National University of Computer and Emerging Sciences, Lahore

# Mosquito Surveillance System

Muhammad Ahmed Javed Roll No. 21L-7692 BS(DS)
Misbah Munir Roll No. 21L-6237 BS(DS)
Ghulam Mustafa Roll No. 21L-6221 BS(DS)

Supervisor: Ms. Mamoona Akbar Khan

Final Year Project

December 5, 2025

# Anti-Plagiarism Declaration

This is to declare that the above publication was produced under the:

**Title: Mosquito Surveillance System**

is the sole contribution of the author(s), and no part hereof has been reproduced as it is the basis (cut and paste) that can be considered Plagiarism. All referenced parts have been used to argue the idea and cited properly. I/We will be responsible and liable for any consequence if a violation of this declaration is determined.

Date: 23-04-2025

Name: Muhammad Ahmed Javed.

Signature: 

Name: Misbah Munir

Signature: 

Name: Ghulam Mustafa

Signature: 

## Author’s Declaration

This states Authors’ declaration that the work presented in the report is their own, and has not been submitted/presented previously to any other institution or organization.

## Abstract

Mosquito-borne diseases cause serious public health problems that need effective surveillance methods. This project, titled 'Mosquito Surveillance System', combines Geographic Information System, remote sensing, and Artificial Intelligence to identify mosquito habitats, species categories using camera-based images, and visualization of Geographic Information System data using the dashboard. Using Sentinel-2 imagery, related numerical data (such as temperature, humidity, vegetation index, precipitation, etc), and machine learning models, the system will identify high-risk habitats. The mosquito surveillance process can be improved by developing an integrated and automatic system that uses Large Language Model (LLM). LLM will be integrated with features, allowing users to query the system and generate actionable reports. With the integration of these technologies, the system provides public health agencies with a scalable solution to effectively monitor and mitigate mosquito-borne diseases.

## Executive Summary

Our FYP is a Mosquito Surveillance System. This project will help researchers, public health officials, and environmental agencies map risks related to mosquito-borne diseases. Diseases caused by mosquitos are becoming a threat. Thus, it is important to track and study the species as well as habitats of Mosquito. The method integrates Geographic Information System, remote sensing, and Artificial Intelligence to provide automated habitat detection and recognition of species and GIS-based visualization. This system uses Sentinel-2 images and environmental parameters (temperature, humidity, vegetation index, and rain) to identify high-risk mosquito habitats. The system employs machine learning algorithms as well as LLM to enable improved data analysis, providing extensive information and responses to queries. The addition of LLM guarantees that users are able to create useful analysis and take data influenced decisions in vector control strategies. One of the important features of this project is the contextualization of the environmental and imagery data for proper risk-level assessment. Surveillance is done manually using old model and is very time-consuming. This system is different because of automating habitat detection, reducing monitoring efforts over larger areas. The dashboard is user-friendly and allows interaction. The dashboard helps visualize and filter the data based on different filters.The combination of LLM, GIS, AI, and mosquito surveillance data is a new approach.

With this method, user is allowed to ask the system which makes easy access to technical or non-technical individuals. The system is flexible. In order to stop the spread of disease, automated habitat detection and visualization help in the decision-making. The uniqueness is its AI analytical feature and LLM integration. This project incorporates an interactive environmental relations monitoring system. The combination of AI and GIS produces automated deep insight that is useful to scientist, public health official, and environmental expert across the globe. In conclusion, Our system is an evolution in the monitoring of mosquitoes. It elegantly integrates remote sensing, AI-based classification and detection, GIS visualization, and LLM-powered analysis into a self-sufficient system that can be used to observe, assess, and respond to the risks posed by diseases transmitted by mosquitoes. With the context provided by its user's designed interface, it significantly enhances the public health outcomes.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1 Introduction

Traditional methods of mosquito surveillance are not practical for large-scale monitoring since they rely on manual data collection and possess limited analysis capabilities. The proposed Mosquito Surveillance System in the project is based on AI, remote sensing, and GIS. It identifies the habitats of mosquitoes, classifies the species of mosquitoes, and represents data interactively. Risk assessment as well as autonomous habitat detection is facilitated through integration of Sentinel-2 imagery with climatic variables in the system. LLM also facilitate better interpretation of data to allow users to produce actionable reports and interact with the system via natural language questions. The purpose of this technique is to enhance mosquito accuracy and efficiency. With this approach, users are allowed to query the system using natural language, which makes easy access to technical experts and non-technical individuals.

## 1.1 Purpose of this Document

It is a manual for researcher, project developer, and stakeholder involved in the project.The system's element, strategy, and the AI technologies to enhance mosquito habitat identification is covered in the report.

## 1.2 Intended Audience

Below is a list of the document's intended audience:

- Public health organizations.
- Epidemiology, Environmental, and Entomology Researchers.
- Government and NGOs In vector-borne disease control
- Students learning AI applications in environmental monitoring

## 1.3 Definitions, Acronyms, and Abbreviations

**FYP:** Final Year Project
**SDG**: Sustainable Development Goals
**API**: Application Programming Interface
**AI**: Artificial Intelligence
**GEE**: Google Earth Engine
**GIS**: Geographic Information System

**LLM**: Large Language Model

**ROI**: Region of Interest

**OSM**: Open Street Map

**UI**: User Interface

**GUI**: Graphical User Interface

**RAM**: Random Access Memory

**SSD**: Solid State Drive

**GB**: Gigabyte

## 1.4 Conclusion

System use mosquito control by using AI, data analytics, and visualization. The objective of the project is to develop a artificial intelligence system that will improve the efficacy mosquito habitat detection. System uses remote sensing and machine learning to improve surveillance. The rfunctional architecture offered by the low-level and high-level system designs can help public health organizations in stoping diseases. The system offers a user interface that uses LLMs to support natural language queries

# Chapter 2 Project Vision

By using AI and LLM study aim is to automate habitat detection and allow interactive data visualization. Sentinel-2 satellite images and climatic data will allow the system to offer risk evaluation and actionable recommendations to assist public health organizations in their disease prevention efforts

## 2.1 Problem Domain Overview

Public health is still threatened in tropical and subtropical areas by mosquito-borne illnesses. Conventional surveillance approaches are defined by hand-collected data, which takes time, results in resource use, and is still deficient in current analysis. Unquestionably, such a massive automated intelligent mosquito monitoring system will help greatly to improve observations and reaction plans.

## 2.2 Problem Statement

Limited integration of technology, dependency on manual observations, and response times in existing mosquito surveillance systems lead to inefficiencies. Disease control measures are hindered by the lack of an automated system that can gather humongous-scale environmental data, detect high-risk mosquito breeding sites, and provide real-time information. With the use of an AI-based mosquito surveillance system combining GIS, remote sensing, and LLM for improved monitoring and decision-making, this research aims to address these challenges.

## 2.3 Goals and Objectives

The primary goals and objectives of this project are:

- Develop an AI-based system to automate the detection of mosquito habitats using numerical data and Sentinel-2 imagery and classify mosquito species using camera-based images.
- Dashboard for visualizing mosquito habitats.
- Providing user-friendly query responses.
- Adding search capabilities, actionable recommendations for mosquito control measures
- AI to automate the detection of mosquito habitats using numerical data and Sentinel-2 imagery.

## 2.4 Project Scope

Aim is to develop an AI system to automate the detection of mosquito habitats using Sentinel-2 images. System will feature a dashboard to visualize different charts and LLM for improved user-friendly query responses. Interactive search capabilities and detailed report generation with visualizations will support decision-making and mosquito control

## 2.5 Sustainable Development Goal (SDG)

Modules of our system such as mosquito habitat detection, mosquito species classification, and GIS-based dashboard provide insights and useful information for public health organizations and researchers. LLM enables user-friendly query responses. Our System contributes to recommendations, disease prevention, and overall community well-being that directly aligns with the objectives of SDG-3.


**Figure 2.1: Support Health and Well-Being for Everyone, at Every Age**

## 2.6 Constraints

System constraints are given below:

- Users must provide images manually as the system does not fetch images in real-time.
- The model's performance depends on image quality. Poor resolution can lead to inaccurate mosquito species classification and habitat detection.
- LLM responses rely on the fine-tuned dataset, and biased training data may affect the accuracy.

## 2.7 Conclusion

Our system offers mosquito habitat detection, species classification, and disease prevention. The GIS-based dashboard supports decision-making and LLM improves accessibility. Hence, this system is helpful for vector-borne disease control

# Chapter 3 Literature Review

This chapter examines the existing platforms and technologies that influenced the creation of My FYP. It reviews prior industry work in areas such as mosquito surveillance technologies, AI-powered habitat detection, and automated environmental monitoring. The exploration of these domains provided the foundation for My FYP helping shape its advanced features such as automated mosquito habitat detection, species classification, and GIS-based dashboard.

## 3.1 Definitions, Acronyms, and Abbreviations

**CNNs**: Convolutional Neural Networks

**RNNs**: Recurrent Neural Networks

**ANNs**: Artificial Neural Networks

**LSTM**: Long-Short-Term Memory

**SVM**: Support Vector Machine

**KNN**: K-Nearest Neighbors

**AUC**: Area Under the Curve

**IoT**: Internet of Things

**YOLOv8**: You Only Look Once, version 8

**VGG-16**: Visual Geometry Group-16

**YOLOv3**: You Only Look Once, version3

**GRUs**: Gated Recurring Units

**RF**: Random Forest

**GBM**: Gradient Boosting Machines

**SNR**: Signal-To-Noise Ratios

**SSL**: Self-Supervised Learning

**MaxEnt**: Maximum Entropy Model

## 3.2 Detailed Literature Review

The following sections provide each review material in detail including its summary, the critical analysis, and relation to our work.

### 3.2.1 MosquitoFusion

#### 3.2.1.1 Summary of the Research Item

In training the deep learning algorithm named ”MosquitoFusion” [1], 1204 images are used. They reached 57.1% mAP@50 performance by using the YOLOv8 system . However, the limited data set availability lowers the ability to generalize across different environmental conditions. The study emphasizes the need to increase the data set's size to increase the model accuracy. As they provide quicker response times, therefore, automated mosquito detection systems make public health monitoring more efficient. The integration potential between this model and GIS-based systems shows positive prospects. The computational resource requirements present a significant obstacle to large-scale implementation. Wide applicability can be obtained by increasing model efficiency and variety of data sets.

#### 3.2.1.2 Critical Analysis of the Research Item

For field applications, YOLOv8 works well. The limited variety of datasets keeps within bounds the model's efficiency in the different geographical locations. The computational cost of the system is another limitation a large-scale application could face. Additionally, high-quality annotated data is required for a model to work but it takes a lot of time.

#### 3.2.1.3 Relationship to the Proposed Research Work

GIS-based tracking systems will benefit spatial monitoring. Researchers and public health agencies can get help from LLM. Dataset diversity and Sentinel-2 imagery will help the system to get scalability over different geographic locations while improving model efficiency.

### 3.2.2 Aedes-AI

#### 3.2.2.1 Summary of the Research Item

Based on environmental factors, this study, named Aedes-AI [2] predicts mosquito abundance using ANNs, LSTM, and GRUs. The potential of AI in vector control tactics is demonstrated by the models' ability to accurately mimic trends in mosquito populations. Surveillance can be improved as predictive analytics provide early alerts. The study shows the tackling of ecological data and mosquito populations using AI. The method requires a lot of real-world validations to ensure correctness. Relying on extensive environmental databases scalability is constrained. AI models with real-time data sources can improve the ability to forecast.

#### 3.2.2.2 Critical Analysis of the Research Item

Contribution comes from its creation of a targeted dataset that enables mosquito detection and closes an existing void in live mosquito surveillance solutions. For field application, YOLOv8 works well and it enables rapid detection capabilities. But limited dataset lowers the model's performance in different geographical regions. Large-scale applications can face limitations because of the computational expenses of the system. Secondly, high-quality annotated data is needed for the model to perform well but the data collection process requires a lot of time. This research establishes a fundamental basis for mosquito monitoring yet requires additional improvements to reach a wider application potential

#### 3.2.2.3 Relationship to the Proposed Research Work

By showing how AI-based models can accurately forecast mosquito population patterns using meteorological and environmental data, the study directly aligns. The habitat detection and classification aspects of the proposed study can be greatly improved by modeling spatiotemporal fluctuations in mosquito abundance using LSTM and GRU networks. Additionally, by including real-time forecasts, probabilistic forecasting of Aedes abundance using neural networks might enhance public health professionals' decision-making by complementing the GIS-based dashboard. Sentinel-2 imaging and other environmental parameters, however, can improve forecasts, making them more accurate and useful across a variety of ecological regions, as these models rely on high-quality training data. Actionable insights for focused mosquito control initiatives can be obtained by combining these forecasting methods with the Mosquito Surveillance engine's LLM-powered query engine.

### 3.2.3 HumBugDB

#### 3.2.3.1 Summary of the Research Item

Dataset named HumBugDB [3] contains sounds related to mosquito wings, which is used to detect different species. The dataset includes audio recordings taken in various settings with the goal of enhancing AI-based methods for detecting mosquitoes. It allows researchers to train machine-learning models for species categorization by providing annotated audio data with mosquito flight tones. It emphasizes how crucial sound-based mosquito monitoring is as an affordable and non-intrusive substitute for conventional surveillance techniques.

