Epidemiology GIS

Project No.2

Punyawat	Phuwawatthanasate	58070501095	Punyawat.cpe@mail.kmutt.ac.th
Supornthip	Sethavarangura	58070503466	Supornthip.set@mail.kmutt.ac.th

Advisor: Dr. Sally E.Goldin

Friday, 17 May 2019

____

I have read this report and approve its content.

Epidemiology GIS

Mr. Punyawat Phuwawatthanasate
Miss Supornthip Sethavarangura

A Project Submitted in Partial Fulfillment of the Requirements
for the Degree of Bachelor of Engineering
Department of Computer Engineering, Faculty of Engineering
King Mongkut’s University of Technology Thonburi
Academic Year 2018

Epidemiology GIS

Mr. Punyawat Phuwawatthanasate
Miss Supornthip Sethavarangura

A Project Submitted in Partial Fulfillment of the Requirements
for the Degree of Bachelor of Engineering
Department of Computer Engineering, Faculty of Engineering
King Mongkut's University of Technology Thonburi
Academic Year 2018

Project Committee

……………………………………………………… Advisor
(Sally E.Goldin, Ph.D.)

……………………………………………………… Committee
(Asst. Prof. Phond Phunchongharn, Ph.D.)

……………………………………………………… Committee
(Assoc. Prof. Naruemon Wattanapongsakorn, Ph.D.)

……………………………………………………… Committee
(Asst. Prof. Nuttanart Facundes, Ph.D.)

Project Title	Epidemiology GIS
Project Credit	3 credits
Project Participant	Mr. Punyawat Phuwawatthanasate
	Miss Supornthip Sethavarangura
Advisor	Dr.Sally E.Goldin
Degree of Study	Bachelor's Degree
Department	Computer Engineering
Academic Year	2018

Abstract

In recent years, many countries have suffered from infectious diseases. Examples of well-known, potentially fatal diseases are bird flu, rabies and dengue fever which have spread and killed many of people in many areas around the world. In our opinion, the deadly infectious diseases are not able to be ignored because one case of an infectious disease can be the origin of many cases.

We have decided to build GIS software in order to store, display and analyze the spatial distribution of an infectious disease. We also would like to build models i.e. spatial regression, etc. to make predictions about future disease incidence.

In this project, there are research in epidemiology, geography, statistics, prediction model(s), gathering sample dataset(s) from Health Organization(s), software components design and defining their functions. We hope the software is able to decrease number of people who die or suffer from diseases by making better prevention plans and strategies.

หัวข้อโครงงาน	ระบบสารสนเทศภูมิศาสตร์เพื่อการวิเคราะห์การระบาดของโรคระบาด
จัดทำโดย	นายปุณยวัจน์ ภูววัฒนาเศรษฐ์
	นางสาวสุพรทิพย์ เศรษฐวรางกูร
อาจารย์ที่ปรึกษา	Dr.Sally E.Goldin
ระดับการศึกษา	วิศวกรรมศาสตรบัณฑิต
ภาควิชา	วิศวกรรมคอมพิวเตอร์
ปีการศึกษา	2561

บทคัดย่อ

ในช่วงประวัติศาสตร์ของมนุษย์เคยพบกับเหตุการณ์ที่ถือว่าเป็นวิกฤติของมนุษยชาติอยู่นับไม่ถ้วน ตั้งแต่อดีตจนถึงปัจจุบันศัตรูของมนุษย์ที่ยังคงคุกคามอยู่ตลอดคือ โรคระบาด

โรคระบาดรุนแรงที่เคยเกิดขึ้นในช่วงไม่สิบปีที่ผ่านมา เช่น ไข้หวัดนก และ ไข้เลือดออก ซึ่งคร่าชีวิตคนไปเป็นจำนวนมาก ดังนั้นการรับมือกับโรคระบาดที่ทันท่วงทีและมีประสิทธิภาพจึงเป็นสิ่งสำคัญ พวกเราจึงตัดสินใจทำระบบสารสนเทศภูมิศาสตร์ (GIS) ขึ้นมาเพื่อใช้ในการเก็บข้อมูล วิเคราะห์ และแสดงผลลัพธ์ในเชิงภูมิศาสตร์ นอกจากนั้นเรายังใช้โมเดลพยากรณ์การเกิดของโรคในอนาคต เช่น Spatial Regression

ในการทำโปรเจคเราจะศึกษาเรื่องระบาดวิทยาเบื้องต้น ภูมิศาสตร์ สถิติ รวมถึงการพยากรณ์โรค และการออกแบบซอฟแวร์ คณะผู้จัดทำหวังว่าซอฟแวร์ตัวนี้จะสามารถช่วยวางแผนป้องกัน สร้างวิธีการที่ดีและมีประสิทธิภาพยิ่งขึ้น เพื่อลดจำนวนผู้ที่ป่วยหรือเสียชีวิต

Acknowledgements

We would like to express our appreciation who gave us possibility to complete this project. A special thanks to our advisor Dr. Sally E.Goldin who has full effort and dedication for providing suggestions, and comments.

We are also grateful to Tatchatan Tangam, Public Health Technical Officer, at Watbot MOPH. We would like to thank him for sharing us some ideas to our work and how officer works on outbreak prevention.

And another special thanks to Teeravadee Korphayakkhin who gave us suggestions and share her related study.

List of Contents

List of Contents (Continue)

List of Contents (Continue)

List of Figures

List of Figures (Continue)

List of Figures (Continue)

List of Figures (Continue)

Figure **Page**

List of Tables

Chapter 1

Introduction

1.1 Problem Statement and Approach

In recent years, many countries have suffered from infectious diseases. Examples of well-known, potentially fatal diseases are bird flu, rabies and dengue fever which have spread and killed many of people in many areas around the world. In our opinion, the deadly infectious diseases should not be ignored because one case of an infectious disease can be the origin of many cases. In April 2009, H1N1 virus started to spread in North America, and it also spread around the world rapidly. By the end of December 2009, more than 12,220 persons around the world had died, and 191 persons were killed in Thailand [1]. Therefore, we have been interested in this topic in order to reduce and prevent loss of life.

In this project, we have decided to build GIS software in order to store, display and analyze the spatial distribution of an infectious disease. We also would like to build models i.e. spatial regression, etc. to make predictions about future disease incidence. Therefore, this can be considered as a project to benefit society.

We hope the software is able to decrease number of people who die or suffer from diseases by making better prevention plans and strategies.

1.2 Objectives

- To store, display, analyze and predict the spread of disease with some specific information.
- To help the government organizations to make plans for decreasing number of deaths from infectious diseases.
- To learn about spatial data representation, display and analysis

1.3 Scope

- The software will be based on open source software components.
- The software will be designed so that it can be adapted to different geographic areas and different diseases.

1.4 Tasks and Schedule

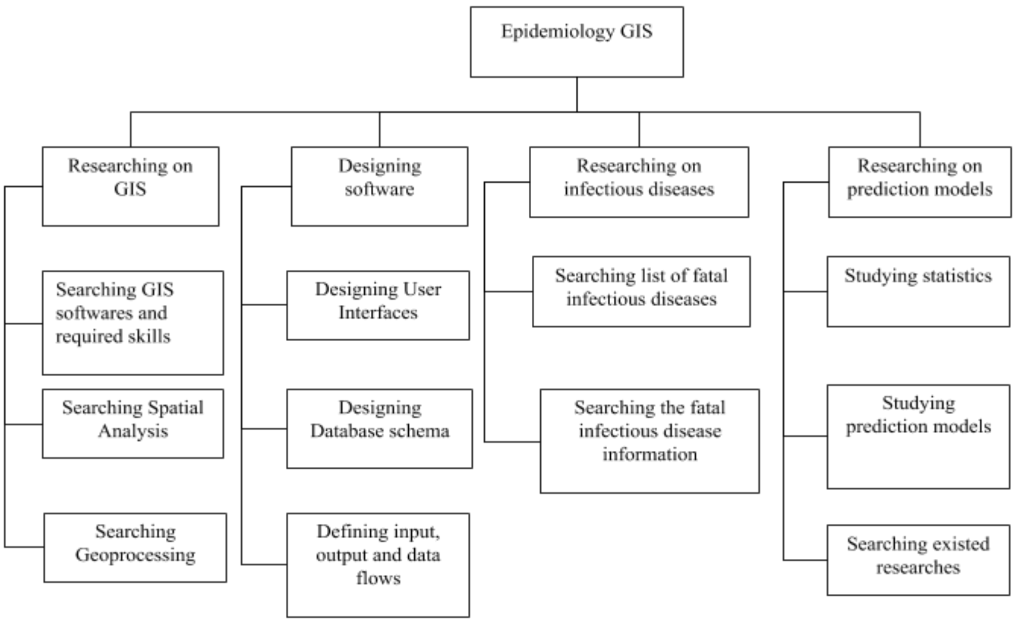

Table 1.1 Schedule and time plan of semester 1

Task	Start Date	End Date	Duration (Days)	August				September				October				November			
				1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Meet advisor and writing project idea	10/08/2018	25/08/2018	16																
Create proposal, deliverables and project management draft	26/08/2018	07/09/2018	12																
Research on GIS	08/09/2018	17/09/2018	10																
Research on Infectious diseases	13/09/2018	19/09/2018	7																
Research on related technology	19/09/2018	25/09/2018	7																

Writing and revision report	26/09/2018	01/10/2018	6
Create presentation	02/10/2018	09/10/2018	8
Exam - Proposal Presentation & Preparation	10/10/2018	12/10/2018	3
Research on prediction models and statistics	13/10/2018	22/10/2018	10
DB schema design	19/10/2018	24/10/2018	6
Architecture design	21/10/2018	30/10/2018	10
UI design	27/10/2018	06/11/2018	11
Design verification	07/10/2018	08/10/2018	3
Semester Report	09/11/2018	15/11/2018	7
Presentation / Exam	16/11/2018	30/11/2018	15

Table 1.2 Schedule and time plan of semester 2

Task	Start Date	End Date	Duration (Days)	January				February				March				April				May			
				1	2	3	1	1	2	3	4	1	1	1	4	1	2	3	4	1	2	3	4
Discuss progress with advisor	15/01/2019	30/01/2019	15																				
Create progress report and deliverable & plan	20/01/2019	31/01/2019	12																				
implement hotspot analysis	01/02/2019	15/02/2019	15																				
Implement layers to be displayed on the map	01/02/2019	15/02/2019	15																				
Implement to show hotspot analysis on the map	15/02/2019	28/02/2019	14																				
Work on midterm report	20/02/2019	03/03/2019	12																				
Implement prediction model	01/03/2019	20/03/2019	20																				
Implement to show prediction model result on the map	21/03/2019	31/03/2019	11																				
Meet stakeholder	01/04/2019	15/04/2019	15																				
Improve the web map to meet goals	15/04/2019	30/04/2019	15																				
Work on final report	20/04/2019	03/05/2019	14																				
Get final report review from advisor	04/05/2019	12/05/2019	9																				
Improve the final report	04/05/2019	12/05/2019	9																				
Prepare for presentation day	13/05/2019	26/05/2019	14																				
Final presentation	27/05/2019	30/05/2019	4																				

1.5 Deliverables for 1st Semester

- Identified boundary of area
- Select infection disease
- Disease data set
- Architectural design
- UI design
- Schema for database
- Selected model

1.6 Deliverables for 2nd Semester

- A map software with map, prediction model, database, result of usability testing, and results of analysis testing/experiment.
- The software will have disease cases on the map, and also predict spread area by using data in the database
- The database will have infection disease cases and area details i.e. population, climate, etc.

Chapter 2

Background, Theory and Related Research

2.1 Core Concepts

2.1.1 Spatial Epidemiology

Spatial epidemiology is used to describe and analyze geographic variations in disease with respect to demographic, environmental, behavioral, socioeconomic, genetic, and infectious risk factors.

Today, most studies apply epidemiologic principles and analytical methods for understanding the cause of a disease in a study area. Therefore, the development of spatial epidemiology is driven by introduction of spatial statistical and geostatistical methods and the application of geographic information systems.

2.1.2 Epidemiology

Epidemiology is defined as the study about cause, incidence, and distribution of disease or other health problems in studied populations. The main objectives of epidemiology are to identify cause and factors of disease that increase a person's disease risk, to determine the disease extent, and to study the development of a health problem over time and space.

There are some terms that are often used in epidemiology studies. *Population at risk* is the groups of people who might catch the studied disease. *Determinant* is any factor that affects the incidence of a disease in a studied population. For example, determinants of cholera are water and bacteria that is called *Vibrio cholerae*.

2.1.3 Disease Mapping

Disease mapping is using a map to investigate distribution or spread of infection in selected areas and to make inferences about the determinants. John Snow's map of cholera cases in London is one of the first known examples of disease mapping in epidemiology. During the cholera epidemic in 1854, he collected information about the location of individual cases throughout the city. He started by representing each death as a bar at the location where it occurred, as shown in the Figure 2.1.

Figure 2.1 John Snow's cholera map of Soho [2]

Before Snow's work, nobody understood the mechanism for the transmission of cholera. From creating the map, Snow found that it obviously clustered around the pump in Broad Street. He concluded that cholera is spread by water.

2.1.4 Spatial Regression

Spatial Regression is statistical modelling that measures the strength of relationship between an independent variable in a location and a dependent variable at same location. The strength of relationship could be used to estimate the value of a dependent variable based on independent variables in nearby locations. For example, if you have the locations of infection disease, you can use spatial regression to understand the likelihood of spread to a new location.

In spatial way, one observed value in a specific location may related on the observed value on nearby area. That is called *spatial dependence*. Once the observed data need to be deployed in mathematical model. If it is considered as spatially lagged dependent variable $\boxed{(\boldsymbol{Wy})}$ which referred to Spatial Lag Model. Then spatial lag term is treated as an endogenous variable. This model will be used when we know that observed values of location i influence the observed values of nearby location i. The formula can be expressed as

$$\boldsymbol{y} = \rho \boldsymbol{W} \boldsymbol{y} + \boldsymbol{X}\beta + \varepsilon$$

where ρ is spatial autoregressive parameter that indicates the strength of spatial dependence. W is $N \times N$ spatial weight matrix, each element w_{ij} represent as weight of correlation, value $\boldsymbol{y}$ at location i related to value at location j. $\boldsymbol{y}$ is $N \times 1$ vector of observation on dependent variable. $\boldsymbol{X}$ is $N \times k$ matrix of explanatory variable, sometimes may referred as regressor or input variable, in other word it is equivalent to influence factor. β is regressor coefficient. ε is error term, this represent as sum of all other factors that affect dependent variable.

Another well-known model is Spatial Error Model. This model treats spatial dependence as disturbance term. Then this term is treated as an endogenous variable. It properly fit when adjusting the biasing influence of spatial autocorrelation. This could be written as

$$\boldsymbol{y} = \boldsymbol{X}\beta + \varepsilon$$
$$\varepsilon = \lambda \boldsymbol{W} \varepsilon + u$$

where λ is spatial autoregressive coefficient of error lag. u is residual.

It can be concluded that spatial regression is about value prediction, and the spatial autocorrelation is used to show the relationship between predicted values of the same variable in different locations. If the data is spatially correlated, it can be used to predict the value at one location based on the value that is sampled from a nearby location.

2.1.5 Hot Spot Analysis

Hot Spot Analysis is a set of techniques to identify locations of statistically significant spatial clusters of cases in your data. For example, if there are many cases of tuberculosis in a village, the village may be a hot spot.

A statistically significant hot spot is a spot that is surrounded by other features with high values. A feature means a row in a table (dataset) that analyst use with the hot spot analysis. For example, a feature could be the number of cases of dengue. Therefore, a high value of feature without surrounding by the others cannot be a statistically significant hot spot. A cold spot is an area that has fewer case occurrences than would be expected by chance. The high or low value can be analyzed by many methods. One of the popular methods is Getis-Ord Gi* (G-I-star).

The Getis-Ord Gi* (G-I-star) is used to look at each feature in the dataset within the context of neighboring features in the same dataset. In other word, it is used to identify locations of statistically significant hot spots. The results of G-I-star analysis are called Z scores. Result of Z is able to tell where the features cluster spatially. The Z score is used to indicate a hot spot or a cold spot. A hot spot is indicated by a high Z score (or positive Z-score). When a Z score is low negative, a cold spot is indicated. Therefore, the density of clustering can be measured by the Z score. A Z score near zero is definitely no spatial clustering.

