced bat activity that was significantly higher than expected. Site MF6 and OB2 experienced abnormally high levels of bat activity during weeks three and seven, respectively. Data from these sites, while not statistical outliers (>2.5 times the inner quartile range), were much higher than the mean bat activity collected on their respective properties. Since cattle attract many insects^{71,72}, the bat activity at these sites may have been influenced by the presence of cows during the sampling periods covered by those instruments.

Statistical Analysis of Bat Activity

Bat activity was quantified by the number of bat passes recorded on each property. The log-transformed bat activity was significantly different among the five properties ($F_{92,4} = 3.91$, $p = 0.0056$) (Figure 4). Based on the Tukey's post hoc analysis, the estimated mean bat activity at John R. Dickey Birch Branch Sanctuary (581.00 ± 161.05 passes/night, mean \pm SE) was significantly different from Orchard Bog (1363.05 ± 315.51 passes/night), Quarry Bog (1200.80 ± 241.25 passes/night), and Schoolyard Springs (1331.30 ± 246.84 passes/night). Bat activity at John R. Dickey Birch Branch Sanctuary was significantly lower than at the wetland properties. While the estimated mean bat activity at the McQueen Farm (702 ± 184.26 passes/night) was lower than at the three wetland properties, the difference was not statistically significant. The mean bat activity at the McQueen Farm was also not significantly different from the mean bat activity at John R. Dickey Birch Branch Sanctuary. Orchard Bog, Quarry Bog, and Schoolyard Springs estimates were not significantly different.

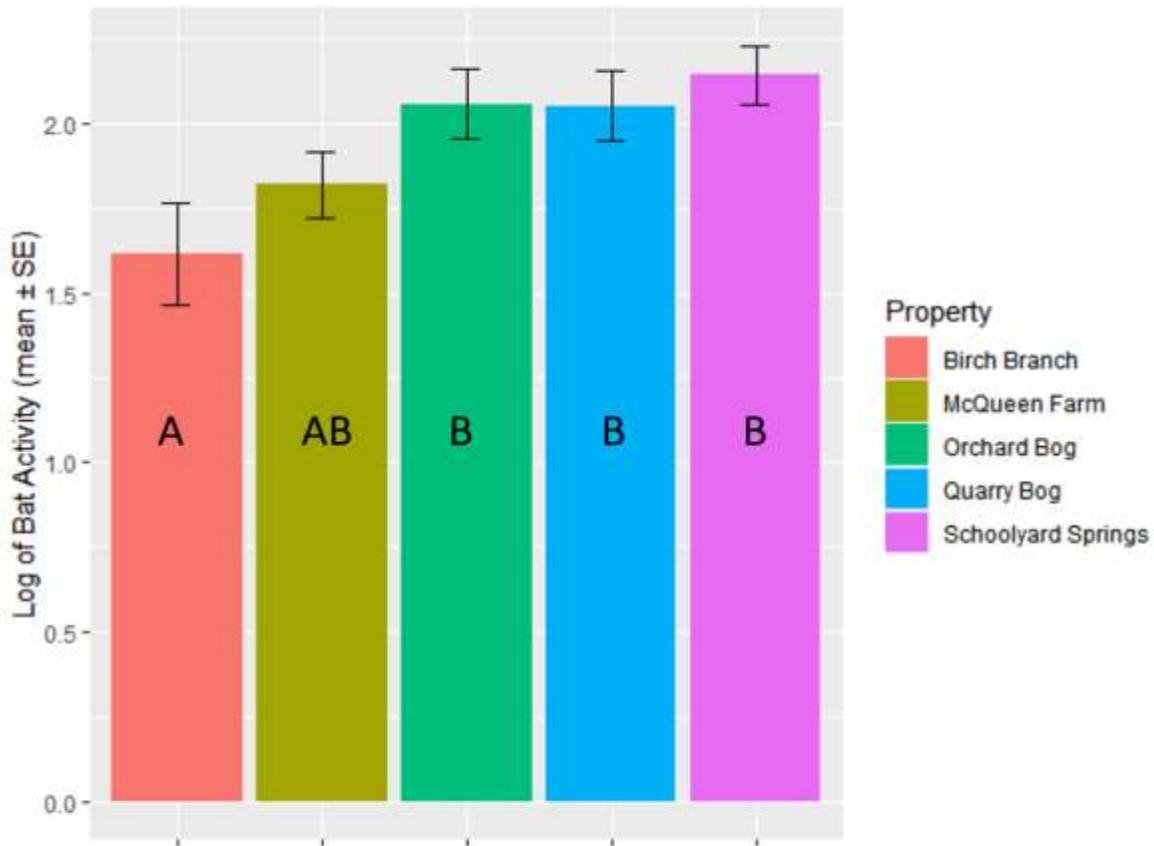


Figure 4: A comparison of the $\text{Log}_{10}(\text{Bat Activity})$ by property. Error bars represent standard error. Differences in means (Tukey, $\alpha = 0.05$) are represented by different letter-groups on the bars.

Google's InceptionV3 Model Retrained

To train the BuzzFinder model to identify bat feeding buzzes visualized in spectrograms, a total of 5,000 spectrograms were manually sorted, and over 140,000 additional spectrograms were vetted from the preliminary passive acoustic monitoring dataset. Seven hundred eighty-three feeding buzzes were identified in the preliminary dataset. To build a training dataset containing 10,000, the feeding buzzes were oversampled to create 5,000 positive training samples, and under-sampling was employed to randomly select 5,000 non-feeding buzz recordings to represent the negative class. Using the Bayesian hyperparameter sweep tool in the

Weights and Biases web application (Weights and Biases AI, San Francisco, CA), 218 unique models were retrained, using Google’s InceptionV3 model as a base model. Matthew’s

Correlation Coefficients for the three best models are listed in Table 2.

Table 2: Individual CNN model performance statistics. The training and validation performances were evaluated on the split training dataset. The testing performance was evaluated base on 10,000 randomly selected files from the study-period dataset.

Model Name	Matthew's Correlation Coefficient (MCC)		
	Training	Validation	Testing
Classic-Sweep-11	0.8292	0.9574	0.7725
Good-sweep-1	0.8366	0.9829	0.894
silvery-sweep-19	0.8233	0.9734	0.8501

Species Distribution Model

The sampled distribution of bats in Shady Valley includes restored wetlands on the west-central valley floor, pastures in the foothills near Winchester Road, and natural open fields and woodland toward the north end of the valley at Birch Branch Road (Figure 2). Because of Shady Valley’s slope with higher elevations in the south and lower valley floor in the north, all sample sites were located at similar elevation (700-800 meters above sea-level) despite the McQueen Farm and John R. Dickey Birch Branch Sanctuary properties being much higher off the valley floor and closer to the mountain ridges.

The SDM (Figure 5) performed well, although it appears to overfit slightly. All original models displayed a moderate predictive capability ($AUC > 0.75$) (Table 3). The true skill statistics ($true\ positive\ rate + true\ negative\ rate - 1$) for the original models are also very good ($TSS > 0.5$)⁷³ indicating that the model is often correctly predicting whether or not bats are present in a given area (Table 3). Overall, the ensemble model resulted in good predictions and was able to capture most of the presence points within areas estimated to have a high likelihood of presence. The habitat suitability was highest in the southern end of the valley on the open

valley floor, excluding the most developed region at the crossroads of Tennessee State Routes 91 and 133, and United States Highway 421. The north end of Shady Valley shows a moderate likelihood of hosting bat species, although not as probable as the south. It is also important to note that all three wetland properties are shown to have a high likelihood of bat presence; however, the model predicts that John R. Dickey Birch Branch Sanctuary is much less suitable for bats. The McQueen Farm shows a low probability of presence on the west side of the property but higher suitability on the east side of the property near the creek and small pond.