#### 3.2.3.2 Critical Analysis of the Research Item

Recordings of real wing-beats is a old methods but that time it increase the training od models for correct identification, it is considered as a biggest plus point or scalable deployment with affordability for traditional surveillance technique. The issue is in the dataset they had used because there were background noise or sound, different changing in the environmental parameters, and redundancy in data frequencies, these issue really lower the accuracy of the model. In addition, there were filtering methods used that also reduce the performance of model because of noisy environemnts.

#### 3.2.3.3 Relationship to the Proposed Research Work

It is relevant to our project because it used dataset that contains some columns which are useful for our system. Using audioi data for identify the mosquito is highly correlated with one of our modules. And conjuction with satellite imagery as well. That is why it is a supporting method but it is not a core aspect for our research and study because of the challenges already discussed above about the dataset. Background noise and sampling techniques are really questionable and these techniques also lower the result of the study as well.

### 3.2.4 Population Mapping Using Dragonfly Robot

#### 3.2.4.1 Summary of the Research Item

For Population mapping and surveillance thorugh robot is conducted. Mosquitous are attracted and tracked by robot and collected real-time data of mosquito behavious and their flying patters. Different types of analysis is conducted to identify the mosquitos species. For this purpose the Dragonfly Robot[4] different sensors and algorithms

#### 3.2.4.2 Critical Analysis of the Research Item

Automatic data collection ability with lessen dependency on manual trapping techniques is one of its main advantages. AI integration improves categorization accuracy, which makes population mapping more effective. Its real-time data transfer also facilitates quick vector control decision-making. Nevertheless, there are issues with the study, namely battery restrictions that impact extended field use. The robot's flying stability and data accuracy may be impacted by environmental conditions such as wind and rain. More validation in different ecological contexts is also necessary. Due to implementation costs and expensive developments adoption at a large scale is restricted. The work advances AI-driven mosquito surveillance methods despite these drawbacks. Future developments can concentrate on increasing adaptation to various climatic situations and maximizing power efficiency.

#### 3.2.4.3 Relationship to the Proposed Research Work

Mosquito surveillance and habitat monitoring align perfectly with my FYP. It uses a dragonfly robot for real-time data collection, with AI-based classification but my project integrates satellite imagery and numerical data for large-scale habitat detection. Unlike the mobile robot, my project relies on static remote sensing and environmental data. Integrating elements from both approaches could improve surveillance efficiency.

### 3.2.5 Deep Learning-Based Mosquito Species Classification

#### 3.2.5.1 Summary of the Research Item

Utilization of deep learning for mosquito species classification using smartphone photos which improves the efficiency and accessibility of species identification. CNN [5] is used to analyze the mosquito photos, which shows high classification accuracy. The method is scalable and economical as it does not require skilled entomologists. However, appropriate data pre- treatment and picture quality are necessary for accurate categorization. The study demonstrates how mosquito surveillance in remote locations may be enhanced using mobile-based AI. All things con- sidered, it is in line with initiatives to automate species categorization and enables AI-based mosquito surveillance.

#### 3.2.5.2 Critical Analysis of the Research Item

Automating species identification with smartphone photos is a very useful and economical method. The method improves accuracy by utilizing deep learning, which makes it a useful vector control tool. The real-time processing capacity can make decisions quickly particularly in areas where mosquito-borne diseases are prevalent. . However, because model performance can be impacted by changes in illumination and picture backdrops, the method has issues with data quality. Furthermore, the model's capacity to generalize could be constrained when it comes to unknown species or morphological differences. Another challenge is computational needs since deep learning models could not function well on low-end devices.

#### 3.2.5.3 Relationship to the Proposed Research Work

Study connected heavily with our FYP, especially in terms of mosquito categorization. The classification technique based on smartphones enhances the suggested system by offering a scalable way to be used in the real world. Furthermore, real-time monitoring and decision-making may be improved by combining smartphone picture catego- rization with the GIS-based dashboard. The suggested work also has to address the study's problems, which include picture variability and computing limitations, which call for fixes like data augmentation and model improvement. Additionally, the requirement for robust dataset collection in the proposed research is reinforced by the reliance on high-quality annotated data. The study's conclusions can direct enhancements to the model's functionality and practicality.

### 3.2.6 Autonomous Mosquito Habitat Detection

#### 3.2.6.1 Summary of the Research Item

The study [6] aims to detect disease risk through the application of CNNs and satellite images for the automatic detection of mosquito breeding sites. The software attempts to spotlight potential sites by analyzing high-resolution satellite images. By automating the detection of habitats, the technique enhances systems that provide early warning of diseases transmitted by mosquitoes. It makes the monitoring of vast territories more efficient by eliminating the need for thorough ground investigations. The challenge of the research is the accessibility to trained sites with labeled images. The precision of obtained results is impacted by these factors as well. Environmental changes cause additional difficulties for the generalization of the model. Multi-spectral data integration is recommended by the study to increase detection accuracy.

#### 3.2.6.2 Critical Analysis of the Research Item

Automatic process of detecting mosquito habitats through deep learning and satellite images can increase the accuracy of disease risk detection while lessen the dependence on field surveys. This approach is much quicker and much easier. It is important to note, however, that the effectiveness relies heavily on the resolution of the picture and the quality of labeled training data. Additionally, seasonal and environmental variations can affect model performance along with the fact that the model would require regular retraining. One more thing worth mentioning is that the data gathered is not enough to offer a legitimate correlation of actual mosquito population behaviors. Plus, the identification of cloudy or densely vegetated regions is greatly restricted due to the nature of satellite imaging.

#### 3.2.6.3 Relationship to the Proposed Research Work

By utilizing CNNs and satellite photos to locate mosquito habitats, this study supports my FYP's objective of automating monitoring. Like my strategy with Sentinel-2 data and environmental elements, it places a strong emphasis on integrating data from several sources. In contrast to this study, my FYP incorporates LLMs to improve user engagement and data interpretation. The study's dependence on seasonal fluctuations and picture resolution draws attention to issues that my research has to resolve. Its extensive mapping methodology enhances my GIS-based risk visualization dashboard. My habitat categorization might be further enhanced by investigating thermal and multispectral data. The accuracy of my system can be improved by using real-world validation techniques. The necessity of model optimization in my research is emphasized by computational difficulties. Important information from this study will help me enhance my AI and LLM-based mosquito surveillance system.

### 3.2.7 MosquIoT

#### 3.2.7.1 Summary of the Research Item

MosquIoT [7] integrates smart traps that are equipped with sensors to collect environmental and mosquito-related data, then this data is transferred via wireless networks to a cloud platform for real-time analysis. Machine learning models categorize mosquitoes and predict infestation levels depending on environmental factors like humidity, carbon dioxide, and temperature. This reduces dependency on manual inspection. This approach enhances early warning capabilities and improves the efficiency of mosquito surveillance programs. Cloud-based storage ensures remote access and scalability.

#### 3.2.7.2 Critical Analysis of the Research Item

Integration of IoT sensors and machine learning models is the main strength. It is adaptable to different geographic locations because of the scalable architecture. The significant drawback is the dependency on IoT infrastructure which is expensive to deploy. Sensor malfunction and data quality have an impact on model accuracy. Another weakness is the absence of comparison with conventional mosquito surveillance techniques. Machine learning models require new data for optimal performance.

#### 3.2.7.3 Relationship to the Proposed Research Work

It leverages machine learning and IoT but our approach is the integration of Sentinel-2 imagery and GIS-based analysis. Our research focuses on remote sensing data for habitat detection, using camera-based images for species classification but the study depends on real-time monitoring using IoT sensors. Additionally, our system uses LLM to improve decision-making, interactivity with users, and adaptive. Unlike MosquIoT, environmental parameters such as temperature, humidity, vegetation index, and rainfall strengthen predictive accuracy. Our research also aims to develop an interactive GIS-based dashboard for report generation, query handling, and visualization.

### 3.2.8 An IoT-Based Smart Mosquito Trap System

#### 3.2.8.1 Summary of the Research Item

IoT and deep learning models [8] are used by mosquito trap systems to track mosquito numbers. The technology is made to automatically classify mosquito species by taking and analyzing photos of the insects in real time. To interpret photos taken from the trap and precisely identify mosquito species, the authors used a neural network model. The system's potential for integration with public health monitoring systems is covered in the article. The benefits of employing IoT for ongoing data collecting are also highlighted by the authors.

#### 3.2.8.2 Critical Analysis of the Research Item

The positive aspect involves combination of IoT and deep learning, which automates mosquito species identification with high accuracy. Real-time image processing reduces the need for manual identification. The system is also designed for scalability, allowing multiple traps to be deployed and connected to a centralized network. However, continuous internet connectivity shows challenge in areas with poor internet. Power consumption is another issue, as continuous data transmission and image processing require heavy energy. The study does not explore how the system handles environmental variations like dust, humidity, or lighting conditions. The cost of deployment and maintenance is also not discussed in detail. Future research should focus on optimizing power efficiency and adaptability.

#### 3.2.8.3 Relationship to the Proposed Research Work

Advantages of automated monitoring, which supports the FYP's goal of reducing manual intervention. However, FYP aims to extend beyond image processing by incorporating LLMs for enhanced query-based analysis. The study provides useful insights into improving the accuracy of mosquito classification models. The integration of IoT-based traps could complement the FYP's mosquito surveillance framework. Future enhancements in this research, such as optimizing energy efficiency, may benefit the FYP's implementation. The findings validate the effectiveness of AI in mosquito monitoring

### 3.2.9 FlightTrackAI

#### 3.2.9.1 Summary of the Research Item

FlightTrackAI [9] aims to provide insights into mosquito movement patterns. Researchers used high-speed video recordings of mosquitoes in a controlled environment to train the AI model. Tool successfully tracks mosquito flight paths with high precision. The study shows the importance of flight behavior in understanding mosquito-borne disease spread. Experimental results show that the tool outperforms traditional tracking methods in accuracy and efficiency.

#### 3.2.9.2 Critical Analysis of the Research Item

Unique application to examine mosquito flying behavior offers important insights into disease prevention. High-speed video recordings make it a trustworthy tool for behavioral research. Precision of the model is greater than traditional tracking techniques. However, dependency on a laboratory setting does not represent real-world mosquito behavior because, in real-world cases, we have issues like wind and fluctuating lighting. Additionally, it is limited to only Aedes aegypti

#### 3.2.9.3 Relationship to the Proposed Research Work

The high-speed image processing technique discussed in the paper can inform the FYP's mosquito classification methodology. However, the research is limited to controlled environments, whereas the FYP aims to apply AI in real-world settings using GIS-based analysis. The study highlights the potential of AI in mosquito monitoring, supporting the project's overall approach. The research's findings can be leveraged to refine the classification model used in the FYP. The proposed tool can be integrated into future improvements for mosquito tracking. The study validates the use of AI-driven approaches for mosquito analysis.

### 3.2.10 Study of Vector Ecology using Machine Learning

#### 3.2.10.1 Summary of the Research Item

Researchers examined environmental factors to forecast the mosquito population. Climate influenced the mosquito species distribution significantly. Machine learning techniques like random forests and neural networks [10] are used to find the vital ecological drivers. To conduct precise vector surveillance, the results show the importance of incorporating climate data. Need for environmental monitoring and remote sensing in mosquito ecology is also emphasized in the study. Data-driven methods for mosquito habitats can improve forecasting tools.

#### 3.2.10.2 Critical Analysis of the Research Item

Using machine learning techniques makes it relevant for vector surveillance. Combining climatic and landscape factors is a positive point, which increases the predictive accuracy of the model. Additionally, leveraging remote sensing and environmental monitoring makes it scalable for large-area mosquito habitat detection. However, a weak point is variability of climatic conditions which needs frequent model retraining for different regions because different regions have different climate patterns. Another limitation is the availability of high-resolution environmental data in all regions.

#### 3.2.10.3 Relationship to the Proposed Research Work

Climate influence and landscape factors on mosquito habitat detection aligns closely with our FYP because of using machine learning techniques. Our FYP focuses on integrating numerical data such as temperature, humidity, precipitation, and vegetation index with Sentinel-2 imagery for mosquito habitat detection. The use of machine learning models for species classification and habitat detection is directly relevant to our approach but our project extends beyond this work by incorporating LLM for enhanced data interpretation.

### 3.2.11 Mosquito Surveillance and Control Using Drones

#### 3.2.11.1 Summary of the Research Item

Application of drone technology offers an economical way of carrying out surveillance [11]. Locations that are not accessible by ground teams, with the help of drones ground teams can reach those locations. In addition, drones have high-resolution cameras and a Global Positioning System (GPS) that aids in precise mapping and monitoring of breeding sites for mosquitoes. Regulatory guidelines are taken into consideration when implementing drones in such programs. Acceptance by the community and formulation of Standard Operating Procedures (SOPs) are necessary for safety and efficiency.

#### 3.2.11.2 Critical Analysis of the Research Item

Use of drones offers major advantages, includes coverage of wide area, real-time monitoring, quick data collection, this makes efficient breeding sites detection. Targeted pesticides spraying minimizes the environmental impact. Drones reduces human exposure to disease-prone areas which improves the worker safety as well. However, high initial costs, regulatory restrictions, and limited battery life are the challenges. Weather conditions, dense vegetation can affect accuracy, and maintenance and technical issues have operational risks.