G statistic is derived base on geography knowledge, a region was surrounded by nearby regions likely to be dependent each other's and rural regions tend to independent. To enable the statistic more realistically, concept of distance needs to be implemented. Physical distance is the simplest way to create the relation between point i and j, travel time could be used also. Any points were located within distance d of point i are dependent. The simplest form of weight is binary weight, points within d are weighted 1 and the remaining are 0

Hot spot analysis is important in disease mapping and epidemiology because it shows location where many cases have occurred. In other words, it can show the clustering location. Once hot spots have been identified, the analyst can consider what environmental or social factors might cause the disease to have high incidence in that particular area.

$$G_i^* = \frac{\sum_{j=1}^{n} w_{i,j} x_j - \bar{X} \sum_{j=1}^{n} w_{i,j}}{S \sqrt{\frac{\left[n \sum_{j=1}^{n} w_{i,j}^2 - \left(\sum_{j=1}^{n} w_{i,j} \right)^2 \right]}{n-1}}}$$

Figure 2.2 The Getis-Ord statistic [3]

In Figure 2.2, it shows The Getis-Ord Gi* (G-I-star) which its notations are:

x_j is the attribute value for feature j

$w_{i,j}$ is the spatial weight between points i and j

n is the total number of features

$\bar{x}$ is sum of all x_j (x_j included) within distance of point i, written as $\frac{\sum_{j=1}^{n} x_j}{n}$

S is standard deviation, written as $\sqrt{\frac{\sum_{j=1}^{n} x_j^2}{n} - \bar{x}}$

2.1.6 Geographic Information Systems

A geographic information system (GIS) is a system that is designed to capture, store, manipulate, analyze, manage, and present all types of spatial data. GIS can be used in many ways, e.g. scientific investigations, resource management, asset management, environmental impact assessment, urban planning, criminology, history, sales, logistics, etc. For example, if a farmer wants to find where are the best locations to plant corn, he can find them by using geographical data that combines information about soils, topography, and rainfall.

Typically, GIS stores different types of information (e.g. population, district boundaries, elevation, land use) as different themes or layers. The GIS can display one or more layers in map-like format.

GIS also provides a set of tools for analyzing relationships and combining information. Usually these operations take one or more input layers, and produce output layer(s) as a result.

- Spatial queries are used to evaluate spatial relationships between geographic elements. The spatial relationships are detected to locate spatial objects. For example, a query might compute “what proportion of the inhabitants of Bangkok lives more than 100m away from a bus stop?”

- Buffer generation is an algorithm that is used to identify areas of a given distance surrounding geographic features. Buffer is useful for analysis problem, where you need to determine the affected area. For example, the pizza delivery service needs to make sure that customers are in operation area or not, where that area is defined in term of a certain distance from the pizza shop.

Figure 2.3 Example of creation buffer polygons around input features to a specified distance

- Overlay Layers is used to combine two layers into a single layer. There are 3 methods that can be used to combine: Intersect, Union, or Erase. This tool needs two inputs that are input layer and overlay layer. Method selection determines how the two inputs combine.

Table 2.1 Details of overlay methods [4]

Overlay Method	Input Features	Overlay Features
Intersect	Points, Lines, or Areas	Points, Lines, or Areas
Union	Areas	Areas

Erase	Areas	Points, Lines, or Areas

Intersect: it keeps the portions of the input that are overlapped by the overlay.

Union: It is combination of the input and the overlay

Erase: It is opposite from intersection. It deletes the portions of the input that are overlapped by the overlay.

Here is an example of Overlay layers. Suppose a development company wants to build a new golf resort in one of three centrally located counties in the state. Before they can begin planning, they need to determine whether there is enough privately owned land within those counties that they may be able to purchase for the resort. Overlay Layers can be used to remove the publicly owned lands from the selected counties.

- Network analysis. In GIS, networks consist of interconnected lines (or edges) and intersections (or junctions) that together represent routes. There are many types of network analysis. Three important types are:

  - Route analysis: Route analysis is used to compute travel directions between two points. It can also find the best route between many stops. Here, the best one could be the one that has shortest distance or the quickest or most scenic or the one with no highways.
  - Service areas analysis layer: It is used to calculate an area that can be reached from a particular location in the network based on time or distance. For instance, you could use service area analysis to determine where schools can be reached by an ambulance from any nearby hospital within 10 minutes.
  - Origin-destination cost matrix analysis layer: It is used to create a cost matrix by using multiple origins and multiple destinations, indicating the cost to travel from each origin to each destination. Once again cost could be interpreted based on total distance, travel time, or some other quantity.

A geographic information system (GIS) processes and analyzes spatial data. The spatial data is the data that consists of points, lines, polygons and other geographic and geometric data primitives, which can be located based on some coordinate system.

Spatial data can be represented as vector or raster data.

- A vector representation separates the spatial information (coordinates) from the attribute data, with the attribute data stored in a database. Here, spatial information means coordinates, and attribute data means values of whatever the layer is intended to represent, e.g. Rainfall.

  - **Points** are represented as a single pair of X,Y coordinates. They are used to represent locations on the ground when the features are too small to be represented by some region with area. For example, the locations of ATM machines or convenience stores on a city map, or the locations of tourist sites on a regional map might be stored as points.

Figure 2.4 Example of points [5]

  - **Lines** are usually used to represent natural or human-created linear features such as rivers, roads and transmission cables. A line is an ordered set of points (sometimes called vertices). Each line uses multiple pairs of X,Y coordinates (points) to make the shape of the line.

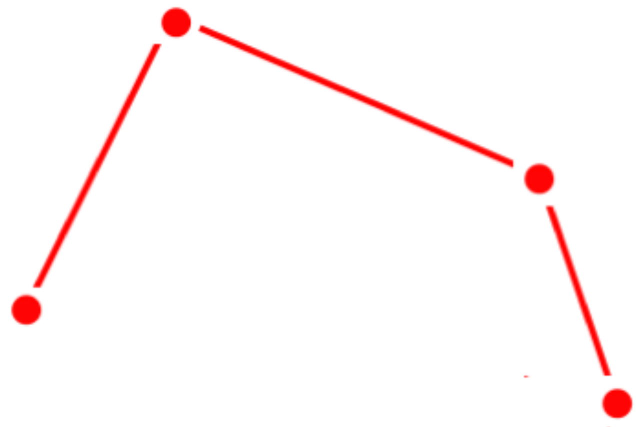

Figure 2.5 Example of lines [5]

- **Polygons** connect vertices and close the path. They are used to show bounded areas. To create a polygon, the first and the last point must be the same location as shown in Figure 2.6

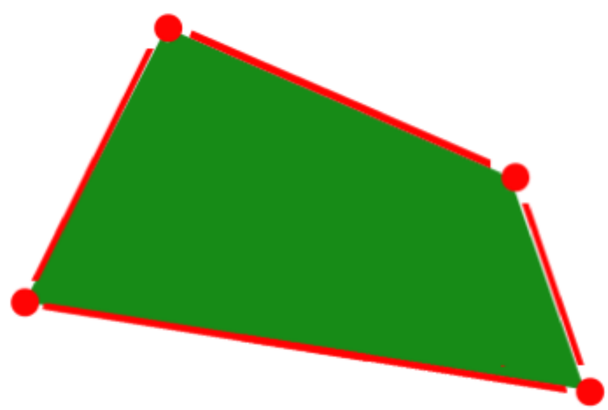

Figure 2.6 Example of polygon [5]

- Raster data represent the landscape as a rectangular matrix of cells, which are usually but not always square. Raster data is grid cells and made by pixels. Each pixel has a value that represents attribute of interest. Raster data layers can be divided into two categories: discrete and continuous.

- Discrete Raster uses attributes whose values are categories. For example, in Figure 2.7, the value 1 represents forest land, the value 2 is urban land, and the 3 value means agricultural land.

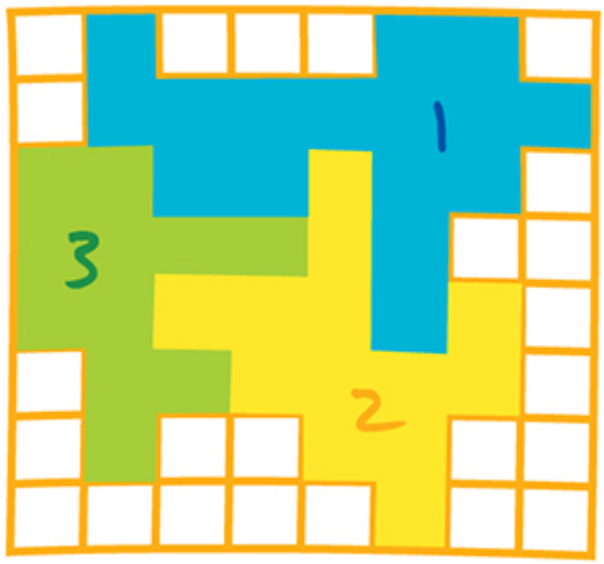

Figure 2.7 Example of discrete raster [6]

- Continuous raster is used for attributes that vary continuously, such as rainfall, temperature or elevation. Continuous raster layers most often represent physical quantities.

The choice of using vector or raster depends on how you conceptualize the features in the map. When you want to work with pixels, raster is the suitable representation. If you want to work with coordinates, a vector representation may be more appropriate. Map scale is also a factor that affects whether you should select vector or raster. Vector layers can scale objects up or down in size, but raster layers have a built-in scale that depends on the ground size of each cell or pixel. The last important factor is where the original data come from. For example, if the original data come from analyzing satellite or drone imagery, you should probably use raster because images are naturally in raster format, but for man-made or conceptual layers such as land ownership, district boundaries, roads, etc.), it is usually more suitable to use vector.

It is also possible to combine both types of representation or to convert from one to the others, although this may add error or noise to the data.

Note that in a vector representation, the type of feature (point, line or polygon) may depend on the scale. In a country-scale spatial data base, villages might be represented as points. However, a province-scale spatial data base might include the boundaries of villages, treating them as polygons. Similarly, a river might be treated as a linear feature in a small scale data base, but as a polygon at larger scale, where the locations of the two banks might be important.

2.2 Dengue Fever

Dengue fever (DF) is a vector-borne viral disease that is transmitted by biting of *Aedes* mosquitoes. There are four dengue virus serotypes, i.e. DEN-1, DEN-2, DEN-3, and DEN-4. Each virus interacts the antibodies in human blood serum differently. The virus infection may show no symptoms or show signs similar to ordinary fever.

Infected patients will have symptoms such as high fever, severe headaches, pain behind the eyes, severe joint pain, severe muscle pain, fatigue, nausea, vomiting, skin rash and mild bleeding. Four to six days after the patients is infected, those symptoms will begin. However, DF is rarely fatal, and patient could recover rapidly. DF may have produced symptoms of unusual bleeding. DF with bleeding should not be considered as DHF. The symptom that differentiates between DF and DHF is plasma leakage. Severe DHF can lead to circulatory failure and Dengue Shock Syndrome (DSS) which has high fatality [17]. Figure 2.8 illustrates the different symptoms of dengue virus infection.

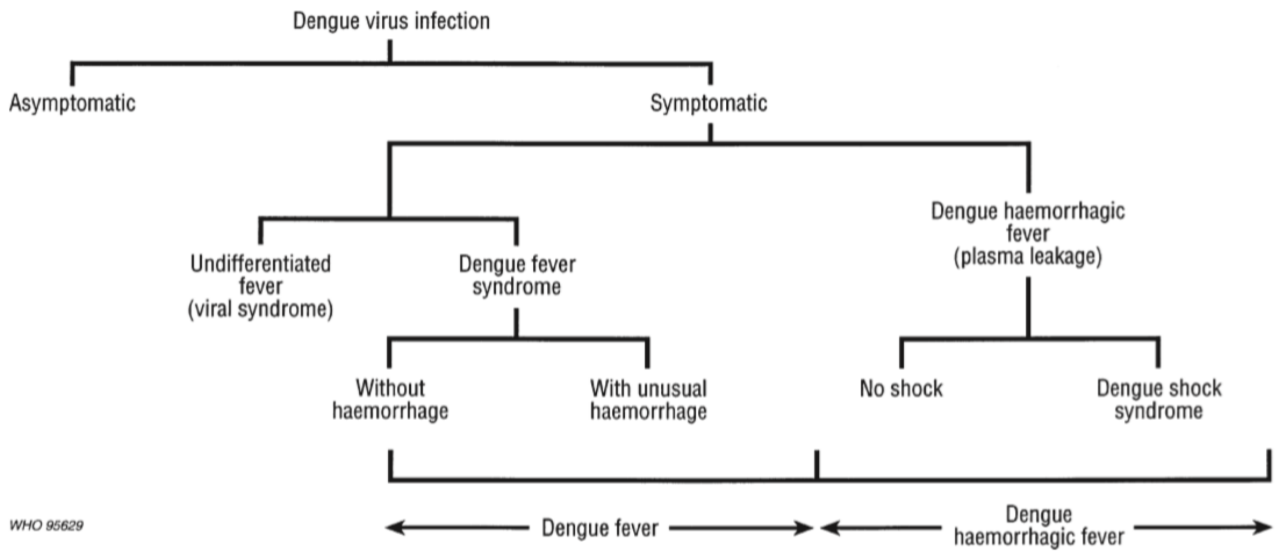

Figure 2.8 Manifestations of dengue virus infection [22]

2.2.1 Dengue environmental risk factors

According to research on dengue [18], many research articles show that the incidence and spreading of dengue is positively associated with various geographical and climatic factors.

Based on available research papers, these papers show the relationship between the incidences and the following factors; temperature and rainfall [7], El Niño–Southern Oscillation [8], degree of urbanization and warm months [9] and land elevation [10] and age.

The increasing of temperature could be described in term of biological effects. Laboratory results show that environment with high temperature affect mosquito size. They are smaller in size which means they could travel for longer distances. Biting rate is increased. Moreover, dengue virus activity is enhanced by high temperatures [7]. Also land elevation and humidity are positively associating to temperature.

Normal rainfall increases incidence rate by increasing the number and size of mosquito breeding sites. However, storms and heavy rain could have negative effects on larvae that have outdoor breeding sites because these sites will be destroyed by flooding or aggressive water flow or washed away.

So that our system is going to use additional factors for modelling: monthly rainfall, monthly average temperature, monthly relative humidity and land elevation.

2.3 Programming languages, libraries, software tools and environments

2.3.1 PostgreSQL and PostGIS

PostgreSQL is an open source object-relational database system. It has its own language which called Procedures Language / PostgreSQL (PL/pgSQL) and was designed to allow extension to be added to the basic database functionality. An object-relational database can store complex types of objects in relational tables and also allow users to create their own data types, functions and operators to works with new custom data types. Access to database can be done with scripting languages such as Perl (Perl DBI), PHP (integrated) and Python (psycopg2), or with compiled languages such as C or Java.

PostGIS is the powerful extension software for PostgreSQL that adds capabilities for storing, processing and querying for spatial objects. Moreover, spatial data that store in PostgreSQL can be accessed by using 3rd party software that use open standard such as Web Map Service (WMS).

PostGIS provides large number of extra features for PostgreSQL including spatial operators, spatial functions, and spatial data types.