Table 3: Performance metrics of original GLM models. The area under the receiver operator curve (AUC) and True skill statistic (TSS)

Run	AUC	TSS
1	0.871	0.693
2	0.891	0.695
3	0.815	0.624
4	0.751	0.520
5	0.897	0.698

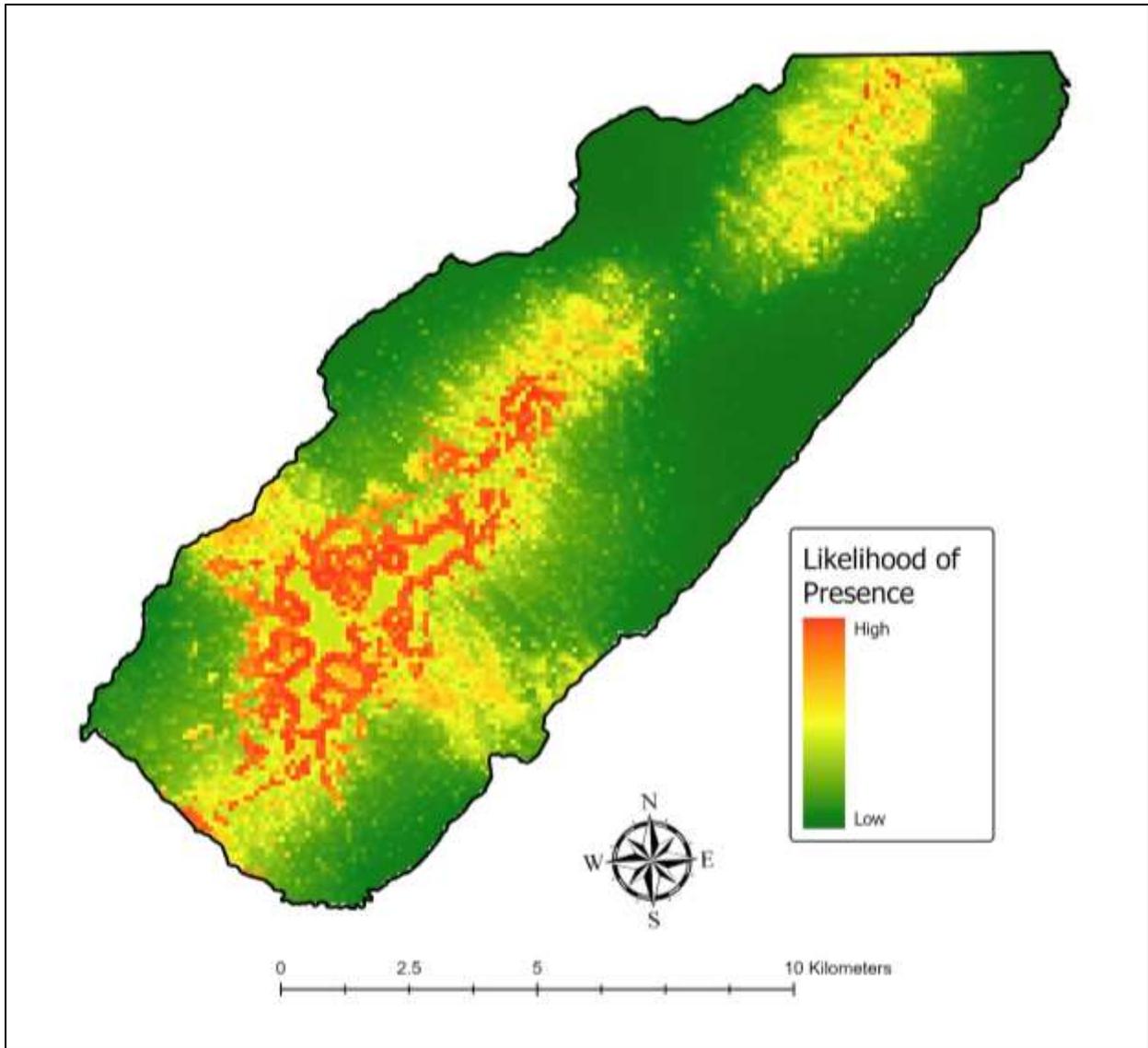


Figure 5: Distribution of bats in Shady Valley, TN using generalized linear modeling. Cells colored red indicate areas of high habitat suitability, yellow cells indicate moderate habitat suitability, and green cells represent areas of low habitat suitability.

Foraging Activity Model

The foraging activity model (Figure 6) appears to perform well (RMSE = 3.72 FB/night), capturing general trends in the foraging areas of Shady Valley. The model accounts for 12.26% of the variation within the data. The likelihood of presence was the most critical factor in predicting the foraging activity of bats in Shady Valley, followed by the TWI, and aspect. The list of variable importance is shown in Table 4. The areas around the restored wetlands and along the western valley floor are predicted to have the highest foraging activity, while heavily wooded areas are predicted to have minimal foraging activity. Interestingly, the model predicts that foraging activity will be slightly higher along creeks and streams when compared to the surrounding woodland. The predicted foraging activity appears very similar among the wetland sites. The foraging activity model predicts that foraging activity will be minimal or absent at the McQueen Farm and John R. Dickey Birch Branch Sanctuary. The foraging abundance model generally follows the same overall distribution as the SDM with areas of a high likelihood of presence also being more likely to exhibit high levels of foraging activity.

Table 4: Importance of variables in determining the estimated foraging activity

Variable	Importance	%
Likelihood of Presence	214.98	25
Topographical Wetness Index	173.63	20
Aspect	128.35	15
Euclidean Distance to Forest	112.48	13
Euclidean Distance to Streams	86.03	10
Euclidean Distance to Buildings	79.78	9
Euclidean Distance to Roads	75.74	9

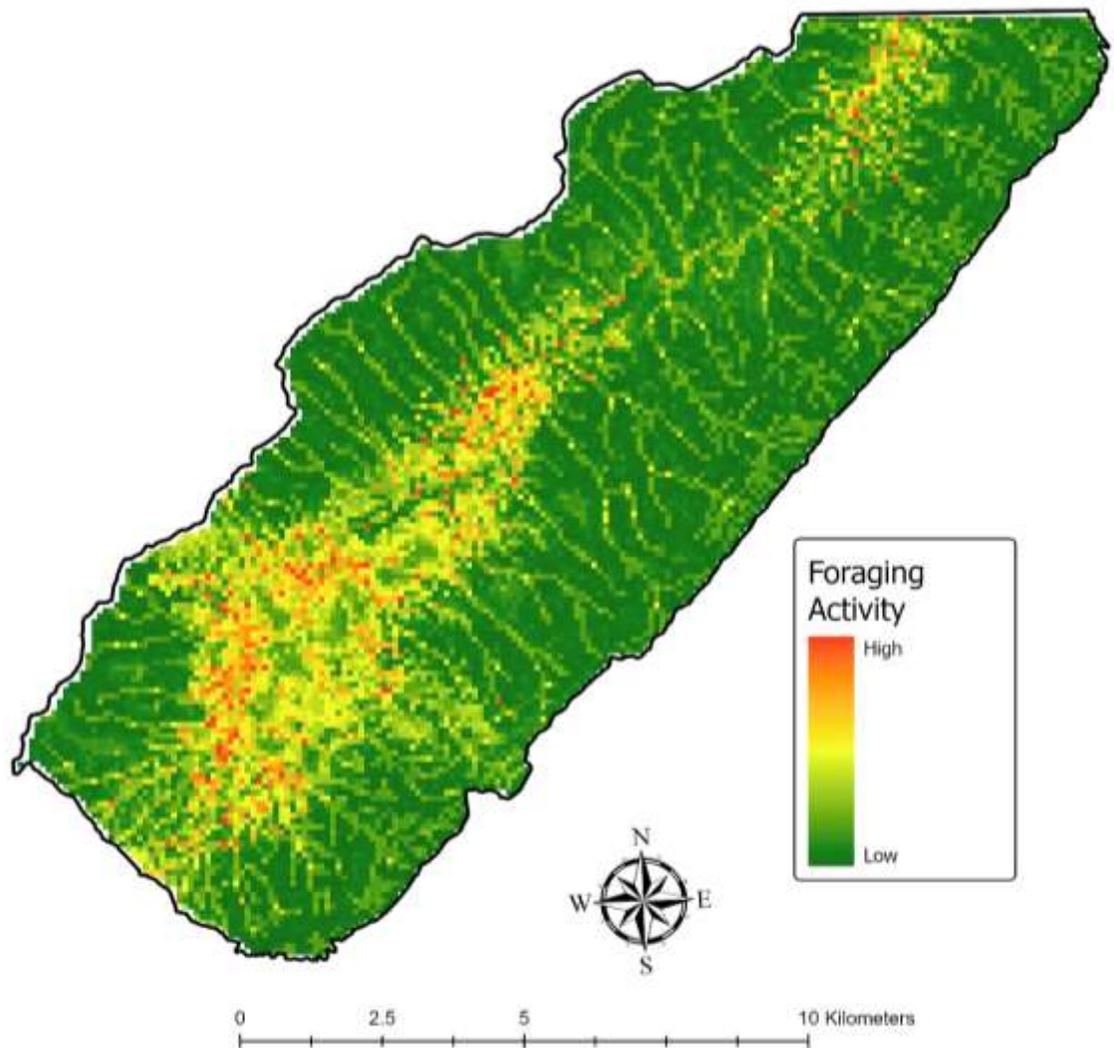


Figure 6: Foraging activity in Shady Valley, TN using Random Forest modeling. Cells colored red indicate areas of high foraging activity, yellow cells indicate moderate foraging activity, and green cells represent areas of low foraging activity.