#### 3.2.11.3 Relationship to the Proposed Research Work

Drones have limitations but our FYP uses Sentinel-2 satellite imagery, numerical data and LLM for automated and intelligent system. Remote sensing data along GIS is used to detect mosquito habitats with operational disruptions. Additionally, GIS-based dashboard with LLM combination enables intelligent query responses to users. In fact, drones are useful for localized monitoring, our system combines multiple data sources to offer cost-effective and holistic approach to mosquito surveillance.

### 3.2.12 Robust Mosquito Species Identification

#### 3.2.12.1 Summary of the Research Item

Deep learning techniques are used to classify mosquito species based on body and wings images. The study explored CNNs [12] to analyze the image datasets which overcomes the challenges such as variations in lighting, pose, and resolution. The researchers compare different deep learning architecture to find the most effective model for species classification. This approach can enhance vector surveillance programs which leads to better-targeted control measures and disease prevention strategies.

#### 3.2.12.2 Critical Analysis of the Research Item

Deep learning techniques for mosquito species identification offers high accuracy, automation which increases vector surveillance and disease control. Strength lies in the image dataset which includes the diverse body and wing images that makes the model more robust against variations in lighting and pose. The comparison of different CNN architectures ensures optimal model selection which improves the classification performance. However, the research face limitation in dataset generalizability because training of specific data might reduce the effectiveness of unseen mosquito species. High computation costs make real-time deployment a big challenge in low resource areas. Additionally, dependency on deep learning black-box models also limits the interpretability for entomologists.

#### 3.2.12.3 Relationship to the Proposed Research Work

Our research extends this by integrating Sentinel-2 imagery, numerical data and GIS based visualization for more comprehensive and effective mosquito surveillance. Additionally, our project comprises of LLM for automated query responses which makes it more interactive and user-friendly. Use of CNN based classification can be adapted. However, our project improves upon this by combining multiple data sources, making it more scalable and suitable for large-scale mosquito surveillance and prevention.

### 3.2.13 Species and Gender Identification in Mosquitoes Vectors

#### 3.2.13.1 Summary of the Research Item

Use of deep learning techniques to classify species and distinguish between male and female mosquitoes is crucial for disease control. The study explored CNNs [13] to analyze mosquito images, addressing limitations such as visual similarities between species, variations in image quality, and dataset imbalances. Researchers identifies optimal architecture that improve classification accuracy after testing different deep learning models such as Visual Geometry Group-16 (VGG-16), Residual Network-

50 (ResNet-50), Inception Network Version 3 (Inception-v3), and Mobile Network Version 2 (MobileNetV2).

#### 3.2.13.2 Critical Analysis of the Research Item

Models shows good accuracy if we compare it with old manual methods. Positive point is testing different models that shows high performance. The use of data augmentation and transfer learning enhances model generalization that makes system more adaptable to different datasets. However, there are some limitations such as dataset size and image quality which can lead to overfitting and reduce performance on unseen data. Dependency on high quality images under some controlled conditions restricts its applicability in real world cases. Additionally, computation complexity of deeper models may affect the deployment in low-resource setting.

#### 3.2.13.3 Relationship to the Proposed Research Work

Study explores CNNs for species and gender classification but our research extends this by combining satellite imagery, numerical environmental data using remote sensing, and GIS based visualization for an effective mosquito habitat detection and species categorization. Additionally, our work includes the LLM to provide automated query-based results that are more user-friendly. Our research overcomes the limitation of the study by combining multi-source data and developing a scalable system for practical scenarios.

### 3.2.14 MosquitoSong+

#### 3.2.14.1 Summary of the Research Item

MosquitoSong+ [14] identifies the species based on audio audio-based deep learning approach. To ensure real-world applicability, the study focuses on model performance against background noise. Various deep learning architectures, such as CNNs, RNNs, LSTMs, Transformers, and a hybrid model are tested. In result, the hybrid model and Transformers perform best achieving high classification accuracy even in challenging conditions. MosquitoSong+ outperforms previous models, proving noise-robust solution.

#### 3.2.14.2 Critical Analysis of the Research Item

The ability to operate effectively in noisy environments is the key strength, which increases its real-world applicability for field surveillance. Use of deep learning models ensures high classification accuracy and ability to capture both spatial and temporal features of mosquito wing beat patterns. However, there are notable weaknesses such as dataset limitations, computational complexity of Transformer for real-time and resource-constrained deployments. Moreover, factors like environmental noise and differences in recording conditions can affect the model performance in diverse field settings.

#### 3.2.14.3 Relationship to the Proposed Research Work

Our work includes mosquito classification using deep learning on camera-based images. Another key difference is that our project uses LLM for interactive query-based analysis that makes it more user-friendly. Our project overcomes the study's real-world limitation by providing more holistic, data-driven, and GIS powered solution making it more practice for vector control programs.

### 3.2.15 Automatic Mosquito Sensing

#### 3.2.15.1 Summary of the Research Item

System applies image-based mosquito detection through CNNs [15] which allows accurate identification of species. Additionally, IoT devices and smart sensors are used to monitor environmental conditions that influence mosquito activity. System can automatically trigger control mechanisms such as insecticide sprays that reduces mosquito populations. The study highlights the effectiveness of deep learning models in mosquito control and surveillance in mitigating the vector-borne diseases.

#### 3.2.15.2 Critical Analysis of the Research Item

Key strength is combination of deep learning models such as Fully Connected Network (FCN) and neural network-based regression for real-time mosquito species classification, improving accuracy over traditional manual identification. Additionally, the system's automation reduces human intervention in mosquito control efforts. Usage of IoT and sensors makes it applicable in real-world cases. However, its weakness is its dependency on high-quality image data, which may not always be available in real-world scenarios with varying lighting and environmental conditions. It also lacks in comparative analysis with traditional mosquito surveillance methods.

#### 3.2.15.3 Relationship to the Proposed Research Work

Their work focuses on mosquito sensing through image-based classification using FCN and neural network regression, our approach extends beyond by adding classification of species, breeding sites identification, GIS-based dashboard with integration of LLM.

### 3.2.16 GIS-Based Platform Prediction of Mosquitoes

#### 3.2.16.1 Summary of the Research Item

Platform [16] utilizes machine learning algorithms and spatial analysis to identify high-risk areas for mosquito-borne diseases in near real-time. By using remote sensing data, weather conditions, and land cover information, system increases its accuracy for breeding sites prediction. Study emphasizes the importance of GIS and data-driven models for effective vector control and prevention specifically in low-resource settings. Platform aims to provide timely and actionable insights for public health officials and researchers. Historical data is used to validate the model's reliability. The paper concludes that combining GIS with predictive analytics can help in proactive disease outbreak prevention.

#### 3.2.16.2 Critical Analysis of the Research Item

Combining dynamic and static environmental data increases the prediction accuracy which is considered as the study's positive point. Utilizing remote sensing data and machine learning algorithms for near real-time mosquito distribution modeling makes it a valuable tool for disease prevention. Additionally, scalability allows adaptation to different regions. However, there are some weaknesses such as dependency on quality of real-time data and data availability across locations. Incomplete or old datasets can affect the accuracy of the model's prediction. Secondly, multiple datasets cause computational complexity.

#### 3.2.16.3 Relationship to the Proposed Research Work

Our proposed research work and this study align closely because both focus on using geospatial data for the surveillance of mosquitoes. The existing study emphasized near real-time prediction using machine learning and GIS integration but our research extends this by incorporating Sentinel-2 imagery, annotated masks, and environmental parameters to enhance mosquito breeding site detection. Additionally, our approach integrated an LLM-based chatbot for user-friendly query responses and decision support.

### 3.2.17 Enhancing Mosquito Classification

#### 3.2.17.1 Summary of the Research Item

Study employs Simple Contrastive Learning Representation (SimCLR) [17] which is a self-supervised learning model that extracts feature representations from mosquito images without needing the labeled data. By using contrastive learning model learns to distinguish between mosquito species more effectively, leads to enhanced classification performance with fine-tuned with small amount of labeled data.

The research shows that self-supervised learning can increase the generalization and robustness of deep learning models for species classification which can consider as promising approach for disease control.

#### 3.2.17.2 Critical Analysis of the Research Item

Positive point is the ability to learn effective feature representations from unlabeled mosquito images that reduces the dependency on large annotated datasets. Additionally, the approach increases classification performance which makes useful for real-world applications in vector surveillance. However, weakness is computational complexity of contrastive learning which requires substantial resources for training. Self-Supervised Learning (SSL) performance depends on quality and diversity of pretraining dataset.

#### 3.2.17.3 Relationship to the Proposed Research Work

SimCLR is a self-supervised learning framework that reduces the need for labeled data. Integrating such methods could enhance our classification model by leveraging unlabeled mosquito images for feature learning, improving accuracy and robustness. Additionally, both studies contribute to mosquito surveillance efforts, aiding vector control strategies through automated species identification, which is a critical component of our GIS-based Mosquito Surveillance System.

### 3.2.18 Mosquito Habitat Mapping Using GIS

#### 3.2.18.1 Summary of the Research Item

GIS and remote sensing, specifically Landsat imagery and Digital Elevation Models (DEM) [18] used to identify and map mosquito breeding habitats. Environmental factors such as water accumulation, vegetation cover, and land use changes impact of coalbed methane development on mosquito populations and spread of West Nile Virus. Spatial analysis techniques include hydrological modeling and land cover classification are applied to increase habitat detection accuracy.

#### 3.2.18.2 Critical Analysis of the Research Item

Positive point is the use of Landsat imagery and DEM to analyze environmental factors that influence mosquito proliferation that provides insights into the impact of coalbed methane development on mosquito populations. However, a key limitation is the dependency on traditional remote sensing and GIS methods without incorporating advanced machine learning or deep learning methods.

#### 3.2.18.3 Relationship to the Proposed Research Work

Our approach advances the approach by integrating Sentinel-2 imagery, annotated masks, and numerical environmental data for improved accuracy. Additionally, our research incorporated deep learning-based image segmentation techniques which increase the habitat detection precision beyond the traditional GIS methods. Moreover, our work extends beyond by combining LLM-powered chatbot and GIS-based dashboard to provide interactive and actionable insights for mosquito control and prevention.

### 3.2.19 Mosquito Species Identification Using Convolutional Neural Networks

#### 3.2.19.1 Summary of the research item

Multitiered ensemble model combines with multiple CNN [19] increases species identification. Model uses transfer learning and data augmentation methods to improve generalization across different mosquito datasets. The researchers trained and evaluated model on publicly available datasets includes MosquitoNet that shows effectiveness in distinguishing species with high accuracy. The approach is significant for vector control programs because it enables precise species identification which is important for disease prevention.

#### 3.2.19.2 Critical Analysis of the Research Item

Key strength is the model's ability to generalize unseen species that improves classification accuracy beyond traditional approaches. Use of ensemble learning increases robustness and decreases misclassification errors. However, limitation is reliance on a specific dataset, such as MosquitoNet, which may not comprehensively represent global mosquito species diversity. Additionally, computational complexity and overfitting due to limited training samples for species are challenges.

#### 3.2.19.3 Relationship to the Proposed Research Work

Use of a multitiered ensemble model aligns with our goal of improving classification accuracy by utilizing advanced deep learning architectures. Additionally, the dataset MosquitoNet highlights the importance of curated, high-quality datasets, which is a critical aspect of our work. While the study focuses on novel species detection, our research integrates LLMs for enhanced interactivity and decision-making support. Furthermore, our GIS-based approach expands on their work by incorporating camera-based classification, providing a more comprehensive mosquito surveillance system.

### 3.2.20 Mosquito Classification Using Wingbeat Audio

#### 3.2.20.1 Summary of the Research Item

Two efficient deep learning models, such as CNN and LSTM [20], are used to process audio spectrograms which enables accurate identification while maintaining computational efficiency. The proposed models are lightweight, which makes them suitable for deployment in real-time, especially in resource-constrained environments. The Study shows promising results that highlights the potential of audio-based classification as non-invasive and scalable solution for mosquito monitoring.

#### 3.2.20.2 Critical Analysis of the Research Item

The use of lightweight deep learning models makes it suitable for real-time applications, even in resource-constrained environments. Additionally, the study offers an alternative to image-based classification methods. However, there are limitations such as potential noise interference in real-world conditions that may affect classification accuracy. The model's generalizability to different mosquito species in different environments is also a challenge, as variations in audio recordings due to environmental factors could impact performance. Furthermore, dependency on high-quality audio data may limit its applicability.

#### 3.2.20.3 Relationship to the Proposed Research Work

Light weight models are used for species based on wing-beat. But our research includes image-based identification using different models. Both methods are used to improve the surveillance but this our system have additional dataset to provide insights. Furthermore, while the wing-beat audio method provides a unique, sensor-based classification approach, our research complements it by utilizing camera-based classification, making it more adaptable to real-world applications such as GIS-based dashboards and large-scale mosquito monitoring systems.

## 3.3 Literature Review Summary Table

We have studied existing research papers that are relevant to the domain of mosquito surveillance. In Table 3.1, we have provided the name of author, method used, results, and limitations.