2.3.1.1 Advantage of PostgreSQL

- Strong support from community
- Open-source licensed
- Excellent support for internationalization (storing non-ASCII text data)
- Comprehensive documentation
- Multi-platform support
- Supports extensions including PostGIS

2.3.1.2 Example of PostGIS function

- Create geometries by ST_GeomFromText

This function is used for creating geometric objects based on a text string that holds their coordinates. The example below shows how to create 3 different objects with coordinate reference by querying. These 3 objects are point, line and polygon. Figure 2.9 through 2.13 show simple functional tests we have done with PostGIS.

test on postgres@PostgreSQL 10 (x86)

```
SELECT * INTO geom_test FROM
    (VALUES
        (ST_GeomFromText('POINT(20 -60)', 4326)),
        (ST_GeomFromText('LINESTRING(50 10, 30 99)', 4326)),
        (ST_GeomFromText('POLYGON((-40 -40, -40 40, 40 40, -40 -40))', 4326))
    )As foo(geom)
```

Figure 2.9 Creating 3 geometriy objects into geom_test table in single query

	geom geometry
1	0101000020E610000000000000000034400000000000004EC0
2	0102000020E61000000200000000000000000049400000000000002440000000000000003E40000000000000
3	0103000020E610000001000000040000000000000000000044C000000000000044C000000000000044C000

Figure 2.10 Objects are stored and represented in PostgreSQL database

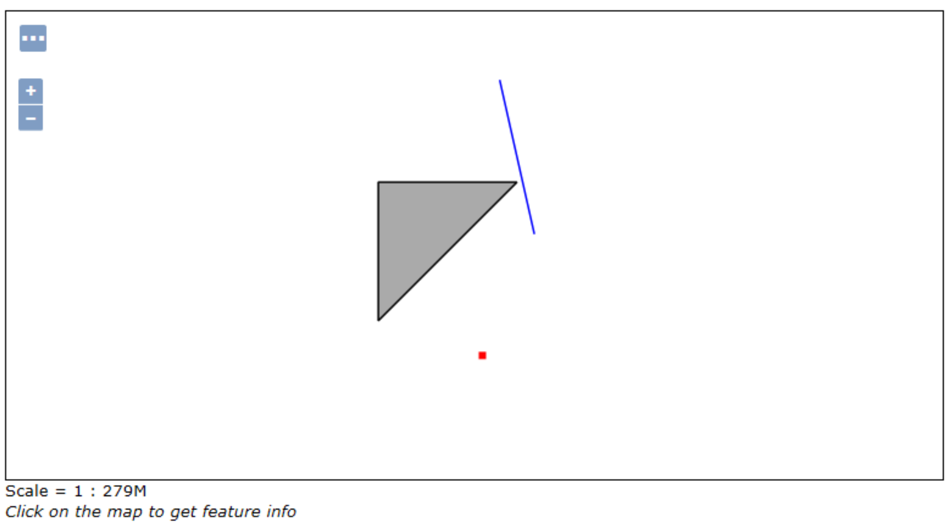

Figure 2.11 Objects are visualized on GeoServer via WMS

- Calculate area of polygons by ST_Area

This function used for calculating the area of polygon, from previous example, we have already created a polygon and two non-polygons.

test on postgres@PostgreSQL 10 (x86)

```
SELECT geom, ST_Area(geom)
FROM geom_test
```

Figure 2.12 Area calculation of 3 objects that have been already created

	geom geometry	st_area double precision
1	0101000020E6100000000000000000034400...	0
2	0102000020E610000002000000000000000...	0
3	0103000020E61000000100000004000000...	3200

Figure 2.13 Result of ST_Area, it returns area value of polygon objects

2.3.2 Leaflet

Leaflet is an open-source JavaScript library that is popular for developing interactive map applications on a browser. Leaflet allows developers to easily implement maps into software without GIS knowledge and it works well on various platforms. Leaflet comes with Web Map Service, GeoJSON and Vector layers support. Another a well-known library that is similar to Leaflet is OpenLayers. [19]

An example of working with Leaflet is shown in Figure 2.14

Figure 2.14 Using GeoJSON with Leaflet [19]

To create objects to pin on the map, let's start with adding maker

```
var marker = L.marker([51.5, -0.09]).addTo(mymap);
```

Figure 2.15 Adding a marker on map layer [21]

Adding a circle with specific parameters to customize circle.

```
var circle = L.circle([51.508, -0.11], {
    color: 'red',
    fillColor: '#f03',
    fillOpacity: 0.5,
    radius: 500
}).addTo(mymap);
```

Figure 2.16 Adding a red circle [21]

And adding a triangle shape polygon.

```
var polygon = L.polygon([
    [51.509, -0.08],
    [51.503, -0.06],
    [51.51, -0.047]
]).addTo(mymap);
```

Figure 2.17 Adding a triangle object [21]

Then we can see the result of adding objects.

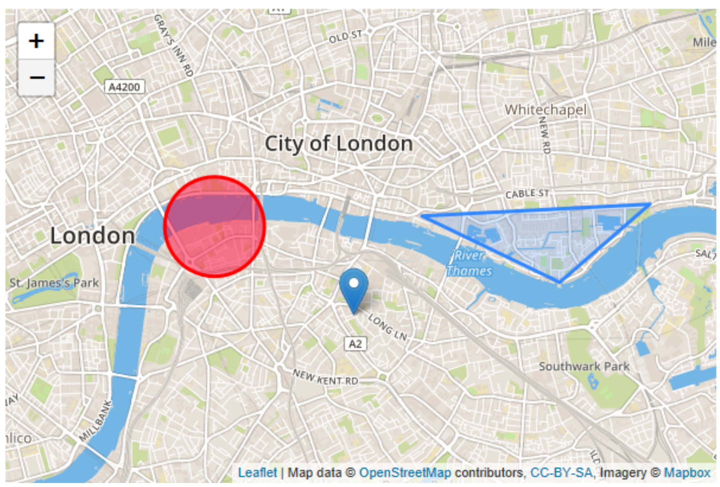

Figure 2.18 All three objects are created by simply coding [21]

Also, we can add a popup on a created object to attach some note about it. All of 3 type object contains different message. The message will show up when a user interacts with the object.

```
marker.bindPopup("<b>Hello world!</b><br>I am a popup.").openPopup();
circle.bindPopup("I am a circle.");
polygon.bindPopup("I am a polygon.");
```

Figure 2.19 Attach some information in 3 type objects [21]

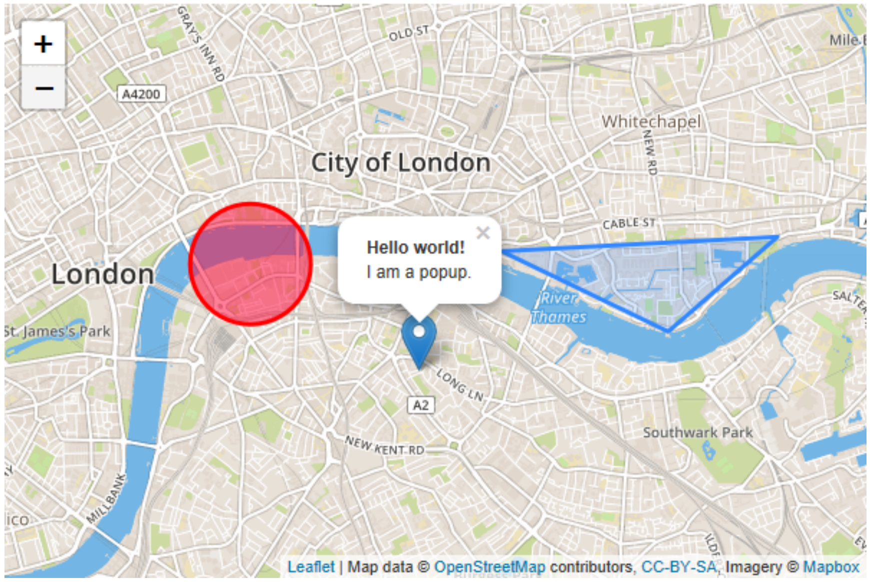

Figure 2.20 Example shows the result of previous steps [21]

2.3.3 OpenStreetMap

OpenStreetMap (OSM) is an editable online map site to which anyone can contribute to provide spatial data on map. It provides access to spatial information such as roads, transportation and points of interest with no cost for everyone. This contrasts with Google Maps which has a pricing plan for commercial use or using Google Map API.

OSM could provide us information and local database for our mapping system base layer by using site export feature. We can specify the location that we need information then export it. Leaflet will show and interact with theses information on map. However, export file is store in OSM format. It needs further conversion into PostGIS support format. Osm2pgsql will handle this transformation.

Alternately, Leaflet can access and display OSM data directly, if we just want the map as a background.

Figure 2.21 OSM could be use as base layer for our mapping system [21]

2.3.4 GeoServer

GeoServer is server-side software system written in Java that can be used to view, process and edit spatial data. It was designed to serve like ordinary HTTP server, but instead of publishing HTML, it provides a service and abstraction layer between HTTP server and geospatial data. It can connect and serve data source from many sources, and also support many data formats e.g., PostGIS, MySQL, Shapefile, GeoJSON. GeoServer implements Open Geospatial Consortium protocols such as Web Map Service and Web Feature Service. GeoServer itself does not store the spatial database. With help by pgAdmin, we can easily let GeoServer access and process spatial database.

Here is example usage of GeoServer. We are going to show how to simply access a table in the existing spatial database. First, we need to setup a connection between GeoServer and our PostGIS database. We are going to connect to local PostGIS database namely ‘test’ by using admin access.

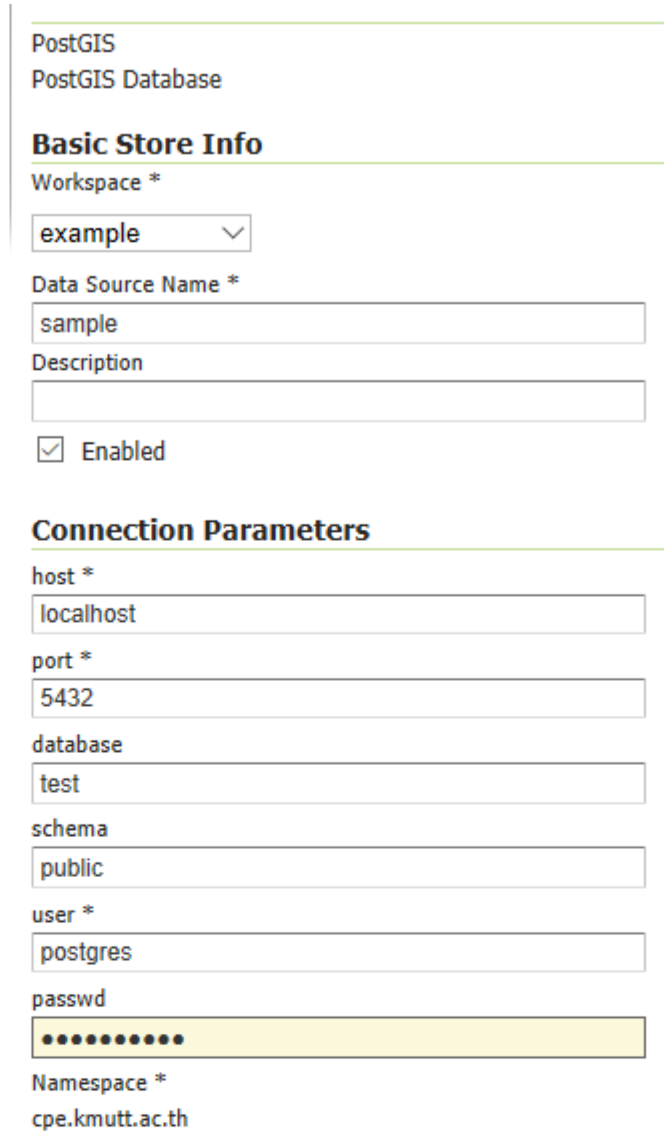

Figure 2.22 Establish connection to 'test' local database

The database table geom_test contains 3 different objects that are going to be represented as a layer, as discussed and illustrated in section 2.3.1. Before publishing database, GeoServer will let us to modify or specify some information such as coordinate system. Result is similar and have already shown in Figure 2.11. Furthermore, GeoServer can display multiple layers of data simultaneously by Layer Groups feature. Also, it allows moving data layers for ordering and specifying a rendering style. These are combination of Manhattan simple data layers display in a single window. For example, Figure 2.23 below consists of 4 data layers, background, points of interest, landmark and roads

GeoServer is a very powerful software tool, but it adds little complexity to the system architecture. In our project, it's used for displaying raster elevation layer data.

Figure 2.23 Layer Groups of Manhattan data layers [25]

2.3.5 GeoJSON

GeoJSON is a text-based format that is based on JSON and designed for encoding geographic data structures. It is simple, lightweight, straightforward and easy to use with Leaflet, and also supports geometry types and geometric objects.

- The geometry types: Point, LineString, Polygon, MultiPoint, MultiLineString and MultiPolygon. These below Figures, i.e., Figure 2.24, Figure 2.25, Figure 2.26, Figure 2.27, Figure 2.28 are example of how each geometry type is written in GeoJSON object.

In Figure 2.24, [100.0, 0.0] is a point coordinate which is [longitude(X), latitude for geographic coordinates(Y)].

```
{
    "type": "Point",
    "coordinates": [100.0, 0.0]
}
```

Figure 2.24 Example of writing Point in GeoJSON object [22]

Figure 2.25 shows that MultiPoint which is an array of point coordinates

```
{
    "type": "MultiPoint",
    "coordinates": [
        [100.0, 0.0], [101.0, 1.0]
    ]
}
```

Figure 2.25 Example of writing MultiPoint in GeoJSON object [22]

In Figure 2.26, there is an array of the coordinates that its first point is equal to its last point. The array is called *LinearRing*, and the array of LinearRing is polygon coordinates. If there are more than one set of LinearRings in the "coordinates" as shown in the Figure 2.27, the first LinearRing will be the exterior ring. The others are interior rings (holes).

```
{
    "type": "Polygon",
    "coordinates": [
        [ [100.0, 0.0], [101.0, 0.0], [101.0, 1.0], [100.0, 1.0], [100.0, 0.0] ]
    ]
}
```

Figure 2.26 Example of polygon in GeoJSON object [22]

```
{
    "type": "Polygon",
    "coordinates": [
        [ [100.0, 0.0], [101.0, 0.0], [101.0, 1.0], [100.0, 1.0], [100.0, 0.0] ],
        [ [100.2, 0.2], [100.8, 0.2], [100.8, 0.8], [100.2, 0.8], [100.2, 0.2] ]
    ]
}
```

Figure 2.27 Example of writing array of LinearRing in GeoJSON object [22]

- A geometric object is called *Feature.* It is a spatially bounded entity. List of features are contained in the object called *FeatureCollection.* The example of a simple Feature in GeoJSON is shown in the Figure 2.28,

```
var geojsonFeature = {
    "type": "Feature",
    "properties": {
        "name": "Coors Field",
        "amenity": "Baseball Stadium",
        "popupContent": "This is where the Rockies play!"
    },
    "geometry": {
        "type": "Point",
        "coordinates": [-104.99404, 39.75621]
    }
};
```

Figure 2.28 Example of writing Feature in GeoJSON [23]

2.3.6 Web Map Service

Web Map Service (WMS) is a protocol that serves georeferenced map images based on a request by using HTTP. When the requestor wants to process the geographic layers and area, it will make a request that defines the wanted geographic layers and area. Then the geographic information systems or one of the distributed geospatial databases will reply the request in an image format e.g., JPEG, PNG. The image forms can be displayed in web browser without any special process.

WMS requests are written in an XML document that specifies all the desired properties of the map such as location, image formats, desired data, etc. Writing software to create WMS request is quite complicated but there are many software tools, including Leaflet, that can do this for us.

2.3.8 jQuery

jQuery is a Javascript library that is able to do many things such as HTML document traversal and manipulation, event handling, Ajax, etc.

Figure 2.29 shows how to get the button element with the class “continue” and change its HTML to 'Next Step...'

```
$( "button.continue" ).html( "Next Step..." )
```

Figure 2.29 Example of DOM traversal and manipulation [26]

Figure 2.30 shows when the #banner-message element that is hidden with display:none in its CSS when any button in #button-container is clicked.

```
var hiddenBox = $( "#banner-message" );
$( "#button-container button" ).on( "click", function( event ) {
  hiddenBox.show();
});
```

Figure 2.30 Example of event handling [26]

Figure 2.31 shows how to call a local script on the server /api/getWeather with the query parameter zipcode=97201 and replace the element #weather-temp's html with the returned text.

```
$.ajax({
  url: "/api/getWeather",
  data: {
    zipcode: 97201
  },
  success: function( result ) {
    $( "#weather-temp" ).html( "<strong>" + result + "</strong> degrees" );
  }
});
```

Figure 2.31 Example of Ajax [26]

2.3.8 Related data

Our existing dataset is at the provincial level, Phitsanulok, collected from 2001 to 2017. It includes the patient's location, patient's ill date, patient's full recovered date, death date, age, gender and occupation. Our system is going to use additional factors; monthly rainfall, monthly average temperature, monthly relative humidity and land elevation.

Land elevation is acquired from ASTER V.2 dataset. ASTER was a Japanese-made sensor which attached to Terra satellite launched into earth orbit by NASA and gathering data since 2000. ASTER V.2 dataset was released to the public in 2011 and improved from previous version. It is the most complete mapping by covering 99% of earth surface with 30x30 meters resolution.

Moreover, boundary of Amphoe and Tambon are provided by KGEO and shown in Figure 2.32 with land elevation layer applied.