CHAPTER 4. DISCUSSION

Bat Activity Among the Wetland Properties

At the start of this study, I hypothesized that the wetlands at Orchard Bog would have higher bat activity because of the two large, open-water ponds on the property. This hypothesis is not supported by the data recorded and analyzed in this study. Based on the one-way ANOVA and Tukey post hoc analysis, there was no significant difference in bat activity when the three wetland sites are compared. This finding agrees with the SDM of bat presence in Shady Valley. The SDM (Figure 5) predicts that all three wetland properties have a high probability of hosting bat species because of the favorable habitat. As bats often prefer lacustrine habitats for foraging areas¹¹, the presence of Beaverdam Creek along the edge of each of the wetland properties may be influencing the overall bat activity in the area. Alternatively, it could be that all ponds located within the Shady Valley wetlands are large enough to promote bat activity. Francl et al. (2004) suggested that large ponds are necessary for wetlands to support bat foraging activity; however, no studies have been conducted to determine the minimum pond size required. This information could be vital in helping farmers determine the optimal size of their constructed wetland ponds and retention ponds to reap the benefits of water runoff filtration while also minimizing the increase in pest insects. Increased bat activity over retention ponds may reduce the need for dosing pesticides, or kerosene, to control mosquitoes.

Bat Activity of Near-Wetland Fields Versus Distant Fields

When comparing the bat activity of the wetlands and the distant fields at McQueen Farm and John R. Dickey Birch Branch Sanctuary, higher activity is expected in the wetlands because of the increased optimal habitat for insect proliferation. While this hypothesis is supported when comparing the wetlands to BB, the bat activity at the McQueen farm was not significantly

different from the bat activity at any of the wetland properties. This is likely a result of the combination of active cow pastures at MF and the presence of multiple retention ponds, a small natural pond, and the McQueen Branch creek running through the property. Active cow pastures are known to promote bat activity because of the increase in insects that congregate around cows⁷¹. Also, McQueen Branch creek likely increases the quality of habitat available for foraging bats. On the East side of the property, the likelihood of bats being present is higher than on the West side of MF, based on the SDM (Figure 5). As MF contains habitat in the west that is slightly similar to the habitats at the wetlands and very similar to the fields at BB on the east side of the property, the lack of a significant difference between MF and all other studied properties is not surprising.

Bat activity at John R. Dickey Birch Branch Sanctuary is significantly lower than the bat activity at any of the three wetland sites based on the Tukey post hoc analysis. This finding is also reflected in the SDM (Figure 5), which shows that the habitat suitability and likelihood of bat species being present are very low when compared to the wetland sites. While there are a few fragmented patches of habitat that may be suitable for a sizeable population of bats, the habitat overall is not conducive to success for foraging bats.

Efficacy of Identifying Feeding Buzzes with Machine Learning

Manually sorting bioacoustic data can be very time-consuming. To make future studies on bat foraging activity more efficient, a convolutional neural network was developed to automate the process of identifying and counting bat feeding buzzes contained in large bioacoustic datasets. The overall performance of the BuzzFinder model is outstanding (Table 2). The three models produced in this study, when combined using a soft-voting system, could differentiate between feeding buzzes and other high-frequency sounds in most audio recordings

collected during the study. The high MCC scores indicate that there was a strong agreement between the predictions the model made and the labels that were applied during human classification. Since other studies have found that machine learning can be used to identify search-phase calls reliably^{52,53}, it is not surprising that computers can also be trained to recognize feeding buzzes.

In some cases, BuzzFinder was able to identify feeding buzzes that were faintly visible in the spectrograms and were missed during the initial manual classification process. These results may not translate to other study areas. CNN's are highly data-dependent, and more massive datasets tend to produce more robust models that make better predictions on unseen data. In the dataset from this study, feeding buzzes were quite rare, being present in approximately 2.7% of the calls recorded, but that is not always the case. Some studies, as in Gillam's (2007) study on the foraging behavior of Brazilian free-tailed bats, suggest that passive acoustic monitors are much more likely to capture some species' feeding buzzes than others⁷⁴. This is because of the high interspecific variability in the amplitude of bat calls, making some species more detectable than others^{75,76}. The frequency of bat calls also influences detectability; in that high frequency, low energy sounds do not travel as far as low frequency, high energy sounds. This means that the area around an acoustic monitor from which a low-frequency bat call may be detected is much larger than the area that a high-frequency call may be detected. While the BuzzFinder model was very accurate for the Shady Valley dataset, adding more audio recordings from other regions containing different bat species would likely increase the model's generalization capability and make the model more useful as a universal tool to quantify bat foraging activity quickly.

Foraging Activity over Near-Wetland Fields Versus Distant Fields

Bat foraging intensity was expected to be significantly higher near the wetlands and lower in the drier regions of Shady Valley. Based on the foraging activity model (Figure 6), this hypothesis is generally supported. The foraging activity model predicts that the foraging activity will be highest along the valley floor near the western wood line and around the wetlands. The foraging activity is also predicted to be much less at John R. Dickey Birch Branch Sanctuary and in all the forested, higher-elevation terrain surrounding Shady Valley. The foraging activity model expects bats to forage in areas that tend to collect water (indicated by a high TWI value) and have a high likelihood of hosting bat species. Oddly, the eastern valley floor is predicted to have a high probability of hosting bat species, but the foraging activity model identifies very few cells with high foraging activity predictions. The overall performance of the model was fair, with the RMSE indicating that the average error was approximately 30%. The amount of variation explained by the model is low.

This model could benefit from more data points and better environmental variables that could allow the model to capture the high variability in bat foraging behavior. While the SDM, the TWI, and the aspect seem to provide some explanation of the variability in the data, the Euclidean distance variables failed to contribute much to the model's ability to predict feeding buzz counts at the detector sites. As bats are highly mobile species, the locations of potential roost sites and unfavorable landscape features seem to have little relevance to the distribution or foraging activity of bats in a 100-meter resolution model. Additional environmental variables (i.e., microclimatic variables, such as minimum low temperature, temperature seasonality, vapor pressure, and humidity) and data points may be expected to produce a better model with higher predictive capability.

Conclusions

Wetlands provide many benefits when maintained near farmland. Crop farms use a significant amount of fertilizers to increase crop yield, and cattle farms produce large quantities of manure, both of which often end up in the groundwater, streams, and rivers via runoff. This pollution contributed to a variety of negative ecological consequences, such as eutrophication⁷⁷. Since wetlands provide an excellent means of preventing non-point source pollution from entering the waterways⁷⁸, preserving wetlands near the farms may significantly reduce the amount of agricultural pollution that reaches the groundwater, streams, and ultimately, the ocean.

While cleaner water is a boon to humanity, pest insect control is beneficial to crop farmers, specifically. Bats are known to eat many pest insects, such as the corn earworm (*Helicoverpa zea*), that decimate crops and could cost the agricultural community billions of dollars annually if left unchecked¹³⁻¹⁵. Cattle farms do not appear to reap all the benefits that crop farms may, such as increased bat foraging. As wetlands and active pastures contain large insect populations, wetlands and active cow pastures most likely compete to become the prime foraging areas that bats seek. Cows draw many prey species that feed bat populations⁷². However, as the number of livestock fluctuates, wetlands may act as a reservoir for times when the bat population exceeds the ability of drier farms to sustain them.

In addition to threatening livestock and crops, mosquitoes and other disease vectors threaten humans with diseases like the Zika virus and malaria. Since many of these dangerous insects are capable of breeding in small pools of water, maximizing bat foraging activity over farmland by converting retention ponds to constructed wetlands may serve to limit human exposure to these pests without requiring the application of pesticides while also providing the many functions that retention ponds impart, such as offering reserve water sources used to

irrigate crops, water cattle and collecting runoff before it enters the waterways. While many methods have been proposed to naturally reduce pest insects on crop and cattle farms⁷⁹, eliminating reliance on pesticides will probably require implementing multiple mitigation techniques. Bats are not capable of reducing all pest insect populations, but when combined with other techniques, such as push-pull and biochemical methods, maintaining wetlands near farmland has the potential to reduce the amount of pesticides required and limit agricultural pollution.