## 3.4 Conclusion

Various methodologies and techniques are examined for mosquito surveillance across different studies. CNNs, LSTM, different versions of YOLO, and deep learning-based classifiers are common models that

**Table 3.1: Overview of Related Research**

| Author | Method | Results | Limitations |
|---|---|---|---|
| Sayeedi et al. [1] | YOLO used to classify mosquitoes | 73.4 % precision,50.5% recall, and 57.1mAP@50 | Dataset biasness |
| Kinney et al. [2] | Feed-Forward, LSTM, and GRU are used | GRU outperformed, achieved 0.15 RMSE | Generalizability due to regional biases in the dataset. |
| Kiskin et al. [3] | 24000 audio recordings are use, annotated, and used for model. | 90 % accuracy with precision and recall exceeding 85 % | Lack of multi-modal data integration. |
| Semwal et al. [4] | Dragonfly Robot with sensors | 92 % accuracy and 88% of precision. | Hardware constraints |
| Jhaveri et al. [5] | VGG16 and InceptionV3 are used to classify.. | 96 % accuracy, 94% precision, and 95% of recall. | Poor lighting conditions. |
| Elango et al. [6] | U-Net and VGG-16 applied for detection | U-Net achieved 82 % IoU | Cloud-covered and low-resolution |
| Aira et al. [7] | IoT-enabled traps with sensors and machine learning models such as Random Forest, SVM, and KNN | Random Forest classifier achieved an accuracy of 94%, the precision of 92%, recall of 93%, and F1-score of 92%, | Extreme weather conditions and sensor calibration is required for consistent accuracy. |
| Liu et al. [8] | IoT-enabled smart traps with cameras and sensors to capture mosquito images, processed by YOLOv3 for real-time species identification | Achieved 92% accuracy, 91.3% precision, recall 90.7%, and F1-score 91.0% | Low light environments and computational demand can limit deployment in low-resource areas. |
| Javed el al.[9] | CNN-based tool, is used to track the flight behavior. | CNN shows 88% accuracy, 85% precision, 84% recall | Dense swarm conditions and hardware constraints. |

are used for mosquito detection and species identification. Some studies explored IoT-based systems and real time image processing to increase the surveillance effectively. But CNNs massively used for mosquito classification. For predictive analysis, Support Vector Machine and Boosting are used. Few studies focus on autonomous habitat detection using satellite imagery. Furthermore, drone and robot-based surveillance is used for real-time monitoring purposes. Our research aims to develop a scalable surveillance system that combines deep learning, remote sensing, and GIS for advanced and automated habitat detection and species classification. Additionally, LLM will be used in the dashboard to facilitate interactive analysis and intelligent query responses, increasing the decision-making for stakeholders.

| Author | Method | Results | Limitations |
|---|---|---|---|
| Ferraguti et al. [10] | RF and GBM to analyze landscape and climatic factors influencing mosquito abundance and species composition in southern Spain. | RF achieved $R^2$ 0.81, RMSE 0.19, and MAE of 0.14, while GBM achieved an $R^2$ 0.78.RF performed better than GBM | Region-specific data affects generalizability and temporal variations |
| Carrasco-Escobar et al. [11] | Drones equipped with cameras and sensors, combined with YOLOv3 and Faster R-CNN models. | Faster R-CNN achieved 88% accuracy, 85% precision, 83% recall, and 84% F1-score. Faster R-CNN performs better than YOLOv3 in this study. | Weather conditions, battery life of drones, and image resolution issues in dense vegetation areas. |
| Nolte et al. [12] | VGG16, ResNet-50, and InceptionV3 were applied to classify mosquito species from diverse body and wing images. | ResNet-50 performed best with 98% accuracy, 97% precision, 98% recall, and 97.5% F1-score. | Poor-quality and incomplete images and in cases of rare species with fewer training samples. |
| Kittichai et al. [13] | VGG-16 and ResNet-50 were used for mosquito species and gender identification from images. | ResNet-50 outperformed VGG-16, achieving 96.8% accuracy, 97.2% precision, 96.5% recall. | Noisy images and difficulty in identifying morphologically species. |
| Supratak et al. [14] | MosquitoSong+ used for mosquito classification from wingbeat sounds. | Model achieved 92.3% accuracy, 91.7% precision, 90.8% recall, and 91.2% F1-score. | Limited availability of diverse species sound datasets. |
| Kim et al. [15] | Used YOLOv3 and Faster R-CNN models for real-time mosquito detection and classification. | YOLOv3 achieved 92.5% accuracy, 91% precision, 90% recall, and 90.5% F1-score. | Small mosquito sizes, and high computational demands for real-time processing. |
| Kuchavaram. M, Huang. Y [16] | RF, SVM, and Gradient Boosting used to predict mosquito distribution using environmental and spatial data. | RF outperformed other models, achieving 92% accuracy, 90% precision, 91% recall, and F1-score of 90.5%. | Model adaptability to rapid environmental changes, and computational cost. |
| Charoenpanyaku et al. [17] | SSL combined with ResNet-50 and EfficientNet models to enhance mosquito species classification. | ResNet-50 with SSL achieved 94% accuracy, 92% precision, 93% recall, and 92.5% F1-score. | Highly imbalanced datasets and extremely poor-quality images. |
| Zou et al.[18] | MaxEnt and Logistic Regression used to map mosquito larval habitats and assess environmental factors affecting. | MaxEnt achieved higher performance with an AUC of 0.91.MaxEnt is the better-performing model | Static mapping without accounting for temporal variations in habitat conditions. |
| Goodwin et al. [19] | VGG16, ResNet50, InceptionV3 used for species identification | Ensemble model achieved 96.1% accuracy, 95.5% precision, 94.8% recall. | Imbalanced datasets, extremely rare species. |

| Author | Method | Results | Limitations |
|---|---|---|---|
| Yin et al. [20] | Lightweight CNN model used to classify species based on wing-beats audio spectrograms | The model achieved 92% accuracy, 91% precision, 90% recall, and 90.5% F1-score, offering competitive performance with minimal computational cost. | Affected by background noise, overlapping frequencies, and limited variety of species in the dataset. |

# Chapter 4 Software Requirement Specifications

Software requirements for our project is given in this chapter which includes main modules. We have listed functional and non-functional requirements for this project. All necessary use cases, description, and risk analysis is also given in this chapter.

## 4.1 List of Features

Bullets are for the features that will be added in our system:

- Sentinel-2 satellite imagery for detection.
- Dashboard for visualizing analyzing historical data.
- LLM model that will handle user questions via Chat box.

## 4.2 Functional Requirements

All necessary requirements for this project is given below:

#### 4.2.0.1 Functional Requirements for System

- System use Sentinel-2 satellite imagery to detect potential mosquito.
- Dashboard for visualizing mosquito hotspots and analyzing spatial data.
- LLM will answer user questions for the better understanding in textual format.

## 4.3 Quality Attributes

All standards and quality attributes are written below for this system:

### 4.3.1 Reliability

Satellite imagery will be processed by the detection module. This will ensure the data quality, low losses and corruption of images and user interaction with chatboc and dashboard for analyzing the historical data to gain insights.

### 4.3.2 Flexibility

Configuration with new modules such as new dataset or new layers such as adding new plots should be easy understandable by the system by adding new features in future use cases.

### 4.3.3 Security

Perfect security protocols should be in the system to stop any external attacks from data breaching or leaking. Safeguarding techniques must be implemented against attacker access and should ensure data quality

### 4.3.4 Maintainability

System should allow updates about its components. For example future improvements can be updated LLM and detection model which can be manage easily without high overhead or plenty.

### 4.3.5 Usability

UI should be friendly to users to provide insights from dashboard and chatbox. And ensuring that users have easy access to satellite imagery, and understanding the insights without the need of technical knowledge.

### 4.3.6 Performance

System should show high performance in terms of computation, processing large MBs of satellite imagery and during visualization with help of dashboard and LLM response as well.

### 4.3.7 Efficiency

Our system should use hardware effecient, mapping, and LLM interactions, even under high usage scenarios.

## 4.4 Non-Functional Requirements

These requirements are used for making optimal performance, usability, and scalability of the Mosquito Surveillance System.

### 4.4.1 Performance

- System show output within 5 seconds.
- GIS dashboard will render visualizations (heatmaps, markers, layers) in under 3 seconds.

### 4.4.2 Usability

- System will allow users to upload satellite images for analysis in no more than 3 steps.

- System will allow health officials to query real-time data via the integrated LLM in under 2 steps.
- The dashboard will be responsive across mobile, tablet, and desktop devices.
- UI will be intuitive and user-friendly for individuals with basic technical knowledge.

### 4.4.3 Scalability

- System will handle concurrent access by at least 100 users, including data scientists, epidemiologists, and municipal authorities.
- Architecture will support modular integration of future datasets.

## 4.5 Assumptions

- Satellite provided as input is of acceptable resolution and quality.
- Availability of internet connectivity for real-time LLM query and dashboard operation.
- LLM APIs will remain accessible and within usage limits.
- No critical legal or ethical restrictions prevent the use of satellite or camera data in the proposed regions.

## 4.6 Use Cases

This section contains key use cases that represent core functionalities of the Mosquito Surveillance System. These scenarios include mosquito habitat detection using satellite imagery, interactive exploration of mosquito-prone zones via the GIS dashboard, and natural language querying through the integrated LLM.

### 4.6.1 Image Upload

### 4.6.2 Field Inputs

### 4.6.3 Chatbox Input

| | | | |
|---|---|---|---|
| **Name** | Image Format Handling | | |
| **Actors** | Public Health Official | | |
| **Summary** | A user attempts to upload an image in a supported format for mosquito habitat detection. The system validates. | | |
| **Pre-Conditions** | The user must have access to the image upload interface. The user must not be uploading a PNG file. FastAPI and model pipeline must be running. | | |
| **Post-Conditions** | The image is rejected, and no data is passed to the model. An error message is displayed, and the user remains on the upload page. | | |
| **Special Requirements** | System must validate file format before processing. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | User navigates to the image upload page. | 2 | System displays the image upload form with file input |
| 3 | User selects an image with an valid format | 4 | The system checks file extension. |
| 5 | User clicks the upload button. | 6 | System proceeds to habitat detection. |
| **Alternative Flow** | | | |
| 3 | User uploads invalid image. | 4-A | The system responds with an error message: Incorrect image or field entered. |

| | | | |
|---|---|---|---|
| **Name** | Field Inputs | | |
| **Actors** | Public Health Official,Field Researcher | | |
| **Summary** | A user attempts to input time and date. The system uses this data to call GEE-API. | | |
| **Pre-Conditions** | The data entry form is accessible. The values must be within expected numeric ranges. | | |
| **Post-Conditions** | Field data is validated and passed to the habitat detection model. | | |
| **Special Requirements** | Input validation must be performed in real-time. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | User navigates to the data input section. | 2 | System displays the input form. |
| 3 | User enters valid time and date. | 4 | The system validates each field. |
| 5 | User submits the form. | 6 | System stores the data and confirms successful entry. |
| **Alternative Flow** | | | |
| 3 | User inputs invalid field. | 4-A | The system responds with an error message: Incorrect field entered. |

### 4.6.4 Filters of World Map

## 4.7 Hardware and Software Requirements

Hardware and software requirements that will be required to develop and deploy the project are listed:

| | | | |
|---|---|---|---|
| **Name** | Chatbox Input | | |
| **Actors** | Public Health Official,Field Researcher | | |
| **Summary** | The user inputs a natural language query through the chatbot interface to retrieve insights or system data. The chatbot processes the input and returns a response. | | |
| **Pre-Conditions** | The chatbot must be integrated with the system backend. The LLM API must be active and authenticated. | | |
| **Post-Conditions** | A valid response is displayed to the user based on the processed input. | | |
| **Special Requirements** | Natural Language Processing support via LLM | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | User types a query | 2 | System sends the query to the LLM, retrieves the processed answer, and displays the response |
| **Alternative Flow** | | | |
| 3 | User submits vague query field. | 4-A | System responds with clarification message. |

| | | | |
|---|---|---|---|
| **Name** | Filters of World Map | | |
| **Actors** | Public Health Official | | |
| **Summary** | The user selects the filters and the map shows output based on the filters adjusted. | | |
| **Pre-Conditions** | Internet must be connected. | | |
| **Post-Conditions** | A valid response is displayed to the user based on the processed filters. | | |
| **Special Requirements** | Reliable internet connectivity is required. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | User selects filters | 2 | System retrieves the data process it and displays the response |
| **Alternative Flow** | | | |
| 3 | User selects vague values | 4-A | System responds with clarification message. |

### 4.7.1 Hardware Requirements

The hardware requirements are listed below to ensure the efficient operation of all components of the Mosquito Surveillance System.

- Intel Core i5 (8th Generation) or Advanced Micro Devices (AMD) Ryzen 5 is required.
- 8 GB of RAM is required.
- 256 GB SSD is required.
- Integrated graphics are sufficient for front-end testing and low-scale image handling.

### 4.7.2 Software Requirements

- Visual Studio Code and Jupyter Notebook
- OpenCV, NumPy, Pillow for image processing
- Scikit-learn, TensorFlow or PyTorch for machine learning and deep learning
- API for LLM-based interaction
- FastAPI for backend development
- MySQL for storing structured data and MongoDB for unstructured data

## 4.8 Graphical User Interface

GUI of the Mosquito Surveillance System is designed to be intuitive and user-friendly.