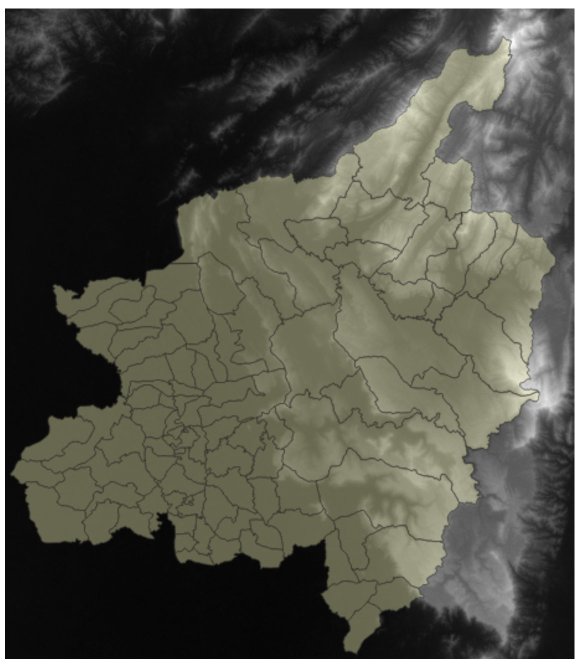

Figure 2.32 ASTER's land elevation of Phitsanulok by coloring and boundary of Amphoe and Tambon [26]

The individual patient data are already geocoded into Moo level. This is an 8 digits code. First 2 digits is Province ID. Next 2 digit is Amphoe ID, then Tambon ID and Moo

ID respectively. We acquired the 6 digits geocode list with Lat/Long coordinate from Digital Government Development Agency.

AD_LEVEL	TA_ID	TAMBON_E	AMPHOE_E	CHANGWAT_E	LAT	LONG
4	650103	Wat Chan	Mueang Phitsanulok	Phitsanulok	16.789	100.250
4	650103	Wat Chan	Mueang Phitsanulok	Phitsanulok	16.812	100.225
4	650114	Ban Khlong	Mueang Phitsanulok	Phitsanulok	16.830	100.239
4	650112	Chom Thong	Mueang Phitsanulok	Phitsanulok	16.902	100.215
4	650115	Phlai Chumph	Mueang Phitsanulok	Phitsanulok	16.854	100.226
4	650110	Pak Thok	Mueang Phitsanulok	Phitsanulok	16.905	100.259
4	650101	Nai Mueang	Mueang Phitsanulok	Phitsanulok	16.815	100.260
4	650503	Khok Salut	Bang Krathum	Phitsanulok	16.577	100.245
4	650116	Makham Sung	Mueang Phitsanulok	Phitsanulok	16.929	100.256
4	650120	Ngio Ngam	Mueang Phitsanulok	Phitsanulok	16.671	100.237
4	650504	Sanam Khli	Bang Krathum	Phitsanulok	16.547	100.256
4	650604	Matum	Phrom Phiram	Phitsanulok	16.936	100.205
4	650105	Tha Thong	Mueang Phitsanulok	Phitsanulok	16.798	100.197

Figure 2.33 Lat/Long coordinate of Tambon in Phitsanulok

For climatic data, Thailand Meteorological Department provide us historical data for 124 stations since 2001 to 2017 but the stations are only available in some part of Thailand. In prediction part, it's required the climatic data for all Tambons in Phitsanulok. We used Inverse-Distance Weight interpolation technique to generate data for area that stations are not available.

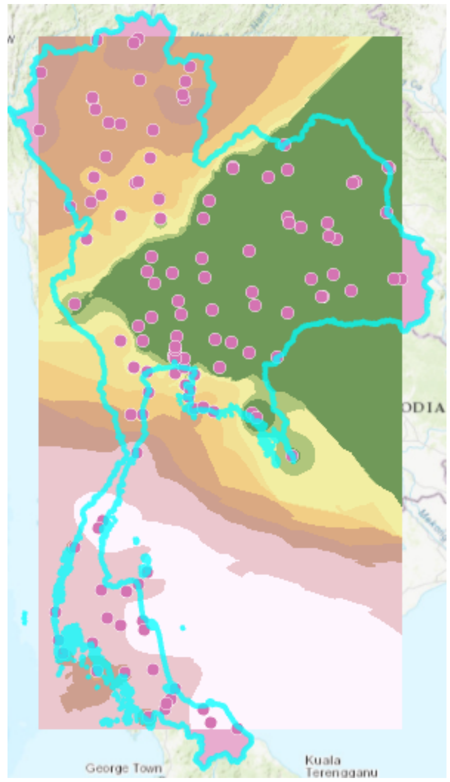

Figure 2.34 Interpolate Rainfall Data for Nationwide

To be more realistic and get better result in prediction part. It is important to know the population in all observation areas. The data are provided in Tambons-level since 2001 – 2017 by Department of Provincial Administration.

2.4 Related Research

In this section, we summarize some studies from the literature using GIS and spatial analysis for infectious disease.

2.4.1 Mapping of Dengue Outbreak Distribution Using Spatial Statistics and Geographical Information System [12]

This research was created because incidence of Dengue fever in Malaysia has been continuously increasing. Therefore, the researchers wanted to map the spread of the Dengue fever outbreaks there. In generating the dengue risk map, they used many spatial statistics techniques with GIS.

The first technique that was used was Moran's I. It is used to analyze clustering by measuring relationship among the values that is based on the location of point and the variation of the attributes of the cases in the location. Therefore, it is used to test whether the dengue cases are spatially correlated or not. Their null hypothesis is the Moran's I will show a random distribution of cases. After using this technique, they produced these values:

Global Moran's I Summary	
Variables	***values***
Moran's Index	0.072028
Expected Index	-0.001164
Variance	0.00159
Z-score	1.835761
p-value	0.066393

Figure 2.35 Global Moran's I Summary [12]

Next, these values were used in the pattern (Figure 2.35) to consider whether the null hypothesis is rejected or not.

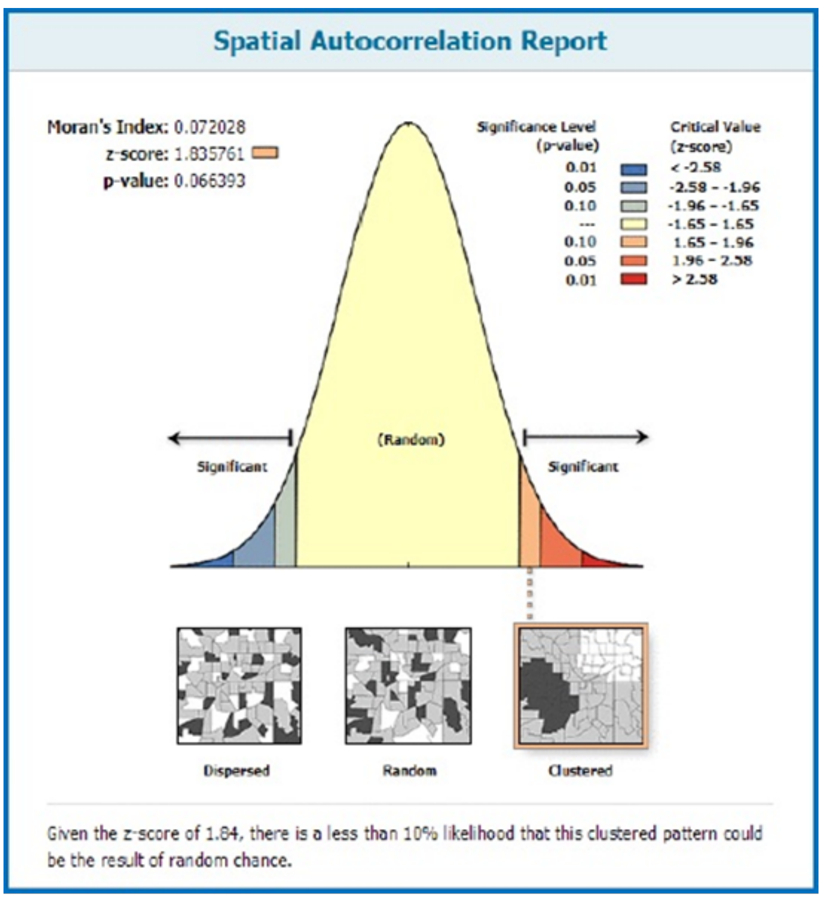

Figure 2.36 Spatial Autocorrelation Report [12]

After using the values and the pattern, they found that the data allowed them to reject their null hypothesis. This means each case is not randomly distributed. In other words, cases tended to cluster near each other. (The Z-score of 1.84 made them know that it is <10% likelihood that this clustered pattern could be the random chance result)

Second technique is Average Nearest Neighbor (ANN). It is used to analyze the dengue spatial distribution. This technique produces Nearest Neighbor Ratio, z-score, and p-value. The z-score is used to determine that they can reject the null hypothesis or not. After comparison the result with the pattern in Figure 2.36, they found that the Z-score of 114296.938491, it is < 1% likelihood that the dispersed pattern could be the random chance result. In other word, it almost has no probability that the pattern could be the random chance result. Therefore, they also reject the null hypothesis.

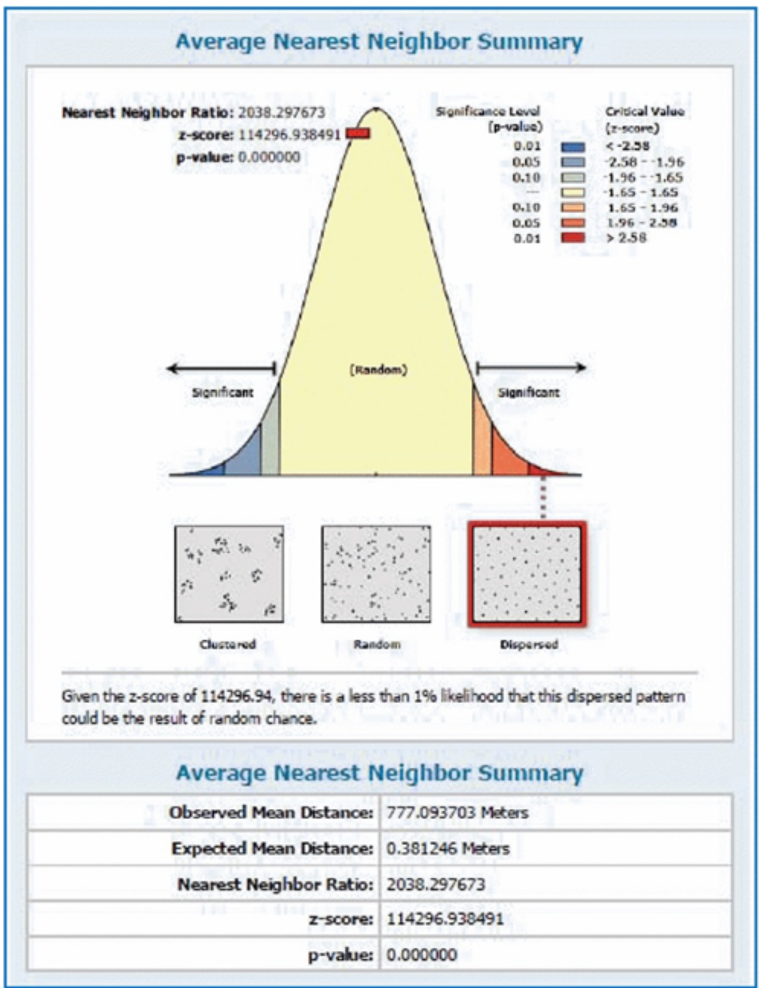

Figure 2.37 Average Nearest Neighbour Summary. [12]

The last technique that they used is Kernel density estimation (KDE). This is a good way to locate a hot spot. For this technique, they can detect the dengue density variation in the area. After using this technique, they found that the worst affected areas were Ampang, Damansara, Kapar, Kajang, Klang, Semenyih, Sungai Buloh and Petaling.

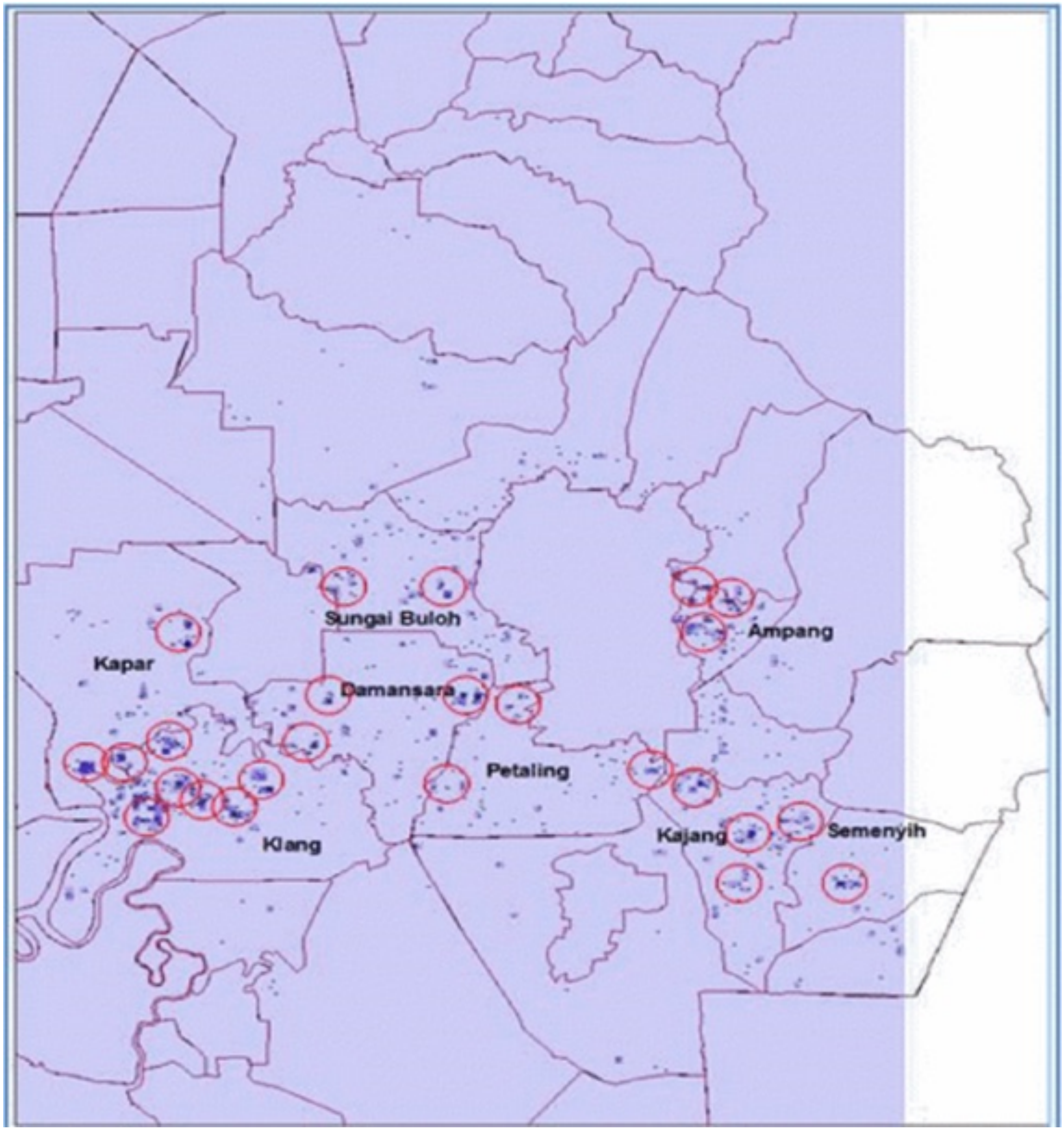

Figure 2.38 Kernel density estimation Summary [12]

From study this research, we think that the third technique or Kernel density estimation (KDE) is most useful in a disease mapping context because it can tell where the exact affected area is.

2.4.2 Dengue cases distribution based on land surface temperature and elevation [10]

Dengue fever is a major disease challenge because it is spreading worldwide. Also, climate change is increasing the areas susceptible to dengue. The case rate of dengue is high in Malaysia. Therefore, the researchers wanted to prevent the disease by controlling its outbreaks. In this study, two methods were used. Figure 2.38 shows the Land Surface Temperature (LST) which is the first method.

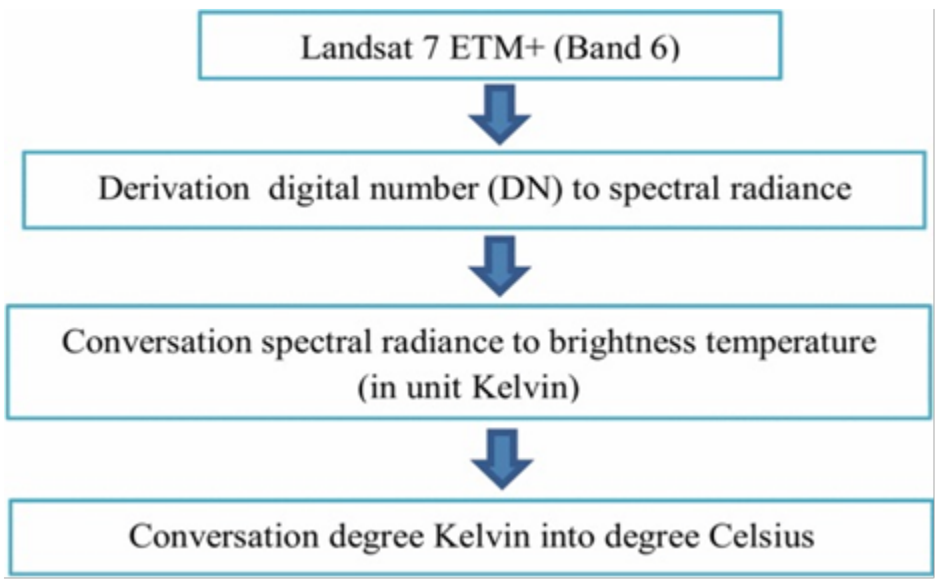

Figure 2.39 The steps of Land Surface Temperature (LST) [10]

They began by using the thermal band 6 from Landsat 7 ETM+ for identifying land surface temperature. The second step is to calculate the spectral radiance and temperature conversion. For the third, after getting the radiance values from the second step, the values are converted to brightness temperature. The last step that they did is to convert degree Kelvin into degree Celsius.