REFERENCES

1. Wood HG. A practical treatise on the law of nuisances in their various forms: including remedies therefore at law and in equity. Albany, N.Y.: J.D. Parsons, Jr.; 1875.
2. Dahl TE, Allord GJ. History of Wetlands in the Conterminous United States. National Water Summary on Wetland Resources. 1996;(2425):19–26.
3. US EPA O. Classification and Types of Wetlands. US EPA. 2015 Apr 9 [accessed 2018 Oct 29]. <https://www.epa.gov/wetlands/classification-and-types-wetlands>
4. Davis DE. Where There Are Mountains: An Environmental History of the Southern Appalachians. Athens: University of Georgia Press; 2005.
5. Society NG. bog. National Geographic Society. 2012 Sep 12 [accessed 2018 Dec 13]. <http://www.nationalgeographic.org/encyclopedia/bog/>
6. Wild Side TV. Wild SideTV-Shady Valley Bog Restoration.mpg. 2011. https://www.youtube.com/watch?v=vGLZv8M0L8w&list=PLnN7OGI1bJ88pzzjPYF-U6pgYTwj_8nHe&index=33
7. Shady Valley Program. [accessed 2018 Nov 26]. <https://www.nature.org/content/dam/tnc/nature/en/documents/shadyvalleybrochure-small.pdf>
8. Lynch G. Of Wolves & Turtles: One Woman’s Journey as a Conservation Biologist in Northeast Tennessee. 2018.
9. Ghermandi A, Bixio D, Thoeys C. The Role of Free Water Surface Constructed Wetlands as Polishing Step in Municipal Wastewater Reclamation and Reuse. Science of The Total Environment. 2007;380(1–3):247–258. doi:10.1016/j.scitotenv.2006.12.038
10. Stahlschmidt P, Pätzold A, Ressler L, Schulz R, Brühl CA. Constructed Wetlands Support Bats in Agricultural Landscapes. Basic and Applied Ecology. 2012;13(2):196–203. doi:10.1016/j.baae.2012.02.001
11. Francl KE, Ford WM, Castleberry SB. Bat Activity in Central Appalachian Wetlands. Georgia Journal of Science. 2004;62(2):87–94.
12. Zimmerman GS, Glanz WE. Habitat Use by Bats in Eastern Maine. The Journal of Wildlife Management. 2000;64(4):1032–1040. doi:10.2307/3803214
13. Boyles JG, Cryan PM, McCracken GF, Kunz TH. Economic Importance of Bats in Agriculture. Science. 2011;332(6025):41–42.

14. Cleveland CJ, Betke M, Federico P, Frank JD, Hallam TG, Horn J, López JD, McCracken GF, Medellín RA, Moreno-Valdez A, et al. Economic Value of the Pest Control Service Provided by Brazilian Free-Tailed Bats in South-Central Texas. *Frontiers in Ecology and the Environment*. 2006;4(5):238–243.
15. Federico P, Hallam TG, McCracken GF, Purucker ST, Grant WE, Correa-Sandoval AN, Westbrook JK, Medellín RA, Cleveland CJ, Sansone CG, et al. Brazilian Free-Tailed Bats as Insect Pest Regulators in Transgenic and Conventional Cotton Crops. *Ecological Applications*. 2008;18(4):826–837. doi:10.1890/07-0556.1
16. Tscharrntke T, Clough Y, Wanger TC, Jackson L, Motzke I, Perfecto I, Vandermeer J, Whitbread A. Global Food Security, Biodiversity Conservation and the Future of Agricultural Intensification. *Biological Conservation*. 2012;151(1):53–59. doi:10.1016/j.biocon.2012.01.068
17. Ramsar Convention on Wetlands. *Global Wetland Outlook: State of the world's wetlands and their services to people 2018*. Gland, Switzerland: Ramsar Convention Secretariat; 2018. https://static1.squarespace.com/static/5b256c78e17ba335ea89fe1f/t/5ca36fb7419202af31e1de33/1554214861856/Ramsar+GWO_ENGLISH_WEB+2019UPDATE.pdf
18. Jenkins WA, Murray BC, Kramer RA, Faulkner SP. Valuing ecosystem services from wetlands restoration in the Mississippi Alluvial Valley. *Ecological Economics*. 2010;69(5):1051–1061. doi:10.1016/j.ecolecon.2009.11.022
19. Galatowitsch SM, Valk AG van der. The Vegetation of Restored and Natural Prairie Wetlands. *Ecological Applications*. 1996;6(1):102–112. doi:10.2307/2269557
20. Lehtinen RM, Galatowitsch SM. Colonization of Restored Wetlands by Amphibians in Minnesota. *The American Midland Naturalist*. 2001;145(2):388–396. doi:10.1674/0003-0031(2001)145[0388:CORWBA]2.0.CO;2
21. Kahara S, Duffy W, DiGaudio R, Records R. Climate, Management and Habitat Associations of Avian Fauna in Restored Wetlands of California's Central Valley, USA. *Diversity*. 2012;4(4):396–418. doi:10.3390/d4040396
22. Menzel JM, Menzel MA, Kilgo JC, Ford WM, Edwards JW. Bat response to Carolina bays and wetland restoration in the southeastern U.S. Coastal Plain. *Wetlands*. 2005;25(3):542–550. doi:10.1672/0277-5212(2005)025[0542:BRTCBA]2.0.CO;2
23. Straka TM, Lentini PE, Lumsden LF, Wintle BA, Ree R van der. Urban bat communities are affected by wetland size, quality, and pollution levels. *Ecology and Evolution*. 2016;6(14):4761–4774. doi:10.1002/ece3.2224

24. Parker KA, Springall BT, Garshong RA, Malachi AN, Dorn LE, Costa-Terryll A, Mathis RA, Lewis AN, MacCheyne CL, Davis TT, et al. Rapid Increases in Bat Activity and Diversity after Wetland Construction in an Urban Ecosystem. *Wetlands*. [https://link.springer.com/article/10.1007/s13157-018-1115-5](https://link.springer.com/article/10.1007/s13157-018-1115-5?utm_source=researcher_app&utm_medium=referral&utm_campaign=RESR_MRKT_Researcher_inbound)?utm_source=researcher_app&utm_medium=referral&utm_campaign=RESR_MRKT_Researcher_inbound. doi:10.1007/s13157-018-1115-5
25. Reichert BE, Lausen C, Loeb S, Weller T, Allen R, Britzke E, Hohoff T, Siemers J, Burkholder B, Herzog C, et al. A guide to processing bat acoustic data for the North American Bat Monitoring Program (NABat). 2018. doi:10.3133/ofr20181068
26. Amorim F, Carvalho SB, Honrado J, Rebelo H. Designing Optimized Multi-Species Monitoring Networks to Detect Range Shifts Driven by Climate Change: A Case Study with Bats in the North of Portugal. *PLoS ONE*. 2014;9(1):e87291. doi:10.1371/journal.pone.0087291
27. Jones G, Jacobs D, Kunz T, Willig M, Racey P. Carpe noctem: the importance of bats as bioindicators. *Endangered Species Research*. 2009;8:93–115. doi:10.3354/esr00182
28. Griffin DR, Webster FA, Michael CR. The echolocation of flying insects by bats. *Animal Behaviour*. 1960;8(3):141–154.
29. Schnitzler H-U, Kalko EKV. Echolocation by Insect-Eating Bats. *BioScience*. 2001;51(7):557. doi:10.1641/0006-3568(2001)051[0557:EBIEB]2.0.CO;2
30. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the Inception Architecture for Computer Vision. arXiv:1512.00567 [cs]. 2015 Dec 11 [accessed 2020 Jun 14]. <http://arxiv.org/abs/1512.00567>
31. Elith J, Leathwick JR. Species Distribution Models: Ecological Explanation and Prediction Across Space and Time. *Annual Review of Ecology, Evolution, and Systematics*. 2009;40(1):677–697. doi:10.1146/annurev.ecolsys.110308.120159
32. Samuel AL. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*. 1959;3(3):210–229. doi:10.1147/rd.33.0210
33. Jordan MI, Mitchell TM. Machine learning: Trends, perspectives, and prospects. *Science*. 2015;349(6245):255–260. doi:10.1126/science.aaa8415
34. Nichols JA, Herbert Chan HW, Baker MAB. Machine learning: applications of artificial intelligence to imaging and diagnosis. *Biophysical Reviews*. 2018;11(1):111–118. doi:10.1007/s12551-018-0449-9