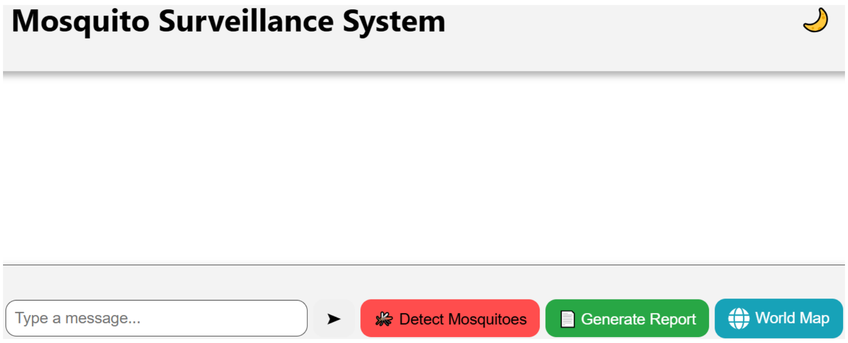


**Figure 4.1: Homepage of the System Contains Three Buttons**

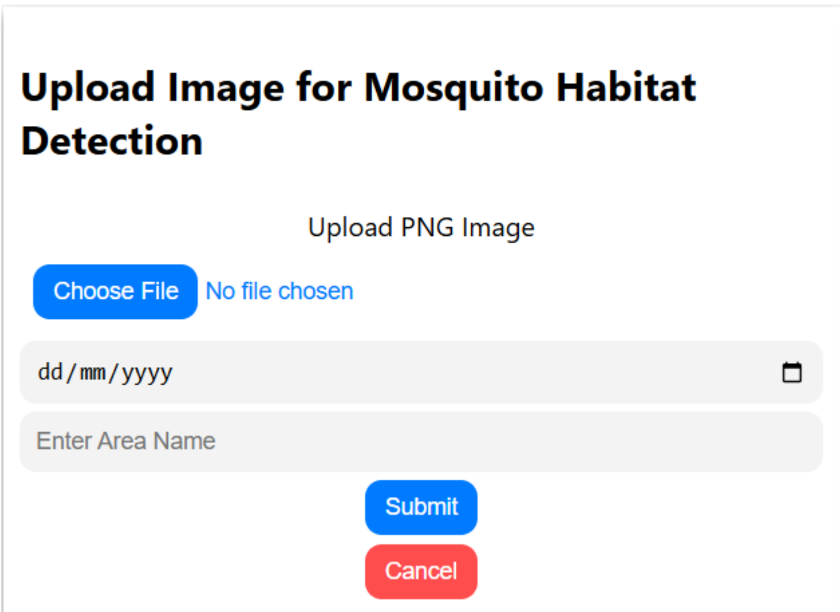


**Figure 4.2: Fields Required for Detecting the Breeding Sites**

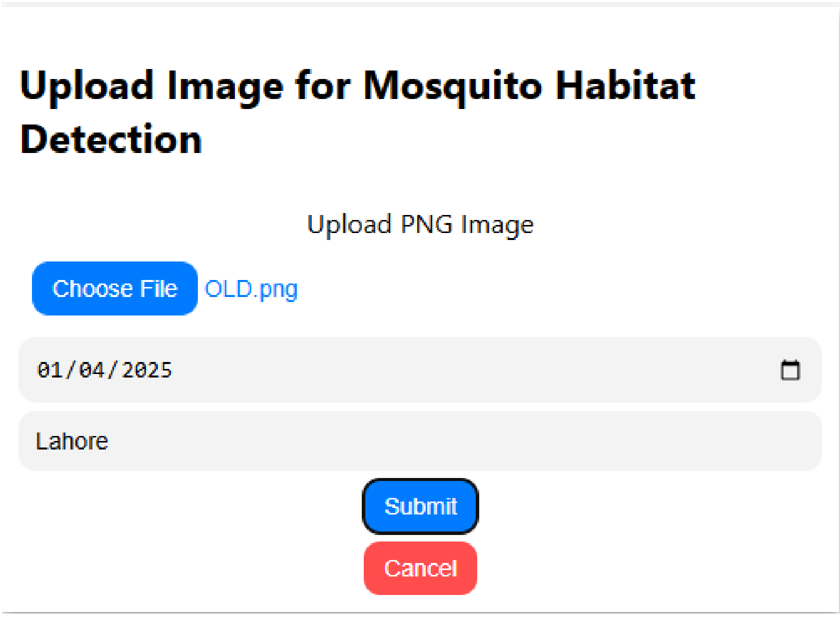


**Figure 4.3: Inputs Are Given to the Fields by the User**

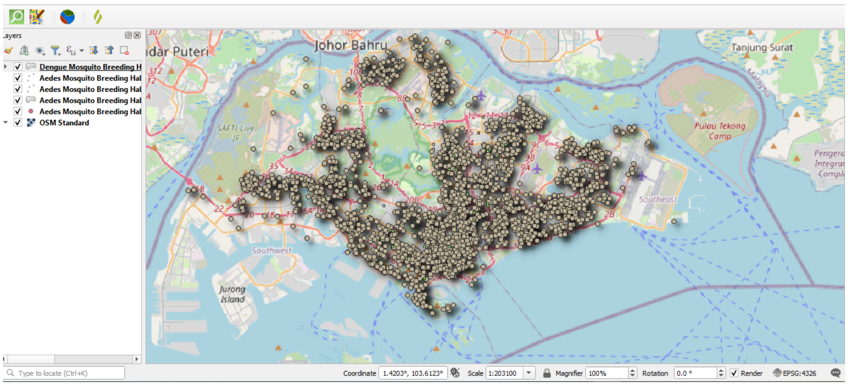


**Figure 4.4: Spatial Distribution of Dengue and Aedes Habitats in Singapore**

## 4.9 Database Design

The database design for our system is developed to provide an efficient framework. To ensure functionality and scalability, the Entity-Relationship (ER) Diagram and the Data Dictionary are implemented. ER Diagram visually represents the system's entities and their relationships. We have used MySQL as promming language and MySQL Workbench as an IDE to use MySQL. The main reason of using these two technologies was that MySQL is easy to implement different data transformation and cleanig the dataset and we have MySQL Workbench to import 4000+ rows automatically instead of manually entering every single row what was not possible in a such a short period of time. As a result, this method leads to project success.

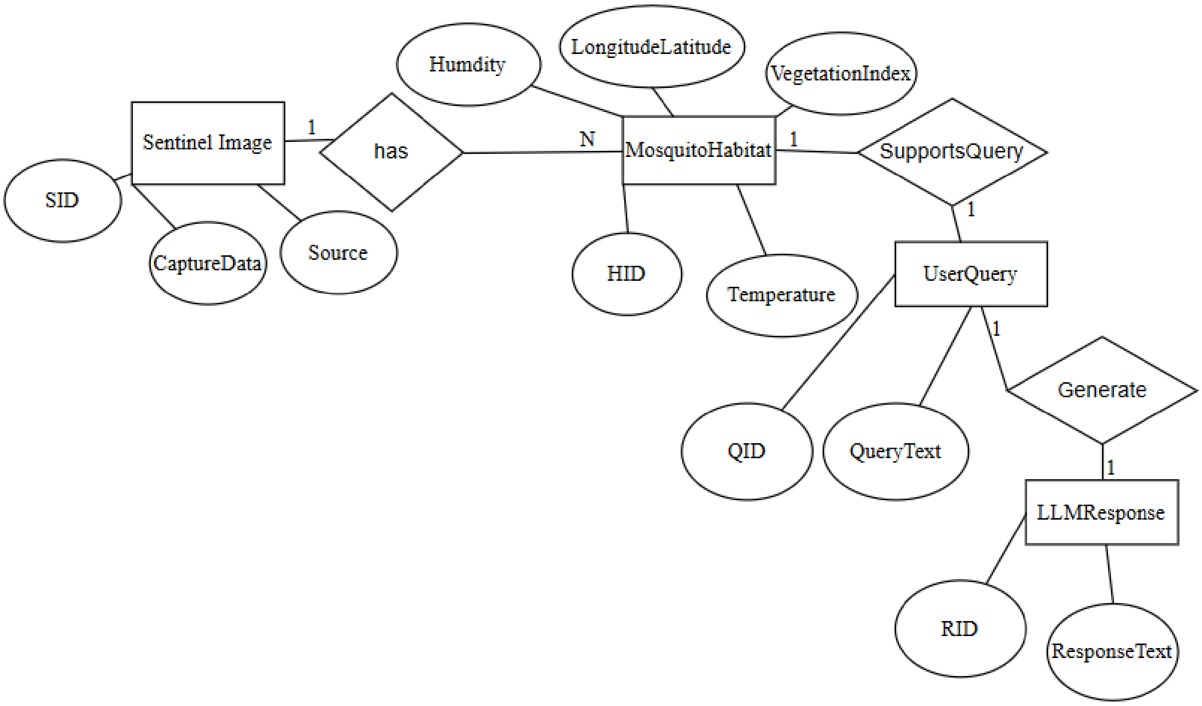


**Figure 4.5: The Relationship Between Entities Is Shown in This Diagram**

### 4.9.1 ER Diagram

### 4.9.2 Data Dictionary

**Table 4.1: Table contains detailed description of entities used in ER diagram**

| Entity | Attribute | Data Type | Nullable | Description |
|---|---|---|---|---|
| Sentinel Image | SID | int | No | Primary Key |
| | CaptureData | string | No | Data captured from the image |
| | Source | string | No | Source of the image |
| Mosquito Habitat | HID | int | No | Primary Key |
| | Humidity | float | No | Humidity level |
| | Temperature | float | No | Temperature level |
| | LongitudeLatitude | string | No | Coordinates |
| | VegetationIndex | float | No | Index value |
| User Query | QID | int | No | Primary Key |
| | QueryText | string | No | Text of the user's query |
| LLM Response | RID | int | No | Primary Key |
| | ResponseText | string | No | Text of the response |

## 4.10 Risk Analysis

There were 2 risks we had faced during project development. One is time contrasints and second is performance challenges. Project contains different stages such as data collection to model deployments and developing front-end and backend. That is why time management was very important. Because delay in any stage can slower the dinal delivery. Second point was about performance so our system contains moder AI technologies to detect habitats. These have a technical challenge in real-worl application such as image quality, different bands availability that can effect the models performance.

## 4.11 Conclusion

Chapter contains both the functional and non-functional requirements, presents the graphical user interface design, describes the database structure of the system, and evaluates potential risks associated with time and performance constraints.

# Chapter 5 Proposed Approach and Methodology

The chapter presents elaborate information about Mosquito Surveillance System development. The methods reveal algorithm functionality along with showing how chosen technological solutions implement these procedures. A full system achieves better real-time mosquito observation and danger assessment methods when deep learning combines with geographic mapping and chatbot technology. Base system features will be explained by combining descriptions about mosquito identification methods with detection procedures and GIS mapping capabilities and chatbot functionality. The subsequent parts detail complete implementation details regarding all system components.

## 5.1 Mosquito Detection

It concerns with implementation detail of our mosquito habitat detection module which will be used to detect the breeding sites.

### 5.1.1 Data Collection

GEE is decided to utilize for collection Sentinel-2 imagery. The reason to select GEE is that it provides high-resolution spectral bands which helps to detect vegetation, water bodies, and land cover changes. Additionally, numerical environmental data of the respective image will be retrieved using GEE-API

### 5.1.2 Image Annotation

QGIS is decided to use for annotating the images. It includes labeling the Sentinel-2 imagery with ground truth masks that highlight key features such as roads, buildings, ponds, water bodies (shallow or deep water), trees. Additionally, QGIS will help us to maintain the attribute table of these classes. This phase is critical because it will improve the segmentation performance and precise detection of breeding sites.

### 5.1.3 Model Training

Combination of Sentinel-2 imagery, corresponding environmental data, and an annotated mask will be given to the model to train. The model will learn associate spectral and environmental patterns with labeled breeding sites. U-Net, DeepLabV3+, or transformer-based architectures are the deep learning-based segmentation models which are commonly used for this task. The selection of best model will depend on the model evaluation metrics like Intersection over Union (IoU), accuracy, and Dice coefficient.

## 5.2 Mosquito Classification

It concerns with implementation detail of our mosquito species classification module which will be used to detect the breeding sites.

### 5.2.1 Data Collection

Kaggle contains publicly available datasets and we have decided to use it. The dataset contains labeled images which makes it best for training deep learning model. The reason of choosing it is that it contains diverse collection of high-quality images that helps improve model generalization.

### 5.2.2 Data Pre-processing

Image resizing, normalizing pixel values, and removing noise are the steps for data pro-processing to enhance feature extraction. Additionally, images may undergo grayscale conversion or color enhancement to improve the classification accuracy.

### 5.2.3 Model Training

Deep learning models such as ResNet, EffecientNet-B3, and VGG-16 are commonly used architectures for feature extraction. We have decided to use EfficientNet-B3 because it gives better accuracy, optimized in terms of speed, and have low computational cost.

## 5.3 GIS-based Dashboard

For user interaction enhancement, we have decided to develop a platform to visual mosquito habitats, species classification, and other relevant environmental factors. There will be a user-friendly interface, allowing seamless exploration of geospatial patterns. The features include interactive map, search functionalities, report generation and downloading, and LLM –powered query responses.