Figure 2.39 is the result of showing temperature zones on the map where they had plotted dengue cases. It shows that dengue cases were distributed in the range of 12°C to >30°C.

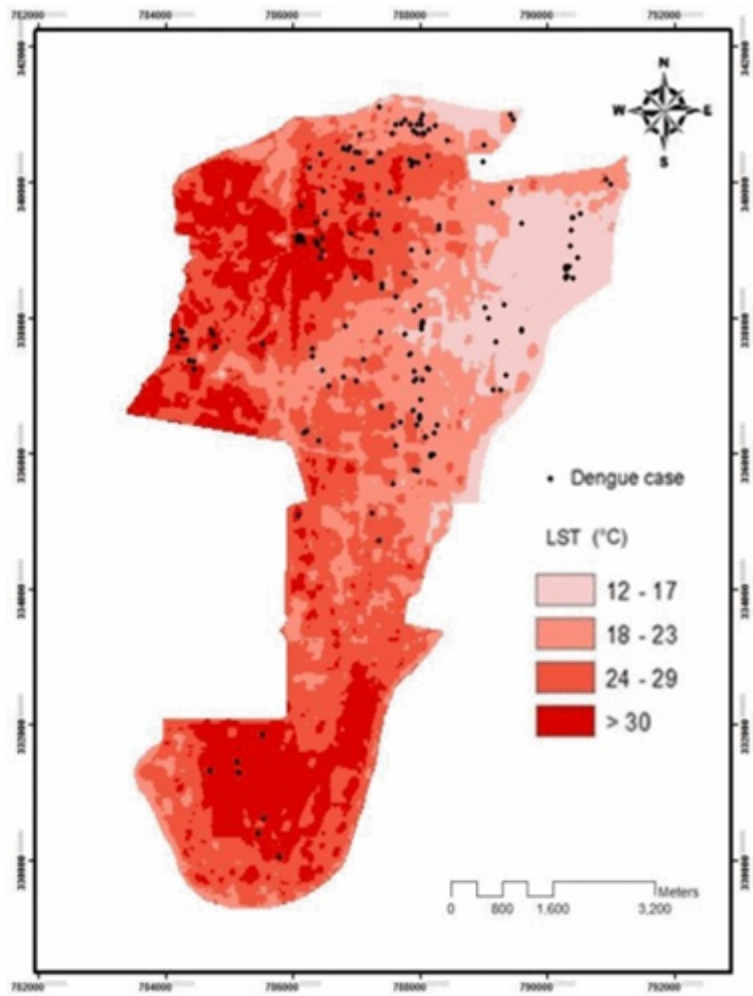

Figure 2.40 The result of Land Surface Temperature (LST) [10]

The second technique they used is the elevation method. Elevation was divided 0 to 150 meters (above mean sea level or AMSL) into five classes. The elevation was separated by using different colors. Then they overlaid the dengue cases on the elevation map. The result shows that the most of dengue's incidence was distributed at low elevation. The result can be concluded that each community that lives in low and warmer areas are more susceptible to dengue transmission

From study this research, we found that method or techniques that are appropriate with the selected disease is up to infectious factors.

2.4.3 Tuberculosis Disease Mapping with Poisson-Gamma Model in Malaysia [13]

The aim of this research is to estimate *the relative risk* for spread of Tuberculosis disease because this disease is a widespread disease and a factor that causes many deaths. The researchers estimated the relative risk by using Poisson-gamma model.

The Poisson-gamma model is one of Bayesian methods. Bayesian is a logic branch that is used to make decision, to infer statistics that deals with probability inference. The probability inference is to use the knowledge of prior events to predict future events. Therefore, the researchers used WinBUGS software to estimate the relative risk because the WinBUGS software is used to implement Bayesian inference on statistical problems. The researcher used the software by inputting number of new infections and variance in order to calculate the expected number of new infections.

After using the software, they got this graph as shown in Figure 2.40.

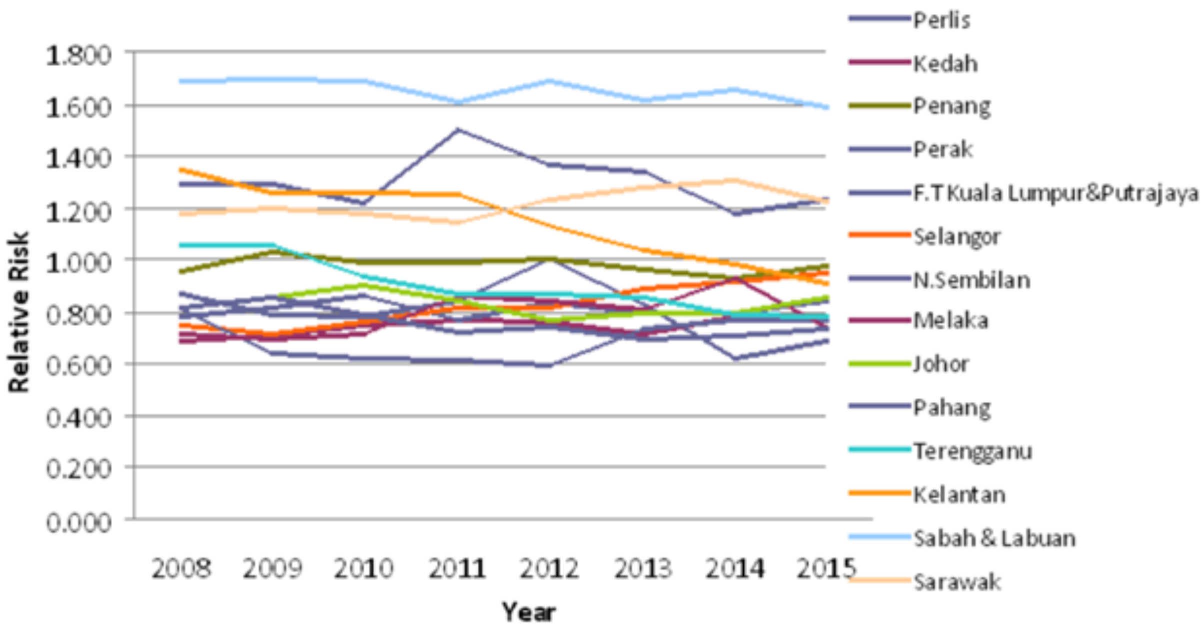

Figure 2.41 Time series plots of the relative risk estimation base on the Poisson-gamma model for different states in Malaysia [13]

If a relative risk is close to 1: There is no significant different probability between the people that are affected by Tuberculosis in a region and within the all populations.

If a relative risk > 1: By comparison with all populations, people in the region are at greater risk of contracting this disease.

Next, they used the graph (Figure 2.42) to build the Tuberculosis disease map as shown in the picture below.

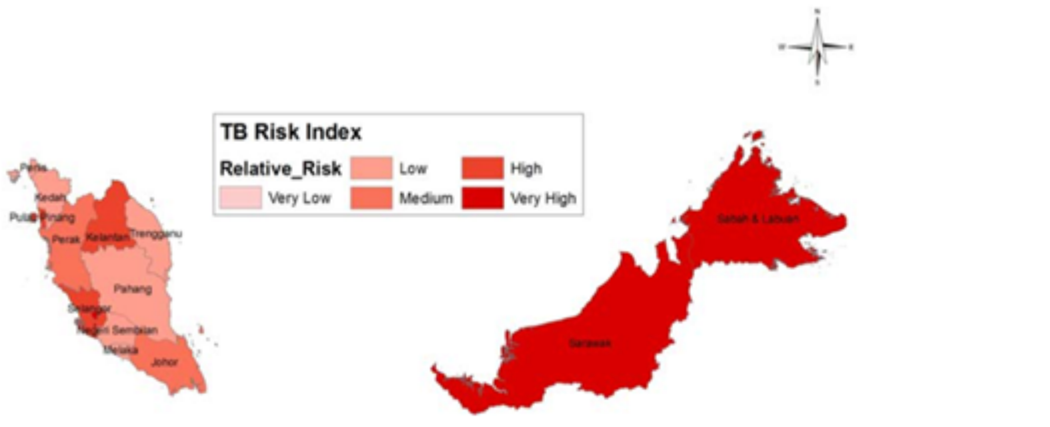

Figure 2.42 Disease map of relative risk estimation based on Poisson-gamma model for year 2015 [13]

After studying this research, to estimate the relative risk is easy way to build a disease map, especially in the case that the disease is spread in the air.

2.4.4 Spatio-Temporal Diffusion Pattern and Hotspot Detection of Dengue in Chachoengsao Province, Thailand [20]

The goal of this research aims to investigate dengue spatio-temporal diffusion patterns in Chachoengsao using dengue cases data and climatic data during 1999-2007 by the help of GIS software.

For temporal analysis, they analyzed dengue cases for temporal patterns. The climatic data was also considered. Pearson Correlation coefficient was used to find relationship between incidence cases and climatic condition. Their model was developed to find correlation between dengue cases for a specific month *t* and monthly mean of rainfall, temperature and relative humidity for months *t* and *t-1*, to evaluate the climatic condition effects on the vector. The Pearson coefficients are shown in the Table 2.2 As the result, number of incidences significantly correlated to these factors.

Table 2.2 Pearson correlation between climate factor and dengue cases in time series, 1999–2007 [20]

Year	RF		TEMP		RH	
	t	t−1	t	t−1	t	t−1
1999	0.460	0.512	−0.042	0.262	0.631	0.610
2000	0.676	0.839	0.258	0.677	0.647	0.777
2001	0.548	0.516	0.380	0.342	0.539	0.506
2002	0.472	0.346	0.608	0.768	0.158	0.252
2003	0.488	0.296	0.538	0.316	0.544	0.319
2004	0.872	0.916	0.177	0.277	0.682	0.777
2005	0.729	0.481	0.525	0.524	0.840	0.551
2006	0.528	0.836	−0.026	0.325	0.580	0.825
2007	0.446	0.726	0.166	0.487	0.461	0.740

In conclusion, dengue incidence case highly correlated to rainfall (RF), temperature (TEMP) and relative humidity (RH) one month before the incidences.

For spatial analysis, mapping incidence for spatial analysis need to use Empirical Bayes Smoothing (EBS) to avoid false statistical result. The difference in population could influence the results differently. Areas with less populations are more sensitive to statistical noise and areas with more populations are less. Then areas with small populations, are usually give spuriously high rate of infected rate. EBS was used to smooth the incidence rate and transform into dengue fever morbidity rate (DFMR). The another essential for mapping visualization is standard deviational ellipsis. It shows how the features distributes in which direction. This method is commonly used for measuring the spreading trend of a set of points. SDE was applied for comparing the position on incidence case and spatial extent. And also, Global Moran's I statistic was used to find the spreading pattern (clustered, dispersed, random) among the villages. The mapping showed that the incidence occurred throughout the province. Moreover, the SDE also show that the global position and directional trend in each year are similar pattern.

For space-time analysis, they used GIS software that provided the ability to analyze space-time data. Onset date of symptoms was analyzed to visualize into space-time animation to investigate the spreading pattern of outbreak from May to September 2007

For hotspot detection, Local Indicators of Spatial Association (LISA) was used to determine the location and pattern of hotspots. The local Getis-Ord $\boxed{GI^*}$ was carried out for determining the spatial dependence. It is well-known techniques for identifying locations of statistically significant spatial clusters of focus information.

For example, the temporal analysis shows that there are 551 dengue cases occurred from beginning of May until the end of September 2007. Almost a quarter of all cases occurred from 2nd to 4th week of June. Monthly number of confirmed cases and average rainfall during 1999-2007 are shown in Figure 2.43.

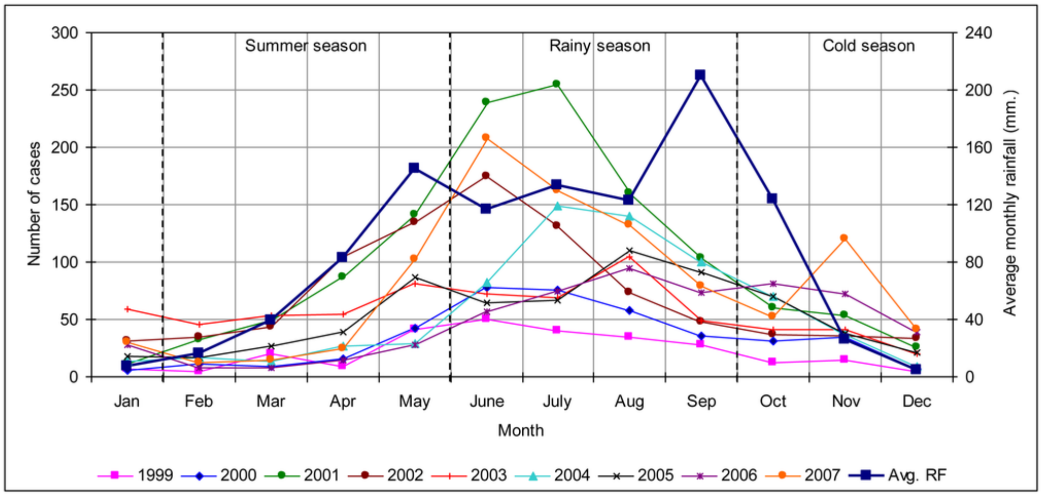

Figure 2.43 Number of dengue cases and average rainfall on monthly basis in the years 1999–2007 [20]

The temporal distribution showed the similar trend every year with the highest incidence in rainy season. Outbreaks usually happened in early part of the rainy season which has higher humidity than average. Moreover, with the increasing of rainfall, temperature and relative humidity in May, an outbreak is usually reported in the next month. In September, rainfall and relative humidity are at peak and number of incidences is decreased.

This study shows valuable information about the spatial diffusion pattern and also indicates that climate is associated to the incidence. The temporal analysis points out rainfall and high humidity of one month before indicating high correlation with the incidence.

Chapter 3

Design and Methodology

3.1 Software Requirements Lists

By gathering information from people who work in the ministry of public health in Phitsanulok about problem of surveillance, they said that the software that they are using has a number of problems.

These are their problems:

- When they are doing fieldwork, they cannot define location of cases because of network problem.
- Information on their map is not enough to be used to identify the location of case. Therefore, they need to create their own map, e.g., geo-social mapping.
- When they want to surveil and predict, they need to calculate by themselves in Excel program. It is quite consuming time and complicated.

3.2 Feature Lists

- Store spatial data, incidence cases information, climatic data.
- Display disease incidence, filtered by various factors such as time, location, age of patient (maybe), morbidity (i.e. did patient die) etc.
- Analyze the diffusion pattern of disease, hotspot detection, Getis-Ord GI*
- Predict spreading direction and indicate the possibility of specific location to be infected area.

Note that this project will not address issues related to fieldwork, although a future senior project might do so.