35. Kersting K. Machine Learning and Artificial Intelligence: Two Fellow Travelers on the Quest for Intelligent Behavior in Machines. *Frontiers in Big Data*. 2018 [accessed 2020 Jul 6];1. <https://www.frontiersin.org/articles/10.3389/fdata.2018.00006/full>. doi:10.3389/fdata.2018.00006
36. Lu Z, Pu H, Wang F, Hu Z, Wang L. The Expressive Power of Neural Networks: A View from the Width. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R, editors. *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc.; 2017. p. 6231–6239. <http://papers.nips.cc/paper/7203-the-expressive-power-of-neural-networks-a-view-from-the-width.pdf>
37. Krizhevsky A, Sutskever I, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ, editors. *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc.; 2012. p. 1097–1105. <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
38. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, et al. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*; New York. 2015;115(3):211–252. doi:<http://dx.doi.org/10.1007/s11263-015-0816-y>
39. Szegedy C, Ioffe S, Vanhoucke V, Alemi A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. arXiv:1602.07261 [cs]. 2016 Aug 23 [accessed 2020 Mar 5]. <http://arxiv.org/abs/1602.07261>
40. Xie Q, Luong M-T, Hovy E, Le QV. Self-training with Noisy Student improves ImageNet classification. arXiv:1911.04252 [cs, stat]. 2020 Apr 28 [accessed 2020 Jun 14]. <http://arxiv.org/abs/1911.04252>
41. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. arXiv:1512.03385 [cs]. 2015 Dec 10 [accessed 2020 Jun 14]. <http://arxiv.org/abs/1512.03385>
42. Pratt LY, Mostow J, Kamm CA. Direct Transfer of Learned Information Among Neural Networks. In: *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91)*. Anaheim, CA; 1991. p. 584–589. <https://www.aaai.org/Papers/AAAI/1991/AAAI91-091.pdf>
43. Huh M, Agrawal P, Efros AA. What makes ImageNet good for transfer learning? arXiv:1608.08614 [cs]. 2016 Dec 10 [accessed 2020 Jun 22]. <http://arxiv.org/abs/1608.08614>
44. Kellenberger B, Marcos D, Tuia D. Detecting Mammals in UAV Images: Best Practices to address a substantially Imbalanced Dataset with Deep Learning. *Remote Sensing of Environment*. 2018;216:139–153. doi:10.1016/j.rse.2018.06.028

45. Norouzzadeh MS, Nguyen A, Kosmala M, Swanson A, Palmer MS, Packer C, Clune J. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proceedings of the National Academy of Sciences*. 2018;115(25):E5716–E5725. doi:10.1073/pnas.1719367115
46. Bahoura M, Simard Y. Blue whale calls classification using short-time Fourier and wavelet packet transforms and artificial neural network. *Digital Signal Processing*. 2010;20(4):1256–1263. doi:10.1016/j.dsp.2009.10.024
47. Esfahanian M, Erdol N, Gerstein E, Zhuang H. Two-stage detection of north Atlantic right whale upcalls using local binary patterns and machine learning algorithms. *Applied Acoustics*. 2017;120:158–166. doi:10.1016/j.apacoust.2017.01.025
48. Shiu Y, Palmer KJ, Roch MA, Fleishman E, Liu X, Nosal E-M, Helble T, Cholewiak D, Gillespie D, Klinck H. Deep neural networks for automated detection of marine mammal species. *Scientific Reports*. 2020;10(1):607. doi:10.1038/s41598-020-57549-y
49. Boulmaiz A, Messadeg D, Doghmane N, Taleb-Ahmed A. Robust acoustic bird recognition for habitat monitoring with wireless sensor networks. *International Journal of Speech Technology*. 2016;19(3):631–645. doi:10.1007/s10772-016-9354-4
50. Fazeka B, Schindler A, Lidy T, Rauber A. A Multi-modal Deep Neural Network approach to Bird-song identification. arXiv:1811.04448 [cs, eess]. 2018 Nov 11 [accessed 2020 Jun 11]. <http://arxiv.org/abs/1811.04448>
51. Bertran M, Alsina-Pagès RM, Tena E. Pipistrellus pipistrellus and Pipistrellus pygmaeus in the Iberian Peninsula: An Annotated Segmented Dataset and a Proof of Concept of a Classifier in a Real Environment. *Applied Sciences*. 2019;9(17):3467. doi:10.3390/app9173467
52. Mac Aodha O, Gibb R, Barlow KE, Browning E, Firman M, Freeman R, Harder B, Kinsey L, Mead GR, Newson SE, et al. Bat detective—Deep learning tools for bat acoustic signal detection Fenton B, editor. *PLOS Computational Biology*. 2018;14(3):e1005995. doi:10.1371/journal.pcbi.1005995
53. Jennings N, Parsons S, Pocock MJO. Human vs. machine: identification of bat species from their echolocation calls by humans and by artificial neural networks. *Canadian Journal of Zoology*. 2008;86(5):371–377. doi:10.1139/Z08-009
54. Franklin J, Miller JA. *Mapping Species Distributions*. Cambridge, United Kingdom: Cambridge University Press; 2009.
55. Miguel B Araujo, Peterson AT. Uses and misuses of bioclimatic envelope modeling. 2012;93(7):14.

56. Rebelo H, Jones G. Ground validation of presence-only modelling with rare species: a case study on barbastelles *Barbastella barbastellus* (Chiroptera: Vespertilionidae). *Journal of Applied Ecology*. 2010;47(2):410–420. doi:10.1111/j.1365-2664.2009.01765.x
57. McCune JL. Species distribution models predict rare species occurrences despite significant effects of landscape context. *Journal of Applied Ecology*. 2016;53(6):1871–1879. doi:10.1111/1365-2664.12702
58. Luizza M, Evangelista P, Jarnevich C, West A, Stewart H. Integrating subsistence practice and species distribution modeling: assessing invasive elodea’s potential impact on Native Alaskan subsistence of Chinook salmon and whitefish. *Environmental Management*. 2016;58(1):144–163. doi:10.1007/s00267-016-0692-4
59. Liang L, Clark JT, Kong N, Rieske LK, Fei S. Spatial analysis facilitates invasive species risk assessment. *Forest Ecology and Management*. 2014;315:22–29. doi:10.1016/j.foreco.2013.12.019
60. Vega GJ de la, Schilman PE. Using eco-physiological traits to understand the realized niche: the role of desiccation tolerance in Chagas disease vectors. *Oecologia; Heidelberg*. 2017;185(4):607–618. doi:http://dx.doi.org/10.1007/s00442-017-3986-1
61. Machado-Machado EA. Empirical mapping of suitability to dengue fever in Mexico using species distribution modeling. *Applied Geography*. 2012;33:82–93. doi:10.1016/j.apgeog.2011.06.011
62. Lee JR, Raymond B, Bracegirdle TJ, Chadès I, Fuller RA, Shaw JD, Terauds A. Climate change drives expansion of Antarctic ice-free habitat. *Nature*. 2017;547(7661):49–54. doi:http://dx.doi.org/10.1038/nature22996
63. Wenger SJ, Isaak DJ, Luce CH, Neville HM, Fausch KD, Dunham JB, Dauwalter DC, Young MK, Elsner MM, Rieman BE, et al. Flow regime, temperature, and biotic interactions drive differential declines of trout species under climate change. *Proceedings of the National Academy of Sciences of the United States of America*. 2011;108(34):14175–14180. doi:10.1073/pnas.1103097108
64. Dozat T. Incorporating Nesterov Momentum into Adam. *Proceedings of 4th International Conference on Learning Representations*. 2016. (Workshop Track).
65. Lembrechts JJ, Nijs I, Lenoir J. Incorporating microclimate into species distribution models. *Ecography*. 2019;42(7):1267–1279. doi:10.1111/ecog.03947
66. Jaberg C, Guisan A. Modelling the distribution of bats in relation to landscape structure in a temperate mountain environment. *Journal of Applied Ecology*. 2001;38(6):1169–1181. doi:10.1046/j.0021-8901.2001.00668.x