## 5.4 LLM-Based Chatbot

Structured and unstructured data will be collected and formatted for a chatbot. Pre-trained LLM will be chosen and fine-tuned on domain-specific data to increase accuracy in responding to mosquito-related queries. Then we will have API integration, where real-time communication between chatbot, GIS-based dashboard, and dataset through standardized APIs will be ensured. After that, we will have a testing and optimization phase that ensures the response accuracy.

## 5.5 Conclusion

Detailed methodology for our system system is shown. It displays all parts such as mosquito habitat detection using Sentinel-2 imagery and an interactive dashboard for visualization, and an LLM chatbot for user interaction. This is a method focuses to ensure accurate detection, classification, and effective communication, which ultimately increases risk assessment.

# Chapter 6 High-Level and Low-Level Design

## 6.1 System Overview

Mosquito surveillance system functions through AI operation by integrating distance measurements with deep learning models, GIS, and LLM to perform automatic mosquito habitat detection along with species identification. The system allows researchers and public health officials to use Sentinel-2 and environmental data and deep learning model for efficient mosquito population tracking. Additionally, our system will use camera-based images to predict the mosquito species, along with GIS-based visualization and report generation.

### 6.1.1 Mosquito Habitat Detection Module

Stagnant water bodies and dense vegetation together with humid regions make up the primary sites where mosquitoes lay their eggs and reproduce. The module assesses high-risk mosquito habitats through analysis of Sentinel-2 satellite images supported by environmental data and remote sensing. Public health agencies can allocate their mosquito control resources which results in less disease transmission from mosquitoes

### 6.1.2 Mosquito Classification Module

The module serves to detect various mosquito types through deep learning image classification techniques. The system receives camera-based mosquito images which then classify specimens. It is important to correctly identify different mosquito species for scientists to understand how diseases move through populations.

### 6.1.3 GIS Data Visualization Module

This module provides users with GIS map to examine mosquito spreads across different areas through an interactive system interface. Users can track the positions of mosquitoes while viewing affected risk areas and environment-related mosquito population factors through the dashboard. Users exercise data analysis through filtering tools that allow them to narrow down search criteria based on different parameters, geographical locations, and species information with environmental factor settings. Lowering decision time happens when the dashboard system displays both mosquito population data visualization and predicted disease outbreak areas.

### 6.1.4 LLM-Based Chatbot Module

The module provides users a verified mosquito data through queries to the system by using the LLM. It functions as an accessibility tool. Through extensive training, the chatbot system extracts information from complete datasets combining entomological research with public health guidelines and vector control methods. This allows users to access knowledge about mosquito prevention steps along with information about disease threats and habitat characteristics and control strategies via LLM.

## 6.2 Design Considerations

Addressing design considerations needed in design phase of our system.

### 6.2.1 Assumptions and Dependencies

Several assumptions and dependencies are made for our system:

#### 6.2.1.1 Software

For database management, MySQL will be used. Google Earth Engine - Application Programming Interface (GEE-API) will be utilized to fetch remote sensing data that contains environmental data. React will be used to develop an interactive web-based dashboard.

#### 6.2.1.2 End-User Characteristics

Public health officials, researchers, and urban planners who need insights related to mosquito habitats or breeding sites and mosquito species should have a consistent internet connection and a modern web browser. Users with minimal technical knowledge can interact with the LLM-powered query system, while advanced users can explore detailed GIS data and visual analytics. A consistent internet connection will be needed.

#### 6.2.1.3 Possible Changes in Functionality

Future improvements include real-time data integration, enhanced deep learning models for habitat detection, mobile support for field data collection, and multi-language support to ensure broader accessibility. The system could also integrate predictive analytics for early warning and proactive mosquito control strategies.

### 6.2.2 General Constraints

Constraints are written below that impact on the design of our system:

#### 6.2.2.1 Hardware and Software Constraints

Graphics Processing Unit (GPU) is required to process Sentinel-2 imageries and camera-based images with deep learning models. Secondly, specialized infrastructure is required for all the features which creates high operational costs may hinder the wide scale deployment in resource-limited regions.

#### 6.2.2.2 Performance Optimization

Managing large geospatial datasets efficiently still needs optimized processing. GIS visualization should be optimized for smooth map rendering without excessive computational overhead.

#### 6.2.2.3 Network Communication

Consistent internet connection is needed to fetch the environmental data with the help of remote sensing. To lower the connectivity issues, local storage mechanisms should be used for GIS data.

#### 6.2.2.4 Memory Limitations

Running deep learning models on standard or legacy devices may lead to memory constraints because they need optimized storage and processing techniques to handle large datasets effectively.

### 6.2.3 Goals and Guidelines

- Utilizing numerical data along with Sentinel-2 imagery for automated mosquito habitat detection and camera-based images for mosquito species classification.
- Development of a GIS-based dashboard for visualizing mosquito habitats and classification results.
- Combining LLM to increase user interactivity and support decision-making.
- For detailed insights and data-driven decision making, report generation functionality is given.
- Providing interactive features such as search capabilities and actionable mosquito control recommendations.

### 6.2.4 Development Methods

Our system operationalizes Agile development methods to maintain. It is understood that many modifications might be required based on the feedback of our advisor. We will divide our work into different sprints and all sprints will have deadlines.

## 6.3 System Architecture

The architecture comprises both front and backend development, ensuring a responsive and efficient web page. SQL Server will be used as a database that will offer data storage.

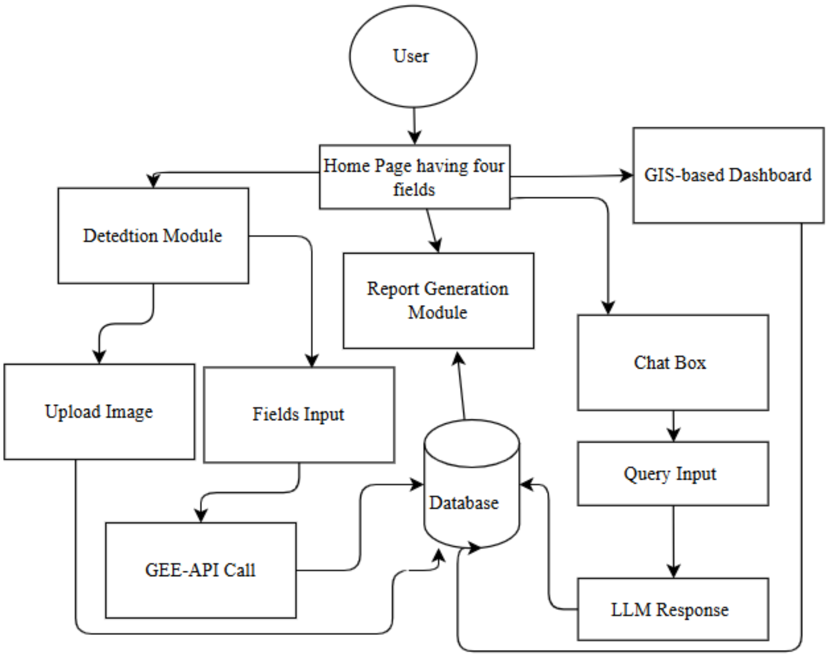


**Figure 6.1: High-Level System Architecture Diagram for Mosquito Surveillance System**

### 6.3.1 Subsystem Architecture

The specific functions of individual subsystems work together with system counterparts to build an entire mosquito surveillance system. This system contains four main subsystems which make up its architecture.

#### 6.3.1.1 Mosquito Classification Subsystem

A deep learning model-based system operates within this subsystem which identifies mosquitoes from images. The system operates through a sequence that includes image preprocessing followed by CNN-based model classification of uploaded mosquito pictures trained on various species data sets. The mapping system displays the results as classified species together with their confidence scores.

#### 6.3.1.2 Mosquito Habitat Detection Subsystem

The subsystem analyzes Sentinel-2 imagery, annotated masks made with QGIS, and environmental data such as temperature, humidity, vegetation index, and precipitation to detect potential mosquito habitats or breeding sites. The subsystem identifies the potential regions with the integration of spatial and numerical data.

##### 6.3.1.3 GIS-Based Visualization Subsystem

The GIS dashboard functions as a core element of the system since it enables users to work with mosquito surveillance data through visual interfaces. This system presents historic mosquito population data which gets combined with environmental variables like temperature and rainfall data alongside humidity levels. Users can exploit the dashboard to perform region queries while also viewing time-based mosquito risk patterns.

##### 6.3.1.4 LLM Chatbot Subsystem

Users can engage with the system through natural language queries because the LLM chatbot serves as the interface. The chatbot receives conditioning through research papers and public health guidelines and vector control strategies related to mosquitoes. The system uses the database to obtain important data which it transforms into a simple-to-understand display. The chatbot engages users by supplying reply statements derived from AI-generated analysis for queries

## 6.4 Architectural Strategies

Casual surveillance through Mosquito Surveillance System uses architectural designs that support speed and growth as well as data processing. The following section presents essential plans of action.

### 6.4.1 Tools and Software

System employs sophisticated software and tools to automate mosquito evaluation together with habitat identification along with data presentation capabilities as well as user interface functions. Deep learning-based mosquito classification through TensorFlow together with Keras frameworks supports the Python implementation of backend procedures. The system depends on Google Earth Engine (GEE) and QGIS for data collection and pre-processing. The system uses APIs from FastAPI for building interfaces between inter-module communication systems. The interface combines a GIS dashboard interface with an LLM-powered chatbot system that functions as a data query interface.

### 6.4.2 Client-Server Model

The LLM chatbot together with the GIS dashboard operates as a client interface to let users access outputs along with visualizing sites identified as habitat risks. The backend server carries out data reception then controls mosquito species identifications together with maintaining spatial risk database storage.

## 6.5 Class Diagram

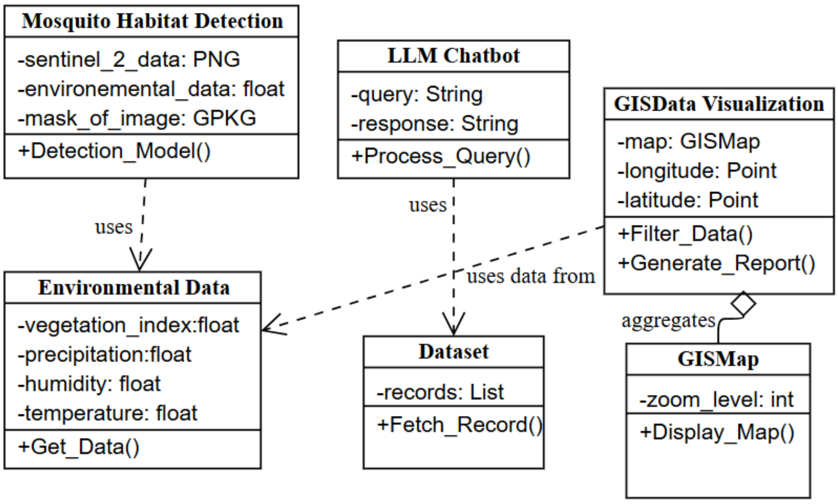


**Figure 6.2: Structure of the System is Shown in this Class Diagram**

## 6.6 Policies and Tactics

Mosquito Surveillance System needs various implementation design policies and tactics that assist its interface development stages to fulfill its essential system requirements for success and continued operation.

### 6.6.1 Tools and Development Environment

Productivity and teamwork functions of the Mosquito Surveillance System become possible through development tool deployments in defined environments. Visual Studio Code operates as the central Integrated Development Environment (IDE) since it presents vital features that integrate Git version control and a developer-friendly interface for front-end and back-end programming needs. Developers work efficiently on projects through the Git version control system because code review management and project administration and CI/CD automation occur within GitHub. Docker containers provide the system with consistent deployment environments.

### 6.6.2 Coding Guidelines

The project maintains strong code quality throughout its sections due to enforced coding guidelines.

Python developers use Python Enhancement Proposal 8 (PEP8) standards for code styling because they create clearer programming code and standardize code presentation. The system uses modular approaches in programming to achieve both easy maintenance and added system flexibility. Each module in the code must include documentation strings because Google Python Style Guide demands this standard for functional descriptions. A controlled branching system implemented in version control systems helps developer teams collaborate better because it minimizes work variability between team members

### 6.6.3 Extensibility and Modularity

The system follows a microservices architecture to split the development and deployment of its main aspects: mosquito classification plus habitat detection and chatbot services and GIS components. There are all functionalities exposed through RESTful APIs within this system design that improves integration

### 6.6.4 Testing the Software

Detailed testing serves as the base for developing reliability and operational stability for the Mosquito Surveillance System platform. The testing method for specific functionalities applies to verify intended operations. System modules need to perform their API actions correctly per the integration test results. During testing the system evaluates the end-to-end experience by assessing the mechanisms that unite GIS visualizations with chatbot capabilities.

### 6.6.5 User Interface Design

Users can operate the Mosquito Surveillance System seamlessly because its interface maintains simplicity together with usability features. The front-end interface of the Mosquito Surveillance System implements React.js to produce dynamic user interface components for its interactive user experience. The system design contains features that allow it to display proper functionality on desktops having advanced updated browser such as Chrome. The GIS visualization module of the system allows users to retrieve geospatial information for making informed decisions. Users can execute queries by using the friendly interface of the chatbot system.