3.3 System Architecture

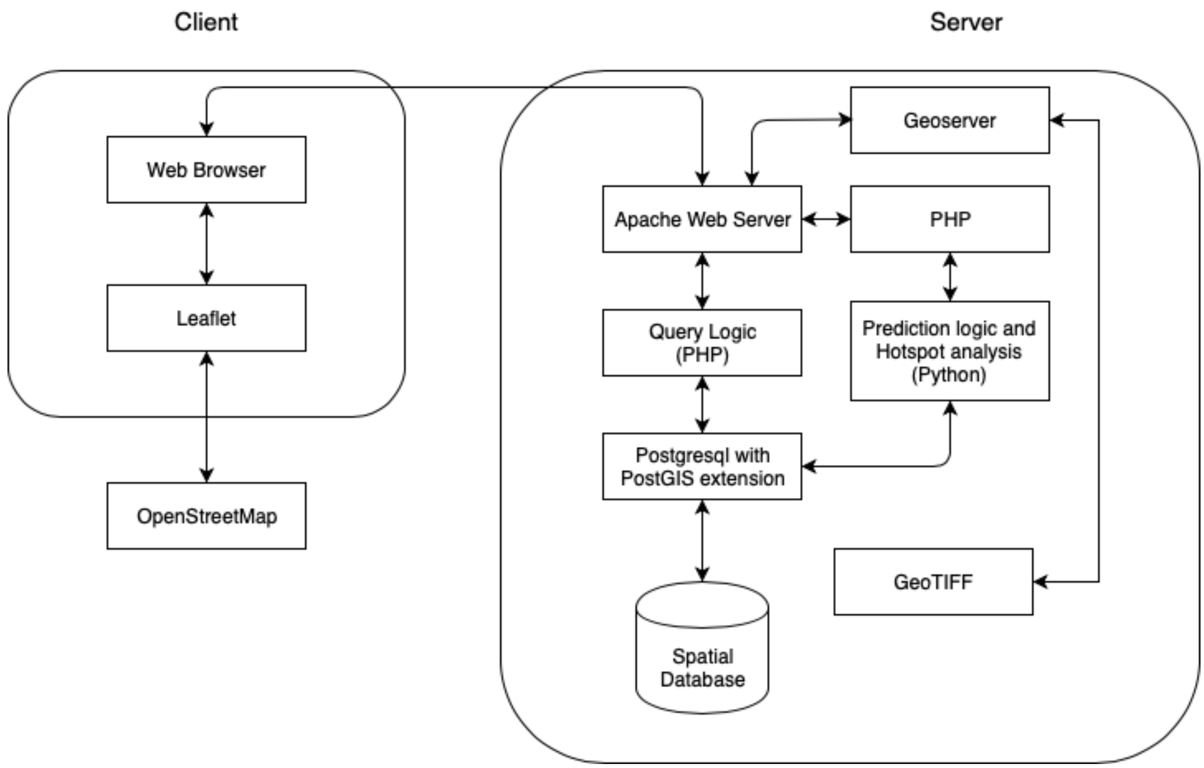

Figure 3.1 System Architecture

The architecture of our system consists of server-side and client-side. On server side, it handles on request queries from client-side and also process the prediction and hotspot analysis module written in Python. It equips with PostGIS module that store spatial database and shapefile converter. On client-side, browser use Leaflet library to create base map that data were provided by OSM server. Leaflet also add geographic objects into map. Moreover, GeoServer serves raster data for web-browser to display land elevation.

3.3.1 Web Browser Module

This module is the part that interacts with user, the user interface will be displayed on this module. On this module, the user can customize, request data, etc.

3.3.2 Leaflet Module

This module uses map from OpenStreetMap, and also manages objects on the map, e.g. marker, popup, polygon and layer.

3.3.3 Query Module

This module can query data by request from the PostGIS DB module

3.3.4 Prediction Module

This module will predict incident cases by following what user customize on the web browser.

3.3.5 PostGIS DB Module

This module will be used to store all data that is used in this system.

3.4 Use Case Analysis

3.4.1 Use Case Diagram

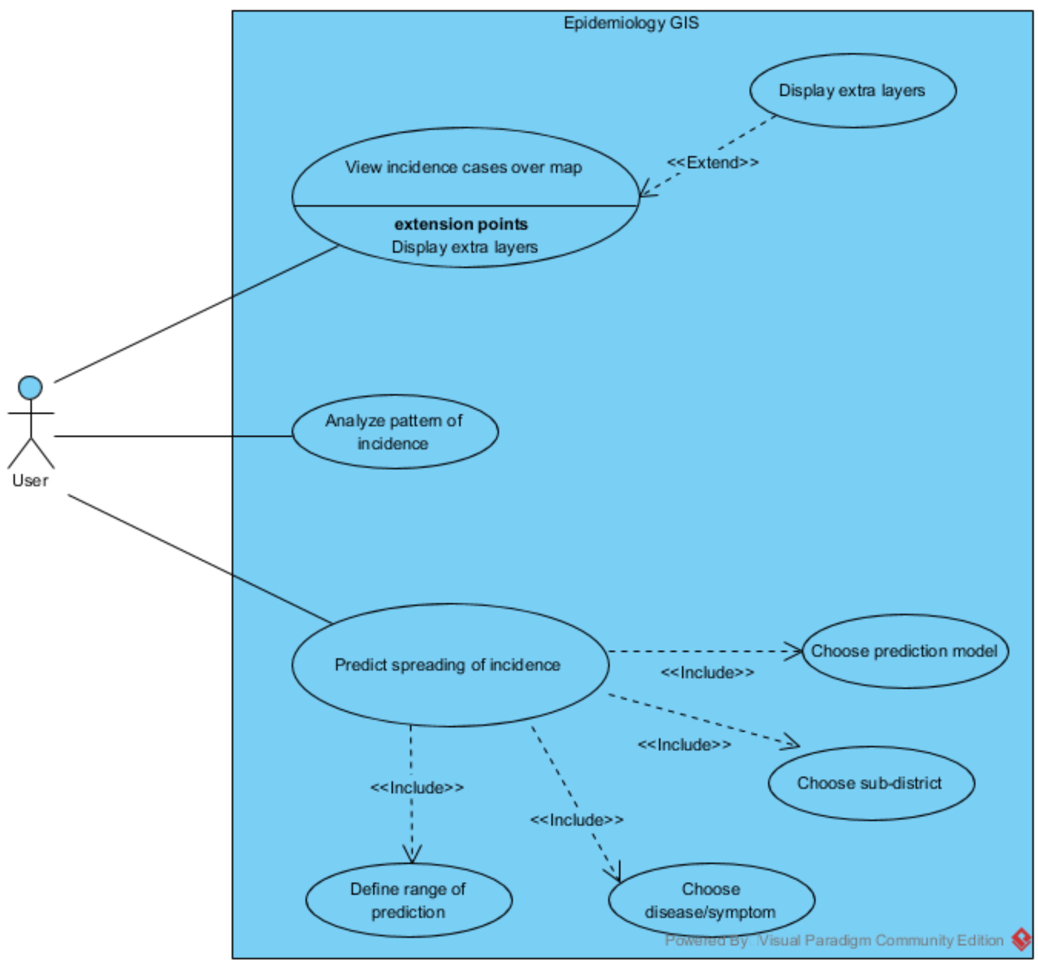

Figure 3.2 Use cases for Epidemiology GIS

3.4.2 Sub Use Case Diagram

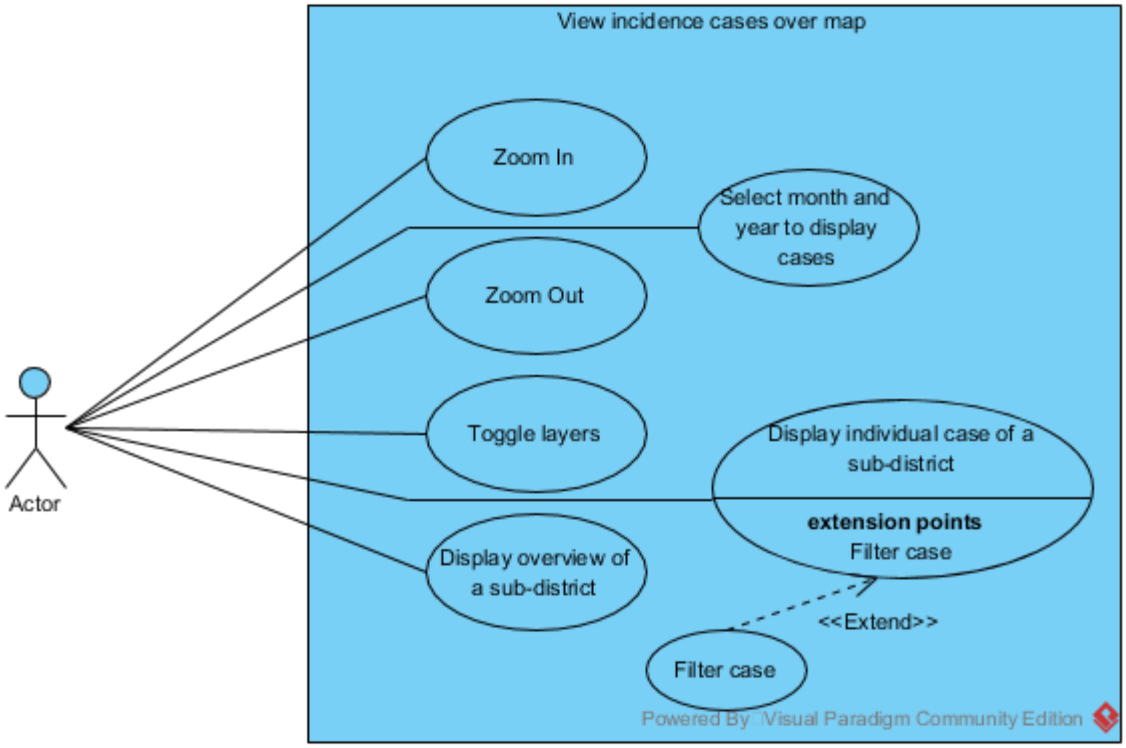

Figure 3.3 Use cases for viewing incidence case on map

3.4.3 Use Case Narrative

Scenario 1: Select month and year to display incidence cases

Goal: Display incidence cases over the map during a specific range of months.

Precondition: Base map is displayed on mapping system.

Main success scenario

1. User can choose a period of time (month and year) to display cases by using drop down menu to define start date and end date.
2. Some area may have no incidence at selected period, the markers are not pinned in that area.
3. If there is no more data available after specific time, alert message is shown, and no marker are pinned.

Scenario 2: Zoom in the map

Goal: Enlarge some detail of map.

Precondition: Base map is displayed on mapping system.

Main success scenario

1. User clicks '+' button or scroll up mouse.
2. Center of map is enlarged into smaller scale.
3. Map generate more detail by using OSM data.
4. Incidence case markers still maintain in same coordinate.
5. If map can't be further enlarged. Maximum zoom in level is reached.

Scenario 3: Zoom out the map

Goal: Inspect overview of map or nearby area.

Precondition: Zoom in map and incidence case markers are displayed on mapping system.

Main success scenario

1. User clicks '-' button or scroll down mouse.
2. Center of map is zoom out into larger scale.
3. Map removes some detail out.
4. Incidence case markers maintain in same coordinate.
5. If map can't be further zoom out. Maximum zoom out level is reached.

Scenario 4: Toggle layers

Goal: Allow users to compare incidence case with other factor layers

Precondition: Base map is displayed on mapping system.

Main success scenario

1. User clicks at a button represents a specific layer.
2. If user activate a layer, information of layer displays on top of map with transparency. The system will also display a legend.
3. If user activate another new layer, information of new layer displays on top and maintains the previous layer.

4. If user deactivates a layer, information of that layer is removed from display map.

Scenario 5: Display overview incidence case in each Tambon

Goal: Inspect overview of incidence cases.

Precondition: Base map and incidence case markers are displayed on mapping system.

Main success scenario

1. User clicks on any existing markers on the map.
2. Marker is transformed into popup contains some information such as population, largest patient age group, proportion of patient gender.

Scenario 6: Display individual cases with filter in each Tambon

Goal: Inspect overview and individual incident cases.

Precondition: User clicks at a marker on a map and popup is shown

Main success scenario

1. User click hyperlinks in the popup for more information.
2. Individual cases are displayed in a separate window.
3. It is possible to filter some information such as filter by age group, gender and disease. Select drop down menu for "filtering by".
4. When a filter is selected, the browser displays a new result with filter applied.

Scenario 7: Analyze the pattern of incidence

Goal: Find hotspot of disease at specific period.

Precondition: Select time period of display map then incidence case markers are displayed on mapping system.

Main success scenario

1. User click on Analyze button on side bar.
2. User can choose a method to calculate hotspots.
3. User click on show hotspot to apply hotspot calculation on map layer.

4. Tambon area with red polygon are determined as hot spot. Blue polygon represents as cold spot
5. If user minimizes the side bar, the hotspot layer still displays on map.
6. If user change time period on top display screen while hotspot layer is activated. Hotspot layer still activated but new hotspot processing is requested. Calculation is based on new time frame.
7. User can deactivate hotspot layer by going to the Analyze tab again then click clear button to deactivate it.

Scenario 8: Predict spreading of incidence

Goal: Forecast the incidence of specific disease at specific time range.

Precondition: Sided is loaded properly.

Main success scenario

1. User click on prediction button on sided bar.
2. User is required to choose forecasting incidence model.
3. User is required to define start month and start year. This date should be present month-year or future month-year.
4. User is required to specify number of months to forecast for.
5. User is required to choose disease as target disease (DF, DHF and DSS).
6. User can adjust parameters for the selected model.
7. If user leave any field in list 2-5 undefined, the system will not do forecasting
8. If user filled in fields properly, result is processed and shown as markers on map.
9. If user request a new prediction result by clicking predict button, the previous result should be removed.

3.5 ER Diagram and Database Schema

Figure 3.16 shows that in the database, there are 6 tables: lu_disease, lu_marital, lu_career, incidence, lu_race and lu_tambon. The tables that start with "lu_" are lookup tables whose values never change. They are used to assign string values to different codes in the incidence table.

In the ER diagram below, columns with data type "geometry" hold spatial data in PostGIS binary format, and "N" letter with white color means the column can be null. These columns can be used for spatial queries and spatial operations.

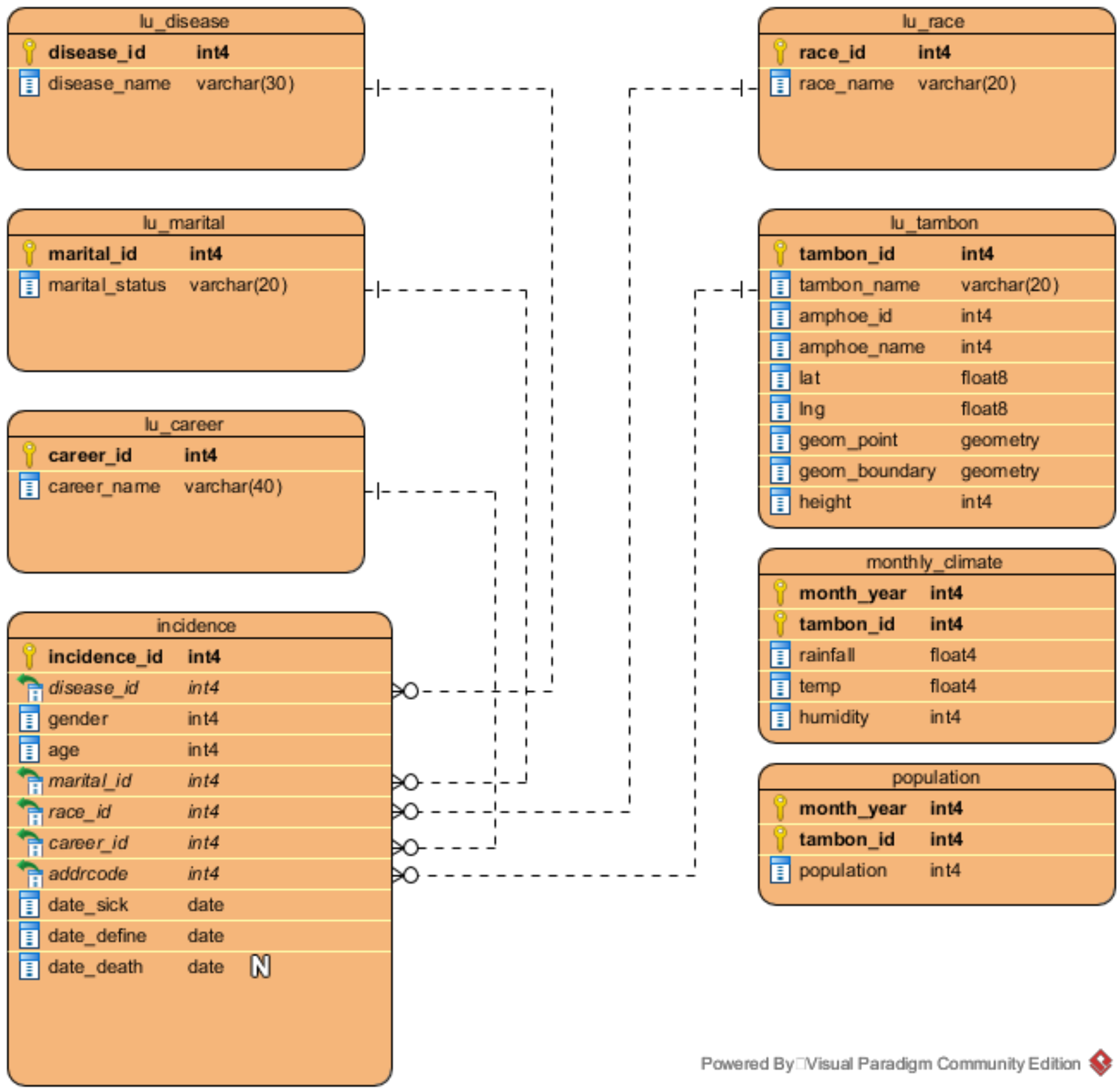

Figure 3.4 ER diagram

3.5.1 lu_disease Table

There are 2 columns in this table. This table used to look up for disease name of each id.