67. Scherrer D, Christe P, Guisan A. Modelling bat distributions and diversity in a mountain landscape using focal predictors in ensemble of small models. *Diversity and Distributions*. 2019;25(5):770–782. doi:10.1111/ddi.12893
68. Thuiller W, Georges D, Engler R, Breiner F. biomod2: Ensemble Platform for Species Distribution Modeling. 2020. <https://CRAN.R-project.org/package=biomod2>
69. Leo Breiman. Random Forests. *Machine Learning*. 2001;45(1):5–32.
70. Barbet-Massin M, Jiguet F, Albert CH, Thuiller W. Selecting pseudo-absences for species distribution models: how, where and how many?: How to use pseudo-absences in niche modelling? *Methods in Ecology and Evolution*. 2012;3(2):327–338. doi:10.1111/j.2041-210X.2011.00172.x
71. Ancillotto L, Ariano A, Nardone V, Budinski I, Rydell J, Russo D. Effects of free-ranging cattle and landscape complexity on bat foraging: Implications for bat conservation and livestock management. *Agriculture, Ecosystems & Environment*. 2017;241:54–61. doi:10.1016/j.agee.2017.03.001
72. Downs NC, Sanderson LJ. Do Bats Forage Over Cattle Dung or Over Cattle? *Acta Chiropterologica*. 2010;12(2):349–358. doi:10.3161/150811010X537936
73. Landis JR, Koch GG. The Measurement of Observer Agreement for Categorical Data. *Biometrics*. 1977;33(1):159–174. doi:10.2307/2529310
74. Gillam EH. Eavesdropping by bats on the feeding buzzes of conspecifics. *Canadian Journal of Zoology*. 2007;85(7):795–801. doi:10.1139/Z07-060
75. Fenton MB, Bell GP. Recognition of Species of Insectivorous Bats by Their Echolocation Calls. *Journal of Mammalogy*. 1981;62(2):233–243. doi:10.2307/1380701
76. Adams R, Kwiecinski G. Sonar Surveys for Bat Species Richness and Activity in the Southern Kalahari Desert, Kgalagadi Transfrontier Park, South Africa. *Diversity*. 2018;10(3):103. doi:10.3390/d10030103
77. Moss B. Water pollution by agriculture. *Philosophical Transactions of the Royal Society B: Biological Sciences*. 2008;363(1491):659–666. doi:10.1098/rstb.2007.2176
78. Jia Z, Tang S, Luo W, Hai Y. Water quality improvement through five constructed serial wetland cells and its implications on nonpoint-source pollution control. *Hydrological Sciences Journal*. 2016;61(16):2946–2956. doi:10.1080/02626667.2016.1171323
79. Hassanali A, Herren H, Khan ZR, Pickett JA, Woodcock CM. Integrated pest management: the push–pull approach for controlling insect pests and weeds of cereals, and its potential for other agricultural systems including animal husbandry. *Philosophical Transactions of the Royal Society B: Biological Sciences*. 2008;363(1491):611–621. doi:10.1098/rstb.2007.2173

APPENDICES

Appendix A

Python Code for Data Processing

WavChop.py

```
from pydub import AudioSegment
from pydub.utils import make_chunks
import os
import wave
import contextlib
input_folder = 'PATH TO BAT RECORDINGS (WAV FILES) '

def main():
    for path, dirs, files in os.walk(input_folder):
        for f in files:
            if f.endswith('.wav'):
                wav_file_path = os.path.join(path, f)
                my_audio = AudioSegment.from_file(wav_file_path, "wav")
                chunk_length_ms = 1000
                chunks = make_chunks(my_audio, chunk_length_ms)
                for i, chunk in enumerate(chunks):
                    chunk_num = "__" + str(i + 1)
                    output_file_name = os.path.basename(wav_file_path.replace('.wav', chunk_num)) +
                                        '.wav'
                    output_dir = os.path.dirname(wav_file_path)
                    chunk_name = os.path.join(output_dir, output_file_name)
                    print("exporting", chunk_name)
                    chunk.export(chunk_name, format="wav")

def remove_short():
    for path, dirs, files in os.walk(input_folder):
        for f in files:
            if f.endswith('.wav'):
                wav_file_path = os.path.join(path, f)
                with contextlib.closing(wave.open(wav_file_path, 'r')) as f:
                    frames = f.getnframes()
                    rate = f.getframerate()
                    duration = frames / float(rate)
                    print(duration)
                    if duration < 1.0:
```

```
os.remove(wav_file_path)
print('Short File deleted: ' + wav_file_path)
```

```
main()
remove_short()
```

Generate_spectrograms.py

```
import os
import numpy as np
from matplotlib import pyplot as plt
import librosa
from librosa import display
input_folder = '/PATH/TO/WAV/SEGMENTS/DIRECTORY'
```

```
def main():
    for path, dirs, files in os.walk(input_folder):
        for f in files:
            if f.endswith('.wav'):
                wav_file_path = os.path.join(path, f)
                print("Checking " + wav_file_path)
                output_file_name = os.path.basename(wav_file_path.replace('.wav', '.jpg'))
                output_dir = os.path.dirname(wav_file_path)
                output_path = os.path.join(output_dir, output_file_name)
                if not os.path.isfile(output_path):
                    print("Creating spectrogram of " + output_path)
                    plot_audio_spectrogram(wav_file_path, output_path)
```

```
def plot_audio_spectrogram(audio_path, plot_path=None, sr=256000):
    y, sr = librosa.load(audio_path, sr=sr)
    plt.figure(figsize=(100, 100), dpi=6, frameon=False)
    D = librosa.amplitude_to_db(np.abs(librosa.stft(y)), ref=np.max)
    me = np.mean(D, 1)
    spec = D - me[:, np.newaxis]
    librosa.display.specshow(spec, cmap='magma')
    plt.subplots_adjust(left=0, right=1, bottom=0, top=1)
    plt.axis('off')
    plt.savefig(plot_path)
    plt.close()

    return spec
```

```
main()
print("Scan Complete!!!")
```

Appendix B

Tensorflow Code for Retraining Google's InceptionV3 Model

retrain.py

```
import tensorflow as tf
import os
import tensorflow_hub as hub
import datetime
import numpy as np
import wandb
from wandb.keras import WandbCallback
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from sklearn.metrics import cohen_kappa_score
from sklearn.metrics import matthews_corrcoef
from sklearn.metrics import confusion_matrix
import logging

# Limit messages
logger = tf.get_logger()
logger.setLevel(logging.WARNING)
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2' # or any {'0', '1', '2'}

gpus = tf.config.experimental.list_physical_devices('GPU')
if gpus:
    try:
        tf.config.experimental.set_memory_growth(gpu, True)
        print(len(gpus), "Physical GPUs")
    except RuntimeError as e:
        print(e)

hyperparameter_defaults = dict(
    Dropout=0.2,
    BATCH_SIZE=4,
    l2=1e-4,
)

wandb.login()
wandb.init(config=hyperparameter_defaults, notes="")
config = wandb.config

print(tf.version.VERSION)
```

```

print("Hub version:", hub.__version__)
print("GPU is", "available" if tf.config.list_physical_devices('GPU') else "NOT AVAILABLE")
CurrTime = datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
print("Training starting at " + CurrTime)

module_selection = ("inception_v3", 299)
handle_base, pixels = module_selection
MODULE_HANDLE =
"https://tfhub.dev/google/imagenet/{ }/feature_vector/4".format(handle_base)
IMAGE_SIZE = (pixels, pixels)
print("Using { } with input size { }".format(MODULE_HANDLE, IMAGE_SIZE))

BATCH_SIZE = config.BATCH_SIZE
N_TRAIN = int(1e4)
STEPS_PER_EPOCH = N_TRAIN//BATCH_SIZE

data_dir = "/Your/Data/Directory/"

datagen_kwargs = dict(rescale=1./255,
                      validation_split=0.20
                      )
dataflow_kwargs = dict(target_size=IMAGE_SIZE,
                       batch_size=BATCH_SIZE,
                       interpolation="bilinear"
                       )

valid_datagen = tf.keras.preprocessing.image.ImageDataGenerator(
    **datagen_kwargs)
valid_generator = valid_datagen.flow_from_directory(
    data_dir, subset="validation", shuffle=False, **dataflow_kwargs)

train_datagen = valid_datagen
train_generator = train_datagen.flow_from_directory(
    data_dir, subset="training", shuffle=True, **dataflow_kwargs)

do_fine_tuning = True

print("Building model with", MODULE_HANDLE)

BuzzFinder = tf.keras.Sequential([
    hub.KerasLayer(MODULE_HANDLE, trainable=do_fine_tuning),
    tf.keras.layers.Dropout(rate=config.Dropout),
    tf.keras.layers.Dense(train_generator.num_classes,
                           kernel_regularizer=tf.keras.regularizers.l2(config.l2),
                           activation='softmax')])