## 6.7 Conclusion

The high-level and low-level design of our system is presented in this chapter, which integrates AI, deep learning, GIS, and LLM technologies for automated mosquito habitat detection and species classification. The system includes four key modules: Mosquito Habitat Detection, Mosquito Classification, GIS-based Visualization, and LLM-powered Chatbot. These modules work together to support researchers and public health officials. Design considerations such as assumptions, constraints, goals, and guidelines are discussed to ensure system usability and performance. This chapter also covers architectural strategies, domain models, coding guidelines, and modular development policies.

# Chapter 7 Implementation and Test Cases

Our system module prototype descriptions are mentioned in this section.

## 7.1 Implementation

Detailed implementation of our system's modules are outlined below.

### 7.1.1 Mosquito Habitat Detection

In this module, Sentinel-2 imagery, numerical environmental data such as temperature, humidity, vegetation index, precipitation, and annotated masks will be integrated to detect potential mosquito breeding sites. All spatial data is sourced from GEE. Pre-processing steps include image correction and feature extraction. For image annotation, QGIS is used. Python as a programming language, and GEE-API to fetch environmental parameters. Rasterio and Geospatial Data Abstraction Library (GDAL) will be used for atmospheric correction. The model is developed using a Scikit-learn Python library.

### 7.1.2 GIS-Based Dashboard

GIS-based dashboard provides user-friendly interaction with the system that helps visualize mosquito habitat results. The dashboard is built using React.js for the frontend and FastAPI for the backend. It fetched data from a MySQL database for sturcutred data MongoDB for unstructured data. Moreover, Leaflet.js is used to handle spatial visualizations. Users are allowed to filter data by location and time, with results displayed through interactive map and charts. Additionally, report generation button is also added to generate a report in Portable Document Format (PDF).

### 7.1.3 LLM Integration

To enhance user interaction with the system LLM is integrated with the modules. A pre-trained model available on Hugging Face will be fine-tuned for the project's specific needs. The model will fetch relevant information for MySQL Database, process user queries, and generate informative responses which will be displayed on the dashboard.

## 7.2 Test case Design and Description

Test cases that are conducted are given below. The purpose of this section is to check the possible errors that our system can face. We tested both normal and edge cases to ensure our system reliability. These test cases assist to verify the friendly interaction of user with our system.

**Table 7.1: Invalid Image Input Test Case**

| | | | |
|---|---|---|---|
| **Invalid Image Input Test Case** | | | |
| **UC-01** | | | |
| **Test Case ID:** | *1* | **QA Test Engineer:** | *Ahmed Javed* |
| **Test case Version:** | *1* | **Reviewed By:** | *Ghulam Mustafa* |
| **Test Date:** | *1 November 2025* | **Use Case Reference(s):** | *4.6.1: Image Upload* |
| **Revision History:** | *None* | | |
| **Objective:** | *To ensure that system correctly handles invalid image formats.* | | |
| **Product/Ver/ Module:** | *Mosquito Habitat Detection Module.* | | |
| **Environment:** | *Module runs on a desktop* | | |
| **Assumptions:** | *Supported format is Portable Network Graphics (PNG).* | | |
| **Pre-Requisite:** | *Image upload functionality and FastAPI backend must be running.* | | |
| **Step No.** | **Execution description** | **Procedure result** | |
| 1 | *User clicks the iutton.* | *System brings the user to new page.* | |
| 2 | *User upload the image.* | *System brings the images to model.* | |
| **Comments:** Requirement meets | | | |
| **Passed** | | | |

**Table 7.2: Missing Required Field Test Case**

| | | | |
|---|---|---|---|
| **Missing Required Field Test Case** | | | |
| **UC-02** | | | |
| **Test Case ID:** | *2* | **QA Test Engineer:** | *Misbah Munir* |
| **Test case Version:** | *1* | **Reviewed By:** | *Ghulam Mustafa* |
| **Test Date:** | *3 November 2025* | **Use Case Reference(s):** | *4.6.2: Field Inputs* |
| **Revision History:** | *None* | | |
| **Objective:** | *To ensure that system correctly displays error.* | | |
| **Product/Ver/ Module:** | *Mosquito Habitat Detection Module.* | | |
| **Environment:** | *Module runs on a desktop* | | |
| **Assumptions:** | *Required fields like image, date, and location are mandatory.* | | |
| **Pre-Requisite:** | *Reliable and consistent internet connectivity is needed for GEE-API.* | | |
| **Step No.** | **Execution description** | **Procedure result** | |
| 1 | *User clicks the button.* | *System brings the user to new page.* | |
| 2 | *User enters the fields.* | *System saves it and uses for GEE-API call.* | |
| **Comments:** Requirement meets | | | |
| **Passed** | | | |

**Table 7.3: Chatbot Input Functionality Test Case**

| **Chatbot Input Functionality Test Case** | | | |
|---|---|---|---|
| **UC-03** | | | |
| **Test Case ID:** | *3* | **QA Test Engineer:** | *Ghulam Mustafa* |
| **Test case Version:** | *1* | **Reviewed By:** | *Misbah Munir* |
| **Test Date:** | *5 November 2025* | **Use Case Reference(s):** | *4.6.3: Chaxbot Input* |
| **Revision History:** | *None* | | |
| **Objective:** | *To ensure that system correctly handles user queries.* | | |
| **Product/Ver/ Module:** | *LLM Chatbot Module.* | | |
| **Environment:** | *Module runs on a desktop* | | |
| **Assumptions:** | *None* | | |
| **Pre-Requisite:** | *Internet connectivity for API access is operational.* | | |
| **Step No.** | **Execution description** | **Procedure result** | |
| 1 | *User enters a query.* | *System brings the user to new page.* | |
| 2 | *User enters the fields.* | *System saves it and uses for GEE-API call.* | |
| **Comments:** Requirement meets | | | |
| **Passed** | | | |

**Table 7.4: Chatbot No Input Edge Test Case**

| **Chatbot No Input Edge Test Case** | | | |
|---|---|---|---|
| **UC-04** | | | |
| **Test Case ID:** | *4* | **QA Test Engineer:** | *Ghulam Mustafa* |
| **Test case Version:** | *1* | **Reviewed By:** | *Ahmed Javed* |
| **Test Date:** | *7 November 2025* | **Use Case Reference(s):** | *4.6.4: Chatbox Input* |
| **Revision History:** | *None* | | |
| **Objective:** | *Ensure pressing send without typing anything does not crash system.* | | |
| **Product/Ver/ Module:** | *LLM Chatbot Module.* | | |
| **Environment:** | *Module runs on a desktop* | | |
| **Assumptions:** | *None* | | |
| **Pre-Requisite:** | *Chat box is empty.* | | |
| **Step No.** | **Execution description** | **Procedure result** | |
| 1 | *User leaves empty box.* | *Input field remains inactive* | |
| 2 | *User clicks the send icon* | *No message sent.* | |
| **Comments:** Requirement meets | | | |
| **Passed** | | | |

**Table 7.5: World Map Button Test Case**

| World Map Button Functionality Test Case | | | |
|---|---|---|---|
| **None** | | | |
| **Test Case ID:** | *5* | **QA Test Engineer:** | *Ahmed Javed* |
| **Test case Version:** | *1* | **Reviewed By:** | *Misbah Munir* |
| **Test Date:** | *9 November 2025* | **Use Case Reference(s):** | *None* |
| **Revision History:** | *None* | | |
| **Objective:** | *Ensure that button correctly opens the interactive map feature.* | | |
| **Product/Ver/ Module:** | *GIS-based Dashboard Module.* | | |
| **Environment:** | *Web-based dashboard on desktop.* | | |
| **Assumptions:** | *Internet connectivity is available for map rendering.* | | |
| **Pre-Requisite:** | *Map API must be integrated and functional.* | | |
| **Step No.** | **Execution description** | **Procedure result** | |
| 1 | *User clicks the button.* | *A fullscreen map is displayed successfully* | |
| **Comments:** Requirement meets | | | |
| **Passed** | | | |

**Table 7.6: World Map Filter Buttons Test Case**

| World Map Filter Button Test Case | | | |
|---|---|---|---|
| **None** | | | |
| **Test Case ID:** | *6* | **QA Test Engineer:** | *Misbah Munir* |
| **Test case Version:** | *1* | **Reviewed By:** | *Ahmed Javed* |
| **Test Date:** | *11 November 2025* | **Use Case Reference(s):** | *None* |
| **Revision History:** | *None* | | |
| **Objective:** | *Ensure that each filter dropdown correctly displays its options,* | | |
| **Product/Ver/ Module:** | *Report Generation Module.* | | |
| **Environment:** | *Web-based dashboard on desktop.* | | |
| **Assumptions:** | *Internet connectivity is available for map rendering.* | | |
| **Pre-Requisite:** | *Map API must be integrated and functional.* | | |
| **Step No.** | **Execution description** | **Procedure result** | |
| 1 | *User applies filter.* | *Map changes based on filters.* | |
| **Comments:** Requirement meets | | | |
| **Passed** | | | |

**Table 7.7: Report Generation Without Data Test Case**

| **Report Generation Without Data Test Case** | | | |
|---|---|---|---|
| **None** | | | |
| **Test Case ID:** | *7* | **QA Test Engineer:** | *Ahmed Javed* |
| **Test case Version:** | *1* | **Reviewed By:** | *Ghulam Mustafa* |
| **Test Date:** | *13 November 2025* | **Use Case Reference(s):** | *None* |
| **Revision History:** | *None* | | |
| **Objective:** | *Ensure that clicking on does not produce broken report.* | | |
| **Product/Ver/ Module:** | *Report Generation Module.* | | |
| **Environment:** | *Web page is running on web browser.* | | |
| **Assumptions:** | *None* | | |
| **Pre-Requisite:** | *There is no data given to system.* | | |
| **Step No.** | **Execution description** | **Procedure result** | |
| 1 | *User clicks button prematurely.* | *System displays message.* | |
| **Comments:** Requirement meets | | | |
| **Passed** | | | |

**Table 7.8: Map Zoom In and Zoom Out Test Case**

| **Map Zoom In and Zoom Out Test Case** | | | |
|---|---|---|---|
| **None** | | | |
| **Test Case ID:** | *8* | **QA Test Engineer:** | *Ghulam Mustafa* |
| **Test case Version:** | *1* | **Reviewed By:** | *Ahmed Javed* |
| **Test Date:** | *15 November 2025* | **Use Case Reference(s):** | *None* |
| **Revision History:** | *None* | | |
| **Objective:** | *Ensure that clicking on does not produce broken report.* | | |
| **Product/Ver/ Module:** | *GIS-based Dashboard Module.* | | |
| **Environment:** | *Web page is running on web browser.* | | |
| **Assumptions:** | *None* | | |
| **Pre-Requisite:** | *Map is loaded already and has stable internet.* | | |
| **Step No.** | **Execution description** | **Procedure result** | |
| 1 | *User clicks zoom-in button.* | *System makes changes accordingly.* | |
| **Comments:** Requirement meets | | | |
| **Passed** | | | |

**Table 7.9: Google Earth Engine API Call Test Case**

| **Google Earth Engine API Call Test Case** | | | |
|---|---|---|---|
| **None** | | | |
| **Test Case ID:** | *9* | **QA Test Engineer:** | *Ahmed Javed* |
| **Test case Version:** | *1* | **Reviewed By:** | *Ghulam Mustafa* |
| **Test Date:** | *17 November 2025* | **Use Case Reference(s):** | *None* |
| **Revision History:** | *None* | | |
| **Objective:** | *Ensure that GEE-API calls work correctly.* | | |
| **Product/Ver/ Module:** | *GIS-based Dashboard Module.* | | |
| **Environment:** | *Web page is running on web browser.* | | |
| **Assumptions:** | *None* | | |
| **Pre-Requisite:** | *Correct url is used and have stable internet.* | | |
| **Step No.** | **Execution description** | **Procedure result** | |
| 1 | *User clicks button.* | *System loads the map.* | |
| **Comments:** Requirement meets | | | |
| **Passed** | | | |

## 7.3 Test Metrics

Summary of common attributes of test case metrics is explained in Table 7.10.

**Table 7.10: This table shows the summary of our test cases.**

| **Metric** | **Purpose** |
|---|---|
| **Number of Test Cases** | 9 |
| **Number of Test Cases Passed** | 9 |
| **Number of Test Cases Failed** | 0 |
| **Test Case Defect Density** | 0 |
| **Test Case Effectiveness** | 0 |

## 7.4 Conclusion

The test cases designed and executed for the Mosquito Surveillance System's front-end successfully validate the core functionalities of the system, including user interaction with the GIS-based map, report generation through LLM integration, and key UI behaviors such as zooming and querying. All tested components, such as zoom in/out, data visualization, and response generation, performed as expected under normal usage conditions.

# Chapter 8 Experimental Results and Discussion

We will discuss the experiements which we had applied during the development of this system. We also attached the different screenshots of different phases that we had implemented during complete data science lifecycle. The data science life cycle includes data collection to model deployments, and development of front-end.