Table 3.1 Structure of lu_disease table

Field	Type
disease_id	Integer (4)
disease_name	Varchar (30)

3.5.2 lu_marital Table

There are 2 columns in this table. This table used to look up for marital status of each id.

Table 3.2 Structure of lu_marital table

Field	Type
marital_id	Integer (4)
marital_status	Varchar (20)

3.5.3 lu_career Table

There are 2 columns in this table. This table used to look up for career name of each id.

Table 3.3 Structure of lu_career table

Field	Type
career_id	Integer (4)
career_name	Varchar (40)

3.5.4 lu_race Table

There are 2 columns in this table. This table used to look up for race name of each id.

Table 3.4 Structure of lu_race table

Field	Type
race_id	Integer (4)

race_name	Varchar (20)

3.5.5 lu_tambon Table

There are 10 columns in this table. This table used to look up for tambon's information of each id.

Table 3.5 Structure of lu_tambon table

Field	Type
tambon_id	Integer (4)
tambon_name	Varchar (20)
amphoe_id	Integer (4)
amphoe_name	Varchar (20)
lat	Float (8)
lng	Float (8)
geom_point	Geometry
geom_boundary	Geometry
height	Integer(4)

3.5.6 incidence Table

There are 11 columns in this table. This table contains details of each incident cases, and it also links to every table.

Table 3.6 Structure of incidence table

Field	Type
incidence_id	Integer (4)
disease_id	Integer (4)
gender	Integer (4)
age	Integer (4)
marital_id	Integer (4)

race_id	Integer (4)
career_id	Integer (4)
addrcode	Integer (4)
date_sick	Date
date_define	Date
date_death	Boolean

3.5.7 monthly_climate Table

There are 6 columns in this table. This table contains information of climatic data including tambon id, temperature, rainfall and humidity.

Table 3.7 Structure of monthly_climate table

Field	Type
month_year	Integer (4)
tambon_id	Integer (4)
rainfall	Integer (4)
temp	Integer (4)
humidity	Integer (4)

3.5.8 poplation Table

There are 3 columns in this table. This table contains information of population including tambon id and population.

Table 3.7 Structure of monthly_climate table

Field	Type
month_year	Integer (4)
tambon_id	Integer (4)
population	Integer (4)

Chapter 4

Results and Discussion

4.1 Creating Database

In our system, there are 8 database tables that we have already created, except monthly_climate. Patient table is shown below in Table 4.1. This table was created by partially extraction from incidence cases that provided by MOPH Phitsanulok office. We don't bring whole table due to some columns are not informative for our project and values are missing. Death date is somewhat informative but seem to incorrect. We think change to Boolean value is appropriate way.

Table 4.1 Stored data in patient table

	id [PK] inte	disease_id smallint	gender smallint	age smalli	marital smallin	race smalli	career smallint	addrcode character vary	tambon_id character vary	date_sick date	date_define date	date_death date
1	1	2	2	3	1	8	11	65010101	650101	2001-09-14	2001-09-17	[null]
2	2	2	1	24	2	1	11	65090203	650902	2001-07-14	2001-07-17	[null]

We also already create all lookup tables for columns. For example, this table shown career look up table.

Table 4.2 Look up table for career columns

	id [PK] integer	career_name character varying (40)
1	1	Farmer
2	2	GovtOfficer
3	3	Worker/Labor
4	4	Merchant

There are others source of spatial data that stored in our spatial database, including sub-district boundary which imported by PostGIS Shapefile Loader, sub-district lat/long and land elevation. This table show information of sub-district lat/long table. Note that geom data type shown in table below is not human-readable format. It stores lat/long coordinate as spatial object.

Table 4.3 Look up table for tambon location as latitude and longitude

	tambon_id integer	tambon_name character varying (20)	amphoe_id integer	amphoe_name character varying (20)	lat double pr	long double pre	geom geometry
1	650101	NaiMueang	6501	MueangPhitsanulok	16.815	100.26	0101000020E6100000713D0...
2	650102	WangNamKhu	6501	MueangPhitsanulok	16.652	100.269	0101000020E6100000F0A7C...
3	650103	WatChan	6501	MueangPhitsanulok	16.8005	100.2375	0101000020E610000033333...

4.2 Implementation

There are many source files in this system. They are listed in the table 4.4

Table 4.4 lists of source files

Source Files	Description
homepage.php	The page is interactive with users. The users can see the incidence, incidence prediction, hotspot analysis and layers
tambonPolygon.php	It is used to retrieve and prepare data to display as polygon on the map (in the homepage.php)
queryRainfall.php	It is used to retrieve and prepare data to display as rainfall layer on the map (in the homepage.php)
queryHumidity.php	It is used to retrieve and prepare data to display as humidity layer on the map (in the homepage.php)
queryMarker.php	It is used to query and prepare the incidence data to be shown on the map in the homepage.php
gistar.py	It is a python file that calculate hotspot.
findHotspot.php	It is used to call gistar.py, get the hotspot result, and prepare the data to be display on the map (in the homepage.php)
reg.py	It is a python file that calculate incident prediction.
findReg.php	It is used to call reg.py, get the incident prediction result, and prepare the data to be display on the map (in the homepage.php)

This project is a map software with map, incidence cases, layers, hotspot analysis and incidence prediction.

4.2.1 Map part

The first important part is the map part which is part of adding map to the software. When the homepage.php is called by a browser. The "startPage" function in homepage.php will be called.

```
window.onload = setTimeout("startPage()",100);
```

Figure 4.1 Calling startPage()

The startPage() is about adding many parts to the map. Figure 4.2 shows the parts of adding map by using leaflet which is JavaScript library to initialize OpenStreetMap on web browser as show in the Figure 4.2.

```
map.setView([dafaultLat, dafaultLng],defaultZoom);        //Phitsanulok at center of the map

L.tileLayer('http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
    maxZoom: 18,
    attribution: 'Map data &copy; OpenStreetMap contributors'
}).addTo(map);
```

Figure 4.2 Initialize OpenStreetMap on web browser

4.2.2 Incidence cases

To show incidence cases by using marker, in homepage.php, when user selects time range, another PHP file called queryMarker.php will be used with the selected time range as shown in the figure 4.3.

```
$.ajax({
    type: "POST",
    url: 'queryMarker.php',
    data: {
          startDay: "01",
          endDay:endDay,
          startMonth: startMonth,
          startYear: startYear,
          endMonth: endMonth,
          endYear: endYear},
    success: initializePage
})
```

Figure 4.3 How we send parameter and call the queryMarker.php file

In figure 4.4, the queryMarker.php will connect to the database and query data by using the sent parameter.

```
$startDay = $_POST['startDay'];
$endDay = $_POST['endDay'];
$startMonth = $_POST['startMonth'];
$startYear = $_POST['startYear'];
$endMonth = $_POST['endMonth'];
$endYear = $_POST['endYear'];

$start = "DATE '".$startYear."-".$startMonth."-".$startDay."'";
$end = "DATE '".$endYear."-".$endMonth."-".$endDay."'";

$sql1 = "SELECT a.*, b.numberOfPatient, b.addrcode
FROM lu_tambon a
FULL JOIN"."("."SELECT count(p.id) as numberOfPatient, p.addrcode
from patient p";
$condition = " WHERE date_sick >= ".$start." AND date_sick <= ".$end;
$sql2= " group by p.addrcode) b ON a.id = b.addrcode";

$allSql= $sql1.$condition.$sql2;

$data = array(); //setting up an empty PHP array for the data to go into

if($result = pg_query($db, $allSql)) {
    while ($row = pg_fetch_assoc($result))
    {
      $data[] = $row;
    }
}
```

Figure 4.4 How to query data

Before returning the data to the homepage.php file, the data need to re-format to GeoJSon format as shown in the figure 4.5. If it's not in the GeoJSon format, it will not be able to be displayed by using leaflet library.

```php
$jsonData =json_encode($data);
$original_data = json_decode($jsonData, true);
$features = array();
foreach($original_data as $key => $value) {
    if($value['numberofpatient']){  //not show the village with null patient
        $features[] = array(
            'type' => 'Feature',
            'properties' => array('tambon_id' => $value['id'],'tambon_name' => $value['tambon_name'],
                            'amphoe_id' => $value['amphoe_id'],'amphoe_name' => $value['amphoe_name'],
                            'numberofpatient' => $value['numberofpatient']),
            'geometry' => array(
                'type' => 'Point',
                'coordinates' => array(
                    $value['long'],
                    $value['lat']
                ),
            ),
        );
  }
}

$new_data = array(
    'type' => 'FeatureCollection',
    'features' => $features,
);


$final_data = json_encode($new_data, JSON_PRETTY_PRINT);
echo $final_data;
```

Figure 4.5 How to re-format the data to GeoJSon format

After the data is returned to the homepage.php file, we show markers on the map by using leaflet javascript library format as shown in the figure 4.6.

```javascript
var MarkerIcon = L.icon({
  iconUrl: 'pin.png',
  iconSize: [32, 32],
  iconAnchor: [16, 37],
  popupAnchor: [0, -28]
});

var markerLayer = L.geoJson(JSON.parse(data), {
  pointToLayer: function (feature, latlng) {
    return L.marker(latlng, {icon: MarkerIcon});
  }, onEachFeature: onEachFeatureNormal
}).addTo(map);
```

Figure 4.6 How to use the return data from the queryMarker.php file

4.2.3 Layers

There are four layers to be displayed on the software.

- Sub-district boundary layer

To shows boundary of each sub-district on the map, a PHP file that is used for querying will be included in homepage.php file as shown in the figure 4.7.

```
<?php include 'tambonPolygon.php';?>
var subPolygon = <?php echo $json; ?>;
```

Figure 4.7 Example of include PHP file

The figure 4.8 is how to query the polygon from the created database.

```
$data = array(); //setting up an empty PHP array for the data to go into
if($result = pg_query($db,"SELECT gid, ST_AsGeoJSON(geom) FROM tambon_boundary")) {
   while ($row = pg_fetch_assoc($result)) {
     $data[] = $row;
  }
}
```

Figure 4.8 Show querying sub-district boundary in tambonPolygon.php file

Before using the queried data in the homepage.php file, the data needs to be in GeoJSon format. The figure 4.9 shows how to re-format the data to GeoJSon format.

```
$features = array();
foreach($data as $key => $value) {
    $features[] = array(
      'type' => 'Feature',
      'properties' => array('gid' => $value['gid']),
      'geometry' => $value['st_asgeojson']
    );
}

$new_data = array(
    'type' => 'FeatureCollection',
    'features' => $features,
);
```

Figure 4.9 How to re-format the data to GeoJSon format

Then, the data is ready to be displayed as Sub-district boundary layer by using the leaflet library as shown in the figure 4.10

```
function subLayer(){
  console.log("subLayer");
  var subBox = document.getElementById("sub");
  if (subBox.checked == true){
    //Add Polygon
    var polygon = L.geoJSON(tambomPolygon).addTo(map);
    polygon.setStyle({fillColor: '#333', color: '#333', weight: 1,
    opacity: 0.3, fillOpacity: 0.3});
  }
}
```

Figure 4.10 Example of using the result from including the PHP file.

- Rainfall layer

  When user select time to display the layer, then the queryRainfall.php will be called and prepare the data in GeoJSon format to be displayed like the sub-district boundary layer.

- Humidity layer

  It is similar to rainfall layer, but the queryHumidity.php file will be called instead.

- land elevation layer

  For land elevation layer, it is adding GeoTIFF file to map by using leaflet javascript library and GeoServer as show in the figure 4.11.

```
function landLayer(){
  var landBox = document.getElementById("land");
  if (landBox.checked == true){
    //add land elevation layer
    var wmsLayer = L.tileLayer.wms('http://localhost:8081/geoserver/EpidemGIS/wms?', {
        layers: 'DEM',
        format: 'image/png',
        transparent: true,
        version: '1.1.1',
        opacity: 0.5
    }).addTo(map);
  }
}
```

Figure 4.11 How we add land elevation to the map

4.2.4 Hotspot Analysis

In hotspot analysis part, user needs to select threshold, start date and end date. After the selection, the parameter will be sent to a PHP file called findHotspot.php as shown in the figure 4.12.

```
$.ajax({
    type: "POST",
    url: 'findHotspot.php',
    data: {
          selectedThreshold: selectedThreshold,
          startDate: startDate2,
          endDate: endDate2},
    success: displayHotspot
})
```

Figure 4.12 How we send parameter and call the findHotspot.php file

In the figure 4.13, a python file called gistar.py will be executed with the sent parameter in order to calculate hotspot.

```
$selectedThreshold = $_POST['selectedThreshold'];
$startDate = $_POST['startDate'];
$endDate = $_POST['endDate'];

$tmp = exec("/Library/Frameworks/Python.framework/Versions/3.6/bin/python3
/Users/macbookpro/Sites/epidem/epidemGIS/source/gistar.py compute-hotspots
$startDate $endDate $selectedThreshold False 2>&1", $output, $return);

$hotspot = array();
foreach ($output as $value) {
  list($part1, $part2) = explode(',', $value);
  $hotspot[$part1] = $part2;
}
```

Figure 4.13 How the python file is executed from PHP file

```python
# Calculate G* for all points
def z_score_gi_star(spatialWeight, incidentCasesByLocation, tambonID, rowStandardize = True): # int location_i = 650001, float threshold = 0.
    with open("log/z_score_gi_star.txt", "w") as log:
        n = tambonListNumber
        A = B = C = D = totalWeight = totalWeightSquare = 0

        spatialWeightAtLocationI = search_weight(spatialWeight,tambonID)
        if(rowStandardize == True):
            spatialWeightAtLocationI = row_standardize(spatialWeightAtLocationI)

        for x in range(len(spatialWeightAtLocationI)):
            A = A + (spatialWeightAtLocationI[x][-1] * incident_location_j(incidentCasesByLocation,spatialWeightAtLocationI[x][1]))
            #print(str(spatialWeightAtLocationI[x][-1]) + " : " + str(incident_location_j(incidentCasesByLocation,spatialWeightAtLocationI[x]
            totalWeight = totalWeight + spatialWeightAtLocationI[x][-1]
            totalWeightSquare = totalWeightSquare + (spatialWeightAtLocationI[x][-1] ** 2)
        B = totalWeight * mean_incident(incidentCasesByLocation)
        C = (variance_incident(incidentCasesByLocation) * (( n * totalWeightSquare) - (totalWeight ** 2))) / (n - 1)

        log.write('totalWeight ' + str(totalWeight) + '\n')
        log.write('totalWeightSquare ' + str(totalWeightSquare) + '\n')
        log.write('spatialWeightAtLocationI ' + str(spatialWeightAtLocationI) + '\n')

        zScore = (A - B) / (math.sqrt(C))
        log.close()
        return zScore
```

Figure 4.14 Example of a function in the gistar.py file

When the gistar.py returns the result to findHotSpot.php, the data will be re-formatted to the GeoJSon file.

```php
$features = array();
foreach($data as $key => $value) {
    foreach ($hotspot as $hotKey => $hotValue) {
      if($hotKey == $value['tambon_idn']){    //check if the same addr code
          $features[] = array(
          'type' => 'Feature',
          'properties' => array('gid' => $value['gid'],'addrcode'=> $value['tambon_idn'], "zScore" => $hotValue),
          'geometry' => $value['st_asgeojson']
         );
      }
    }
}

$new_data = array(
 'type' => 'FeatureCollection',
 'features' => $features,
);
```

Figure 4.15 How to re-format the data to GeoJSon format

Finally, the result in GeoJSon will be returned to the homepage.php and displayed on the map.

```
function displayHotspot(data){    //data is the result
    L.geoJson(JSON.parse(data), {
          style: hotspotStyle
      }).addTo(map);
}
```

Figure 4.16 How to display the hotspot analysis result on the map

4.2.5 Incidence Prediction

When user wants to calculate the incidence prediction, the prediction function in homepage.php will be called, and findReg.php will be run.

```
function prediction(){
  $.ajax({
      type: "POST",
      url: 'findReg.php',
      data: {},
      success: displayPrediction
  })
}
```