```

```

BuzzFinder.build((None,)+IMAGE_SIZE+(3,))
BuzzFinder.summary()
BuzzFinder.compile(
    optimizer=tf.keras.optimizers.Nadam(learning_rate=0.0001,
        beta_1=0.9,
        beta_2=0.999,
        epsilon=1e-07,
        name='Nadam'),
    loss=tf.keras.losses.BinaryCrossentropy(label_smoothing=0.2),
    metrics=['accuracy'])

steps_per_epoch = train_generator.samples // train_generator.batch_size
validation_steps = valid_generator.samples // valid_generator.batch_size

# Callbacks
es_callback = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=5)
ckpt_path = '/Directory/to/save/checkpoints' + CurrTime
ckpt_callback = tf.keras.callbacks.ModelCheckpoint(filepath=ckpt_path, save_best_only=True)

def scheduler(epoch):
    if epoch < 10:
        return 0.0001
    elif epoch < 20:
        return 0.0005
    else:
        return 0.0001

lr_schedule = tf.keras.callbacks.LearningRateScheduler(scheduler)

hist = BuzzFinder.fit(
    train_generator,
    epochs=50, steps_per_epoch=steps_per_epoch,
    validation_data=valid_generator,
    validation_steps=validation_steps,
    callbacks=[WandbCallback(),
        es_callback,
        lr_schedule,
        ckpt_callback])

# Directory path to save the final model. The best checkpoint model was previously saved.
model_save_path = '/Directory/to/save/final/model_' + CurrTime
BuzzFinder.save(model_save_path)

```

```

def get_model_perf(generator):
    filenames = generator.filenames
    nb_samples = len(filenames)

    preds = BuzzFinder.predict(generator, verbose=1, steps=nb_samples)

    preds_cls_idx = preds.argmax(axis=1)
    y_true = generator.labels

    print('*****')
    print('| Confusion Matrix |')
    print('*****')
    print('    ')

    cm = confusion_matrix(y_true, preds_cls_idx)
    cm_norm = cm / cm.astype(np.float).sum(axis=1)
    print(cm_norm)

    print("TP: ", cm[1, 1])
    print("FP: ", cm[0, 1])
    print("TN: ", cm[0, 0])
    print("FN: ", cm[1, 0])
    print('*****')
    print('| STATS |')
    print('*****')
    print('    ')

    precision = precision_score(y_true, preds_cls_idx)
    recall = recall_score(y_true, preds_cls_idx)
    f_score = f1_score(y_true, preds_cls_idx)
    kappa = cohen_kappa_score(y_true, preds_cls_idx)
    mcc = matthews_corrcoef(y_true, preds_cls_idx)

    print('Precision: ', np.round(precision, decimals=4))
    print('Recall: ', np.round(recall, decimals=4))
    print('F-score: ', np.round(f_score, decimals=4))
    print('Kappa: ', np.round(kappa, decimals=4))
    print('MCC: ', np.round(mcc, decimals=4))

# Reload the best model and make predictions
print("Reloading Best Model: " + ckpt_path)
print("This may take a minute...")
BuzzFinder = tf.keras.models.load_model(ckpt_path)

```

```

print("Best model: Ready, Set... Predict!")
print('Predict on Training Set...')
train_generator = train_datagen.flow_from_directory(
    data_dir, subset="training", shuffle=False, **dataflow_kwargs)
get_model_perf(train_generator)

print('Predict on Validation Set...')
get_model_perf(valid_generator)

print('Predict on Test Set...')
pred_dir = '/PATH/TO/test_data/spectrograms'
pred_datagen = valid_datagen
pred_generator = pred_datagen.flow_from_directory(
    pred_dir, shuffle=False, **dataflow_kwargs)
get_model_perf(pred_generator)

```

predict.py

```

import tensorflow as tf
import tensorflow_hub as hub
import numpy as np
import pandas as pd
import datetime

# Setup Memory Growth fix for GTX 1660 Super
gpus = tf.config.experimental.list_physical_devices('GPU')
if gpus:
    try:
        # Currently, memory growth needs to be the same across GPUs
        for gpu in gpus:
            tf.config.experimental.set_memory_growth(gpu, True)
        print(len(gpus), "Physical GPUs")
    except RuntimeError as e:
        # Memory growth must be set before GPUs have been initialized
        print(e)

print(tf.version.VERSION)
print("Hub version:", hub.__version__)
print("GPU is", "available" if tf.config.list_physical_devices('GPU') else "NOT AVAILABLE")
CurrTime = datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
print('Predicting starting at ' + CurrTime)

```

```

IMAGE_SIZE = (299, 299)
BATCH_SIZE = 16

train_dir = "/PATH/TO/Data/train/"

datagen_kwargs = dict(rescale=1./255,
                      validation_split=.20)

dataflow_kwargs = dict(target_size=IMAGE_SIZE,
                       batch_size=BATCH_SIZE,
                       interpolation="bilinear")

valid_datagen = tf.keras.preprocessing.image.ImageDataGenerator(
    **datagen_kwargs)
valid_generator = valid_datagen.flow_from_directory(
    train_dir, subset="validation", shuffle=False, **dataflow_kwargs)

train_datagen = valid_datagen
train_generator = train_datagen.flow_from_directory(
    train_dir, subset="training", shuffle=True, **dataflow_kwargs)

# Load Model
model_name = 'YOUR SELECTED MODEL NAME'
model_path = '/PATH/TO/SAVED/MODELS/' + model_name

BuzzFinder = tf.keras.models.load_model(model_path)

# Get Predictions (Test)

site_name = '_Total' # Prediction data identifier. Can be any string of text.
pred_dir = '/PATH/TO/UNCLASSIFIED/DATA/'

pred_datagen = tf.keras.preprocessing.image.ImageDataGenerator(rescale=1./255)

pred_generator = pred_datagen.flow_from_directory(
    pred_dir,
    target_size=IMAGE_SIZE,
    color_mode='rgb',
    classes=None,
    class_mode='binary',
    batch_size=1,
    shuffle=False,
    interpolation='bilinear')

filenames = pred_generator.filenames

```

```
nb_samples = len(filenamees)

preds = BuzzFinder.predict(pred_generator, verbose=1, steps=nb_samples)

print('Match Prediction to Class Label')

preds_cls_idx = preds.argmax(axis=-1)
idx_to_cls = {v: k for k, v in train_generator.class_indices.items()}
preds_cls = np.vectorize(idx_to_cls.get)(preds_cls_idx)

print('Store Predictions in DataFrame')
predictions = pd.DataFrame({"filenamees": filenamees, "prediction": preds_cls, "Buzz Prob":
preds[:, 1]})

print('Saving Predictions')
# Save Predictions to CSV
result_save_dir = '/DIRECTORY/TO/SAVE/RESULTS/'
pred_report_name = model_name + '_predictions.csv'
predictions.to_csv(result_save_dir + pred_report_name)
```

Appendix C

Environmental Variables for Spatial Modeling

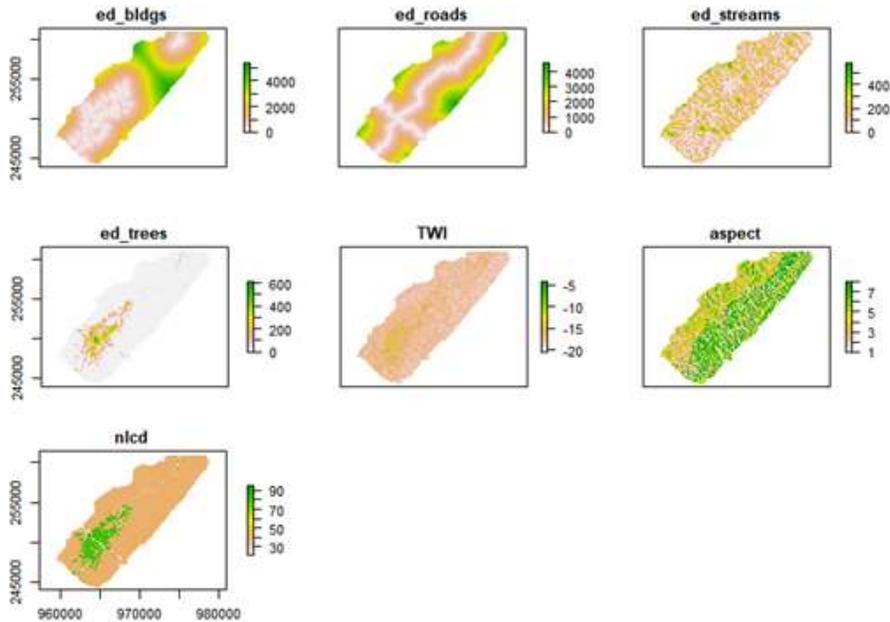


Figure 7: Environmental variables used to model species distribution and bat foraging activity

Digital Elevation Model

The 10-meter DEM was obtained from the 3DEP dataset produced by USGS and masked to the study area in Shady Valley, TN, USA.