## 8.1 Sentinel-2 Imagery Data Collection

Satellite imagery collection was the first stage because the detection of habitat detection was performed on them. Data Annotation and model training is performed on them. In data preprocessing phase we use different transformation. The images were filtered for minimal cloud coverage. The raw data was collected through the GEE and pre-processed to extract only the relevant bands (Band 2, Band 3, Band 4). These bands were selected for enhancement, which are known mosquito breeding indicators.

**Figure 8.1: Sentinel-2 Imagery of Lahore City with 10 Percent Cloud Coverage**

## 8.2 Image Annotation for Habitat Detection

After collecting the imagery, we proceeded to annotate the data to train the model for mosquito habitat identification. Manual annotation was performed using the QGIS platform. We use ”water”, ”low-vegetation”, ”habitat”, ”tree”, ”building”, ”impervious surface”, and ”clutter background” as annotation labels.

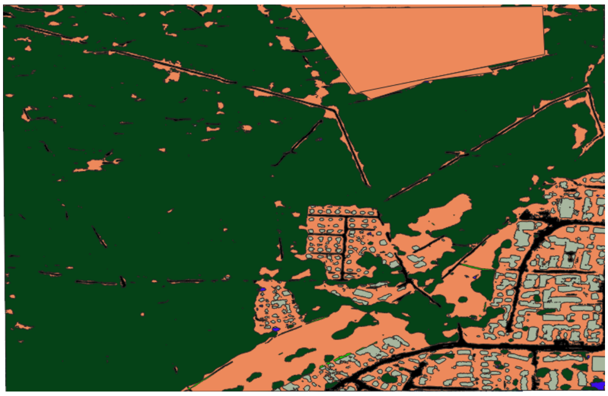

**Figure 8.2: Annotation of Sentinel-2 Imagery Using QGIS**

## 8.3 Google Earth Engine Script Details

GEE script is designed to collect Sentinel-2 imagery for a specified rectangular region in Lahore, focusing on March 2024. To ensure data quality, it filters images with less than 10 percentage cloud coverage using "CLOUDY-PIXEL-PERCENTAGE". The script selects only the B2, B3, and B4 bands to create a true-color composite. A median composite of all filtered images is generated and clipped to the ROI. This composite is then centered on the map with a zoom level of 13 for better visualization, and finally exported to Google Drive as a GeoTIFF image with a 10-meter resolution for further analysis.

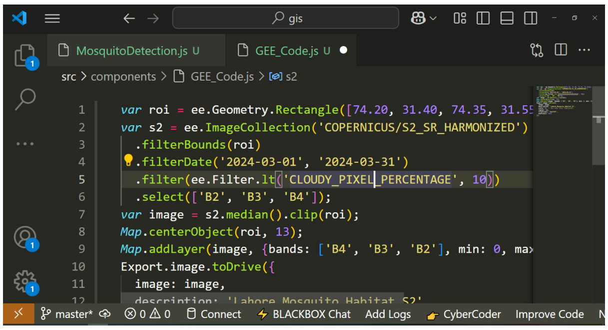


**Figure 8.3: GEE Script to Collect Sentinel-2 Imagery with Three Bands**

## 8.4 GIS Visualization of Mosquito Breeding Habitats

This GIS visualization map displays the geospatial distribution of mosquito breeding habitats across Singapore, specifically focusing on Dengue and Aedes mosquito breeding sites. This the demo map which is visualized using QGIS with OSM base layer. This map does not have the feature to filter the markers to districts level because we did not find any district level relevant data on the internet. That is why QGIS and OSM tool is used to display this demo.

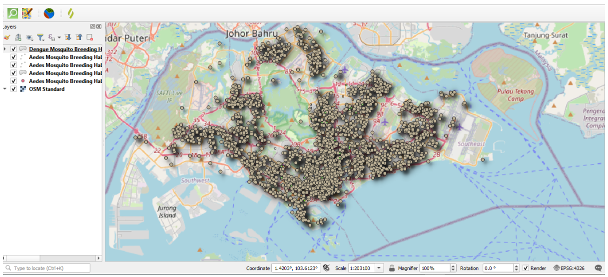


**Figure 8.4: GIS-Based Visualization of Mosquito Breeding Habitats in Singapore**

## 8.5 GIS-based Dashboard

Dashboard built with React and FastAPI.It contains different charts such as linke chart, histogram, pie-chart, and a table to show historical data in representable and understandable way.

```
import React, { useState, useEffect } from "react";
import ChatBox from "./ChatBox";
import MosquitoDetection from "./MosquitoDetection";
import ThemeToggle from "./ThemeToggle";
import ReportGeneration from "./ReportGeneration";
import MosquitoMap from "./Mosquito_Map";
import "./GISDashboard.css";
Tabnine | Edit | Explain
const GISDashboard = () => {
  const [messages, setMessages] = useState([]);
  const [showDetectionPage, setShowDetectionPage] = useState(false);
  const [theme, setTheme] = useState("dark");
  const [detectionData, setDetectionData] = useState(null);
  const [showMap, setShowMap] = useState(false);
  const { generateReport } = ReportGeneration();
  const handleGenerateReport = () => {
    generateReport(messages, detectionData);
  };
```

**Figure 8.5: React.js Used to Design the GIS-Based Dashboard**

## 8.6 Conclusion

Starting from Sentinel-2 satellite imagery collection. System uses true-color bands (B2, B3, B4) to potential breeding. Challenges like cloud coverage and large file handling, the image preprocessing and manual labeling enabled accurate training data creation for our habitat detection model.

# Chapter 9 Conclusion and Work Completed

Project is a solution that was trransforming mosquito monitoring with machine learning and LLM. The system is structured into three main modules: Mosquito Habitat Detection, and a Dashboard, LLMs. In phase one, we was keeping the system lightweight for initial testing. One of the biggest hurdles was the prompts for LLMs that often gave us the right answers. The reason was the use of lightweight models because of hardware contrasints.

In FYP-II, we successfully implement the functionality of each module and developed a production-ready system. The dashboard was enhanced with environmental data representation using different charts and plots, and habitat detection were improve and scaled using additional image sources. We refined the LLM integration to enable deeper, user-specific interactions and response was getting better. Extensive testing was conducted to ensure system robustness, while the user interface were being optimized for public health officials.

In addition, We decided to incorporate a 3 threat-level prediction system that displays risk zones in 3 levels(High, Intermediate, and Low) . As far as LLM development is concerened, we have completed our chat bot that handles the user questions. The last module was about the dashboard development and for this purpose we performed compelete data science life cycle such as data collection, data cleaning-preprocessing, conducted EDA, and then imported the cleaned data on MYSQL Workbench and then connected it with our frontend using FastAPI.FYP-II output is a excellent solution for mosquito-borne disease monitoring and prevention, effectively bridging AI innovation with public health needs.

In conclusion, system shows use of emerging technologies for a important public health application. Through FYP-I, we have builded a solid foundation that validates the system's concept and demonstrate its initial effectiveness. The integration of LLMs for intelligent querying, satellite imagery for habitat detection, and deep learning for mosquito classification has been proving to be both feasible and impactful. As we transition into FYP-II, we are committed to scaling, refining, and optimizing the system to ensure it will serves real-world needs. Our ultimate goal is to deliver an innovative, user-friendly, and intelligent surveillance system that can contribute to mosquito control efforts and reducin the spread of mosquito-borne diseases. We believe this project not only offers technical novelty but also holding the potential to significantly assist public health initiatives across vulnerable regions. With continued development, our system could become a model framwork. In addition to it, our system is ready to deploy the system on cloud for real-world use by the audience.

# Bibliography

[1] M. F. A. Sayeedi, F. Hafiz, and M. A. Rahman, "Mosquitofusion: A multiclass dataset for real-time detection of mosquitoes, swarms, and breeding sites using deep learning," *arXiv preprint arXiv:2404.01501*, 2024.

[2] A. C. Kinney, S. Current, and J. Lega, "Aedes-ai: Neural network models of mosquito abundance," *PLoS Computational Biology*, vol. 17, no. 11, p. e1009467, 2021.

[3] I. Kiskin, M. Sinka, A. D. Cobb, W. Rafique, L. Wang, D. Zilli, B. Gutteridge, R. Dam, T. Marinos, Y. Li, *et al.*, "Humbugdb: a large-scale acoustic mosquito dataset," *arXiv preprint arXiv:2110.07607*, 2021.

[4] A. Semwal, L. M. J. Melvin, R. E. Mohan, B. Ramalingam, and T. Pathmakumar, "Ai-enabled mosquito surveillance and population mapping using dragonfly robot," *Sensors*, vol. 22, no. 13, p. 4921, 2022.

[5] S. Lee, H. Kim, and B.-K. Cho, "Deep learning-based image classification for major mosquito species inhabiting korea," *Insects*, vol. 14, no. 6, p. 526, 2023.

[6] S. Elango, N. Ramachandran, and R. Low, "Autonomous mosquito habitat detection using satellite imagery and convolutional neural networks for disease risk mapping," *arXiv preprint arXiv:2203.04463*, 2022.

[7] J. Aira, T. Olivares, F. M. Delicado, and D. Vezzani, "Mosquiot: A system based on iot and machine learning for the monitoring of aedes aegypti (diptera: Culicidae)," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–13, 2023.

[8] W.-L. Liu, Y. Wang, Y.-X. Chen, B.-Y. Chen, A. Y.-C. Lin, S.-T. Dai, C.-H. Chen, and L.-D. Liao, "An iot-based smart mosquito trap system embedded with real-time mosquito image processing by neural networks for mosquito surveillance," *Frontiers in Bioengineering and Biotechnology*, vol. 11, p. 1100968, 2023.

[9] N. Javed, A. J. López-Denman, P. N. Paradkar, and A. Bhatti, "Flighttrackai: a robust convolutional

neural network-based tool for tracking the flight behaviour of aedes aegypti mosquitoes," *Royal Society Open Science*, vol. 11, no. 10, p. 240923, 2024.

[10] M. Ferraguti, S. Magallanes, C. Mora-Rubio, D. Bravo-Barriga, F. de Lope, and A. Marzal, "Landscape and climatic factors shaping mosquito abundance and species composition in southern spain: A machine learning approach to the study of vector ecology," *Ecological Informatics*, vol. 84, p. 102860, 2024.

[11] G. Carrasco-Escobar, M. Moreno, K. Fornace, M. Herrera-Varela, E. Manrique, and J. E. Conn, "The use of drones for mosquito surveillance and control," *Parasites & vectors*, vol. 15, no. 1, p. 473, 2022.

[12] K. Nolte, F. G. Sauer, J. Baumbach, P. Kollmannsberger, C. Lins, and R. Lühken, "Robust mosquito species identification from diverse body and wing images using deep learning," *Parasites & vectors*, vol. 17, no. 1, p. 372, 2024.

[13] V. Kittichai, T. Pengsakul, K. Chumchuen, Y. Samung, P. Sriwichai, N. Phatthamolrat, T. Tongloy, K. Jaksukam, S. Chuwongin, and S. Boonsang, "Deep learning approaches for challenging species and gender identification of mosquito vectors," *Scientific reports*, vol. 11, no. 1, p. 4838, 2021.

[14] A. Supratak, P. Haddawy, M. S. Yin, T. Ziemer, W. Siritanakorn, K. Assawavinijkulchai, K. Chiamsakul, T. Chantanalertvilai, W. Suchalermkul, C. Sa-Ngamuang, *et al.*, "Mosquitosong+: A noise-robust deep learning model for mosquito classification from wingbeat sounds," *Plos one*, vol. 19, no. 10, p. e0310121, 2024.

[15] K. Kim, J. Hyun, H. Kim, H. Lim, and H. Myung, "A deep learning-based automatic mosquito sensing and control system for urban mosquito habitats," *Sensors*, vol. 19, no. 12, p. 2785, 2019.

[16] M. Kuchavaram and Y. Huang, "Gis-based platform for near real-time prediction of mosquito distribution," *Transactions in GIS*, vol. 29, no. 1, p. e70009, 2025.

[17] R. Charoenpanyakul, V. Kittichai, S. Eiamsamang, P. Sriwichai, N. Pinetsuksai, K. M. Naing, T. Tongloy, S. Boonsang, and S. Chuwongin, "Enhancing mosquito classification through self-supervised learning," *Scientific Reports*, vol. 14, no. 1, p. 27123, 2024.

[18] L. Zou, S. N. Miller, and E. T. Schmidtmann, "Mosquito larval habitat mapping using remote sensing and gis: implications of coalbed methane development and west nile virus," *Journal of medical entomology*, vol. 43, no. 5, pp. 1034–1041, 2006.

[19] A. Goodwin, S. Padmanabhan, S. Hira, M. Glancey, M. Slinowsky, R. Immidisetti, L. Scavo, J. Brey, B. M. M. Sai Sudhakar, T. Ford, *et al.*, "Mosquito species identification using convolutional

neural networks with a multitiered ensemble model for novel species detection," *Scientific reports*, vol. 11, no. 1, p. 13656, 2021.

[20] M. S. Yin, P. Haddawy, B. Nirandmongkol, T. Kongthaworn, C. Chaisumritchoke, A. Supratak, C. Sa-Ngamuang, and P. Sriwichai, "A lightweight deep learning approach to mosquito classification from wingbeat sounds," in *Proceedings of the conference on information technology for social good*, pp. 37–42, 2021.