Land Cover

The land cover data were obtained from the National Land Cover Database 2011 (NLCD 2011) produced by the Multi-Resolution Land Characteristics (MRLC) consortium. The NLCD 2011 was masked to Shady Valley, TN, and then resampled to a 100-meter resolution using the “majority” resampling technique. Using this technique, the value for each cell is determined by the land cover type that is most commonly found within the cell.

Aspect

The aspect layer was calculated based on the 3DEP DEM and then resampled using bilinear interpolation to 100-meter resolution for use in the SDM and SAM analyses.

Topographic wetness index

A topographic wetness index layer was calculated from the 3DEP DEM using the TWI one-step tool in SAGA GIS. The TWI layer was then exported to ArcGIS Pro and resampled using bilinear interpolation to 100-meter resolution for use in SDM and SAM analyses.

Euclidean distances to buildings

A building footprint layer was obtained from the Tennessee Geographic Information Council. I used the Euclidean distance tool in ArcGIS to calculate the Euclidean distance from the center of each 100-meter cell to the closest building edge using a 100-meter output cell size and the planar distance method.

Euclidean distances to roads

The Euclidean distance to the nearest major road was calculated based on the TL 2019 database produced by the United States Census Bureau. Since most bat species avoid busy roads, the polylines for Tennessee State Route 91, Tennessee State Route 133, and US Route 421 were extracted from the dataset and used for analysis. The Euclidean distance tool in ArcGIS was used to calculate the Euclidean distance from the center of each 100-meter cell to the closest major road using a 100-meter output cell size and the planar distance method.

Euclidean distances to streams

ArcGIS Pro was used to create a layer of potential streams and waterways based on the 3DEP DEM. First, the fill tool was used to fill sinks in the DEM. Then the flow direction tool

was used to calculate the D8 flow direction. Next, the flow accumulation tool was used to calculate where water would potentially accumulate on the landscape. To determine the location of ecologically important waterways, the raster calculator was used to extract all locations with greater than 2,000 cells of flow accumulation. Then, the raster-to-polyline tool was used to extract a polyline layer with the location of potential waterways. The Euclidean distance tool in ArcGIS was used to calculate the Euclidean distance from the center of each 100-meter cell to the closest waterway using a 100-meter output cell size and the planar distance method.

Euclidean distances to the forest edge

To map the forested and open regions in Shady Valley, the NLCD 2011 was reclassified in ArcGIS Pro. Deciduous forest, evergreen forest, mixed forest, and woody wetlands were reclassified as “forests,” and all other land cover types were classified as “open.” Then, I extracted the “forests” polygon to a new layer and used the Euclidean distance tool to calculate the Euclidean distance from the center of each 100-meter cell to the closest tree line using a 100-meter output cell size and the planar distance method.

Appendix D

R Code for Ensemble Species Distribution Modeling in Biomod2

```
packages <- c('biomod2', 'randomForest', 'kernlab', 'raster', 'caret', 'dismo', 'rgdal', 'maptools',
'hexbin')

if (length(setdiff(packages, rownames(installed.packages())) > 0) {
  install.packages(setdiff(packages, rownames(installed.packages())))
}

library(biomod2)
library(randomForest)
library(raster)
library(caret)
library(dismo)
library(rgdal)
library(maptools)

setwd("//Your//Working//Directory// ")
par.defaults <- par(no.readonly=TRUE)
save(par.defaults, file="R.default.par.RData")

speciesData <- read.csv("presence.csv")

ed_bldgs <- raster("ed_buildings.asc")
ed_roads <- raster("ed_roads.asc")
ed_streams <- raster("ed_streams.asc")
ed_trees <- raster("ed_forest.asc")
TWI <- raster("twi.asc")
aspect <- raster("aspect.asc")
nlcd <- raster("nlcd.asc")

myExpl <- stack(list(ed_bldgs=ed_bldgs,
                    ed_roads=ed_roads,
                    ed_streams=ed_streams,
                    ed_trees=ed_trees,
                    TWI=TWI,
                    aspect=aspect,
                    nlcd=nlcd
                ))
names(myExpl)
plot(myExpl)

myRespName <- "Bats"
```

```

myRespXY <- speciesData[,c("LON_DD","LAT_DD")]

myResp <- as.numeric(speciesData[,myRespName])
myBiomodData <- BIOMOD_FormatingData(      resp.var = myResp,
      expl.var = myExpl,
      resp.xy = myRespXY,
      resp.name = myRespName,
      PA.nb.rep = 1,
      PA.nb.absences = 10000,
      PA.strategy = 'random')

plot(myBiomodData)

myBiomodOption <- BIOMOD_ModelingOptions()

myBiomodModelOut <- BIOMOD_Modeling(
  myBiomodData,
  models = c( 'GLM' ),
  models.options = myBiomodOption,
  NbRunEval = 5,
  DataSplit = 70,
  Prevalence = 0.5,
  VarImport = 3,
  models.eval.meth = c("TSS", "ROC"),
  SaveObj = TRUE,
  rescal.all.models = TRUE,
  do.full.models = FALSE,
  modeling.id = paste(myRespName, "Bats", sep=""))

myBiomodModelEval <- get_evaluations(myBiomodModelOut)

myBiomodModelEval["TSS", "Testing.data" ,,,]
myBiomodModelEval["ROC", "Testing.data" ,,,]

get_variables_importance(myBiomodModelOut)

myGLMModels <- BIOMOD_LoadModels(myBiomodModelOut, models=c('GLM'))
myRespPlotGLM <- response.plot2(models = myGLMModels,
  Data = get_formal_data(myBiomodModelOut,'expl.var'),
  show.variables= get_formal_data(myBiomodModelOut,'expl.var.names'),
  do.bivariate = FALSE,
  fixed.var.metric = 'median',
  col = c("red", "blue", "green"),
  legend = TRUE,
  data_species = get_formal_data(myBiomodModelOut,'resp.var'))

```

```

myBiomodProj <- BIOMOD_Projection(
  modeling.output = myBiomodModelOut,
  new.env = myExpl,
  proj.name = 'current',
  selected.models = c( "Bats_PA1_RUN1_GLM",
                      "Bats_PA1_RUN2_GLM",
                      "Bats_PA1_RUN3_GLM",
                      "Bats_PA1_RUN4_GLM",
                      "Bats_PA1_RUN5_GLM"),
  binary.meth = 'TSS',
  compress = 'xz',
  build.clamping.mask = FALSE,
  output.format = '.grd')

plot(myBiomodProj, str.grep = "GLM")

myBiomodEM <- BIOMOD_EnsembleModeling( modeling.output = myBiomodModelOut,
  chosen.models = c( "Bats_PA1_RUN1_GLM",
                    "Bats_PA1_RUN2_GLM",
                    "Bats_PA1_RUN3_GLM",
                    "Bats_PA1_RUN4_GLM",
                    "Bats_PA1_RUN5_GLM"),
  em.by = 'all',
  eval.metric = c('ROC'),
  eval.metric.quality.threshold = c(0.6),
  prob.mean = TRUE,
  prob.cv = FALSE,
  prob.ci = FALSE,
  prob.ci.alpha = 0.05,
  prob.median = FALSE,
  committee.averaging = FALSE,
  prob.mean.weight = TRUE,
  prob.mean.weight.decay = 'proportional')

EMplot <- BIOMOD_EnsembleForecasting(projection.output = myBiomodProj,
  EM.output = myBiomodEM)

plot(EMplot)

Ensemble_raster <- raster("Bats/proj_current/proj_current_Bats_ensemble.grd")
writeRaster(Ensemble_raster, file="Bats/proj_current/proj_current_Bats_ensemble",
  format="ascii", overwrite=TRUE)

```

VITA

PHILIP ALLAGAS

- Education: M. S. Biology, East Tennessee State University,
Johnson City, Tennessee, 2020
B.S. Biology, East Tennessee State University,
Johnson City, Tennessee, 2018
A.S. Science, Virginia Western Community College,
Roanoke, Virginia, 2016
Northside High School, Roanoke, Virginia, 2005
- Professional Experience: Teaching Assistant for Biology I & II Labs,
East Tennessee State University,
Johnson City, Tennessee, 2018-2020
- Professional Presentation: Philip Allagas, Dr. Richard T. Carter. November 2019.
Identification of feeding buzzes using machine learning.
Tennessee Bat Working Group, Chapel Hill, Tennessee.