Figure 4.17 Prediction function

The findReg.php is used to call reg.py, get the incident prediction result, and prepare the result to be displayed on the map.

```
$tmp = exec("/Library/Frameworks/Python.framework/Versions/3.6/bin/python3 /Users/macbookpro/Sites/epidem/epidemGIS/source/reg.py rho", $output, $return);
$tmp = exec("/Library/Frameworks/Python.framework/Versions/3.6/bin/python3 /Users/macbookpro/Sites/epidem/epidemGIS/source/reg.py residual", $output2, $return);
$tmp = exec("/Library/Frameworks/Python.framework/Versions/3.6/bin/python3 /Users/macbookpro/Sites/epidem/epidemGIS/source/reg.py pred", $output3, $return);
```

Figure 4.18 How the findReg.php calls the reg.py

```python
ww=ps.io.open("test_hotspot.gwt")
db=ps.io.open("test_hotspot.csv")

ds_name = "test_hotspot.csv"
y_name = "amount"
y = np.array(db.by_col(y_name)).T
y.shape = (len(y),1)

x_names = ["tambon_id"] #REPLACE with rf, elevation, temperature
x = np.array([db.by_col(var) for var in x_names]).T

w = ww.read()
ww.close()
w_name = "test_hotspot.gwt"
w.transform = 'r'

mllag = ML_Lag(y,x,w,name_y=y_name,name_x=x_names,name_w=w_name,name_ds=ds_name, method = 'full')

def main(function):
    if function == 'rho': # gistar.py get-threshold
        print(np.around(mllag.rho, decimals=10))
    if function == 'residual': # gistar.py get-threshold
        print(np.around(mllag.u, decimals=10))
    if function == 'pred': # gistar.py get-threshold
        print(np.around(mllag.predy, decimals=4))

# print(np.around(mllag.rho, decimals=10)) #rho value
# print(np.around(mllag.u, decimals=10)) #residual
# print(np.around(mllag.predy, decimals=4)) #pred values

if __name__ == '__main__':
    if len(sys.argv) == 2:
        main(sys.argv[1])
    else:
        print("Please check arguments")
```

Figure 4.19 Example of reg.py file

```php
foreach($data as $key => $value) {
      $features[] = array(
      'type' => 'Feature',
      'properties' => array('gid' => $value['gid'],'addrcode'=> $value['tambon_idn'],
      "zScore" => $hotValue, "rho" => $rho_final, "residual" => $residual[$i], "pred" => $pred[$i]),
      'geometry' => $value['st_asgeojson']
     );
     $i++;
}

$new_data = array(
 'type' => 'FeatureCollection',
 'features' => $features,
);
```

Figure 4.20 How to re-format the result to GeoJSon format

Then, the result will be added on the map with pop-up box by using the leaflet library as shown in the figure 4.20

```
function displayPrediction(data){
    console.log(data);
    L.geoJson(JSON.parse(data), {
          style: hotspotStyle,
          onEachFeature: function(feature, layer) {
            layer.bindPopup('rho value: '+feature.properties.rho+'<br> residual: '+feature.properties.residual+'<br> pred: '+feature.properties.pred);
          }
    }).addTo(map);
  }
```

Figure 4.21 How to add the result to the map in homepage.php

In this module, PySAL, spatial econometrics toolbox is implemented in our work for ease us to accomplished deploying Spatial Lag Regression on site. This module had deal with intensive mathematic work e.g., Calculus, Optimization problem and Probability. Spatial Lag Regression required data that could be represents as spatial way e.g., number of populations in an area.

In table 4.1, the metrics are shown to indicates the performance of models, AIC and SIC metrics are derived from information theory and only meaningful when there are alternative models to be compared with. The lower value of AIC and SIC are considered as the best model in a set of models. They mean how much information lose during inferencing. The higher value of both means R-squared is coefficient of determination. The 1.0 R-squared means the model's input could perfectly explain the observation values. And 0 means the model's input could not explain the observation values. The result is shown below in table 4.1, AIC and SIC of Spatial Error Regression has slightly better result. There are lost in information during inferencing. And R-Squared implies that the input could determine the observation value not very well.

Table 4.5 Model performance

Metrics	Spatial Lag	Spatial Error
Akaike Information Criteria	476.149	474.649
Schwarz Information Criteria	493.877	489.845
R-Squared	0.3361	0.3260
Mean Squared Error	8.399	8.543

4.3 User Interface

The first page of web-based software of this project, there will be world map, and we are focusing on only Phitsanulok, Thailand. Therefore, there will be pins on only the focused area. Note that a pin refers to a sub-district.

User can select month and year of data on the map by using the tool at center of the top.

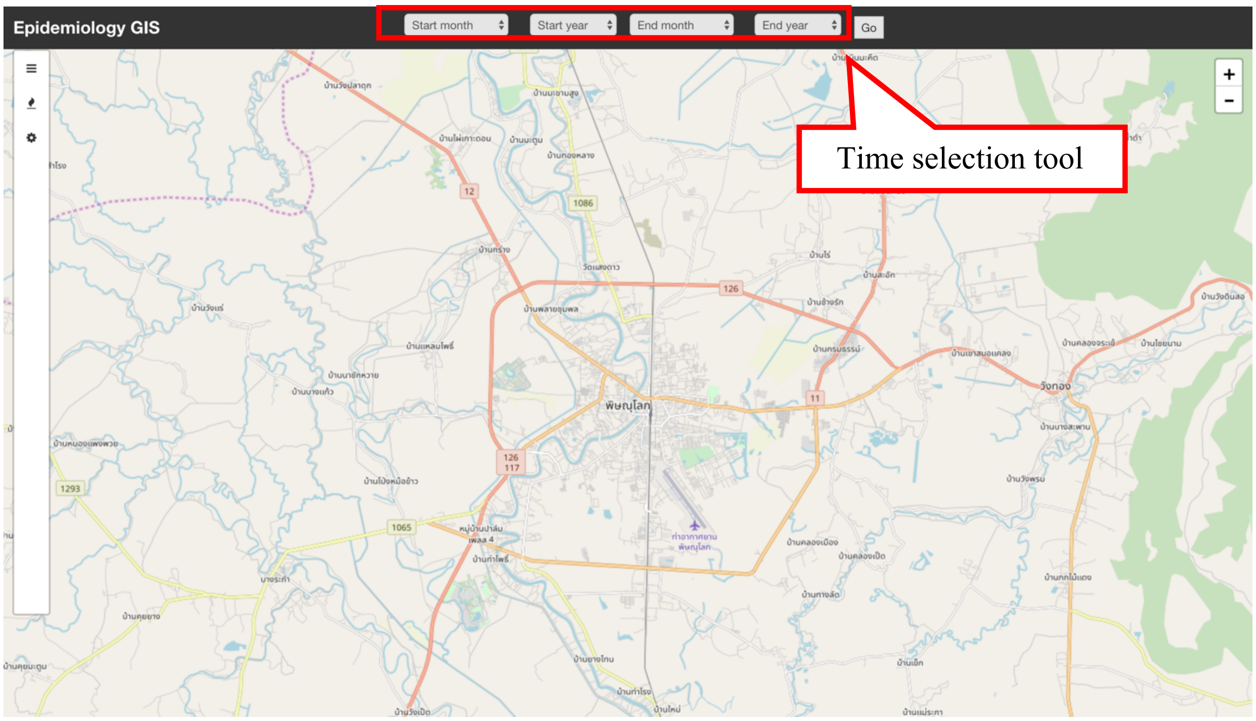

Figure 4.22 The first page of epidemiology GIS

User can click on the plus or minus sign at the right top, when the map is zoomed in, it will show more detail of each area as shown in the Figure 4.22.

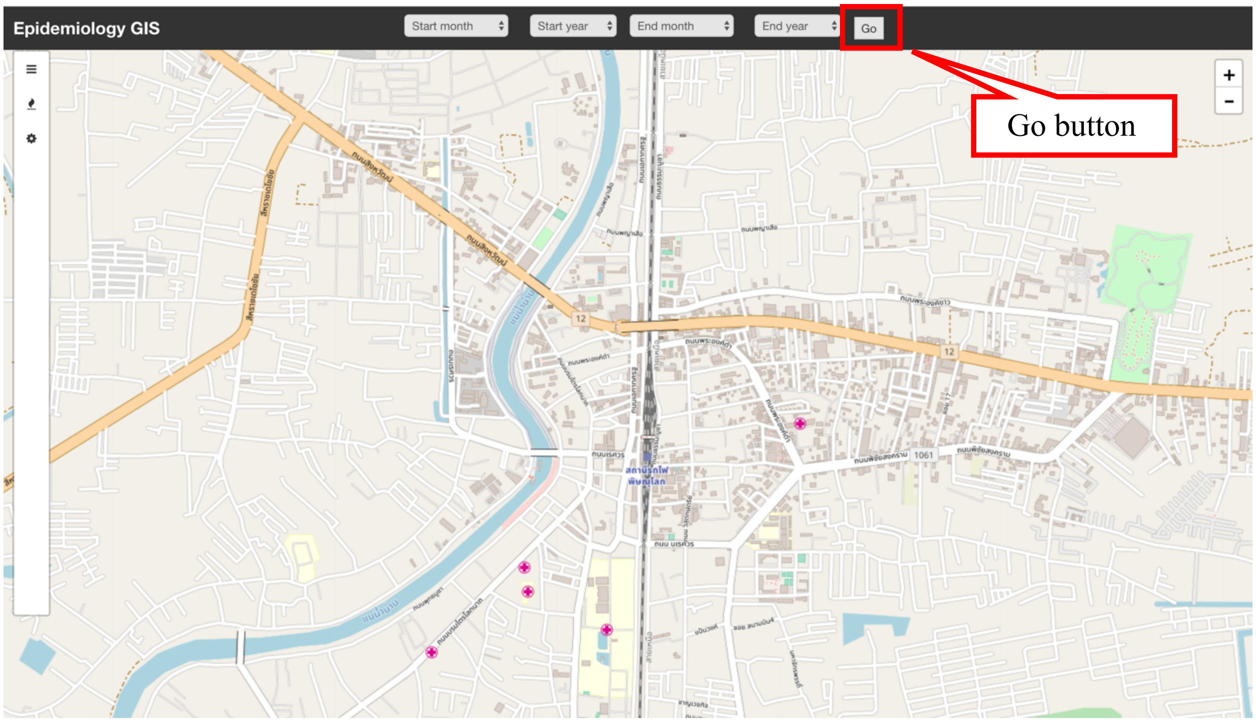

Figure 4.23 Example of zoom-in

If users want to see number of incident cases in each sub-district area with a period, they can do it by using the black navigation bar.

First of all, the users need to click on the “Go” button. Then all layers will be ready to be shown on the map, and the incident prediction will be ready also.

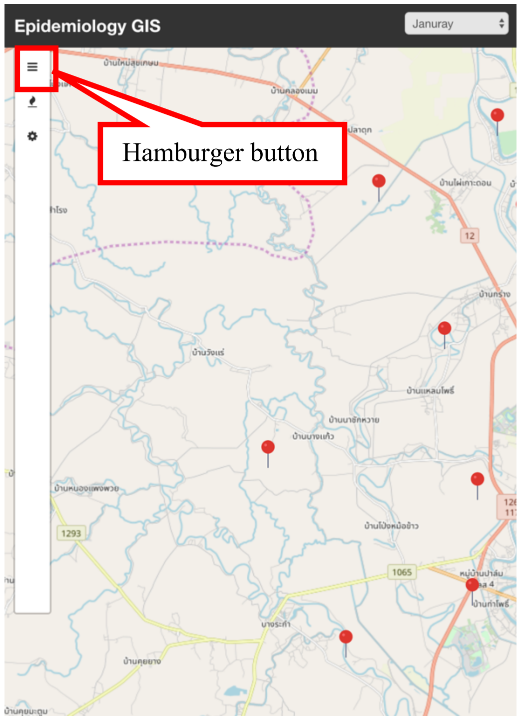

Figure 4.24 The hamburger button

When users want to predict value of cases, they can customize the factors of the prediction by themselves. To do prediction, user needs to click on the hamburger ( ) button at side menu, window of incidence prediction will be shown as in Figure 4.24. The user can select a prediction model, time and threshold.

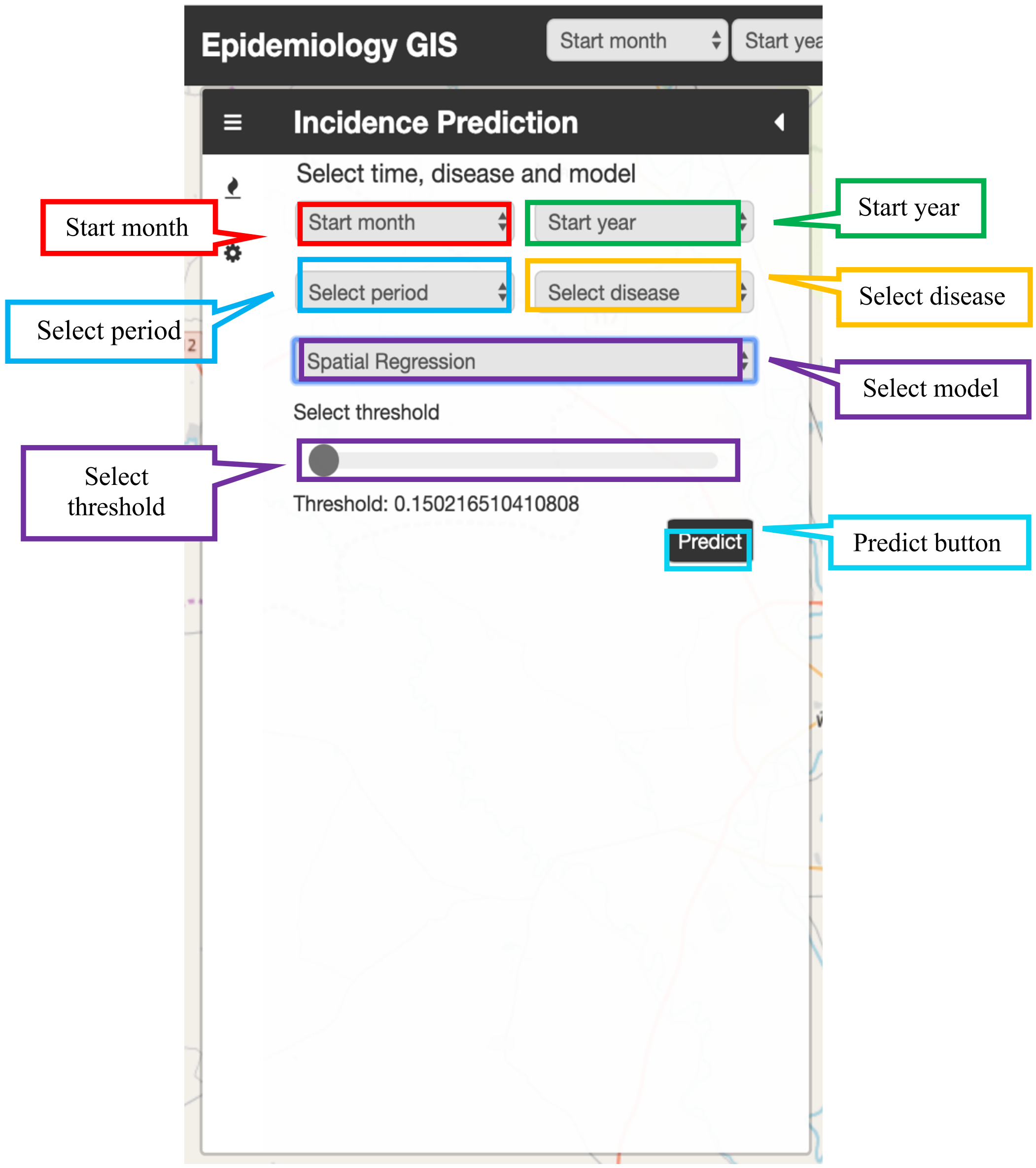

Figure 4.25 Show prediction panel

For time selection, users are allowed to select start month and year of the period, and to select period of the prediction: 1 month, 3 months and 6 months. For example, if start month is November, and 3 months are selected, the prediction value will be the result of November 2018 to January 2019.

For disease selection, by using mapping system, the users can choose which disease they want to predict, but in this project, we started with a disease called Dengue Fever (i.e., DF, DHF and DSS).

For model selection, it shows that the user can select the models that they can use. For factor selection, user can select more than one factor.

When the user clicks on predict button, the result will be shown in the map as shown in the Figure 4.26.

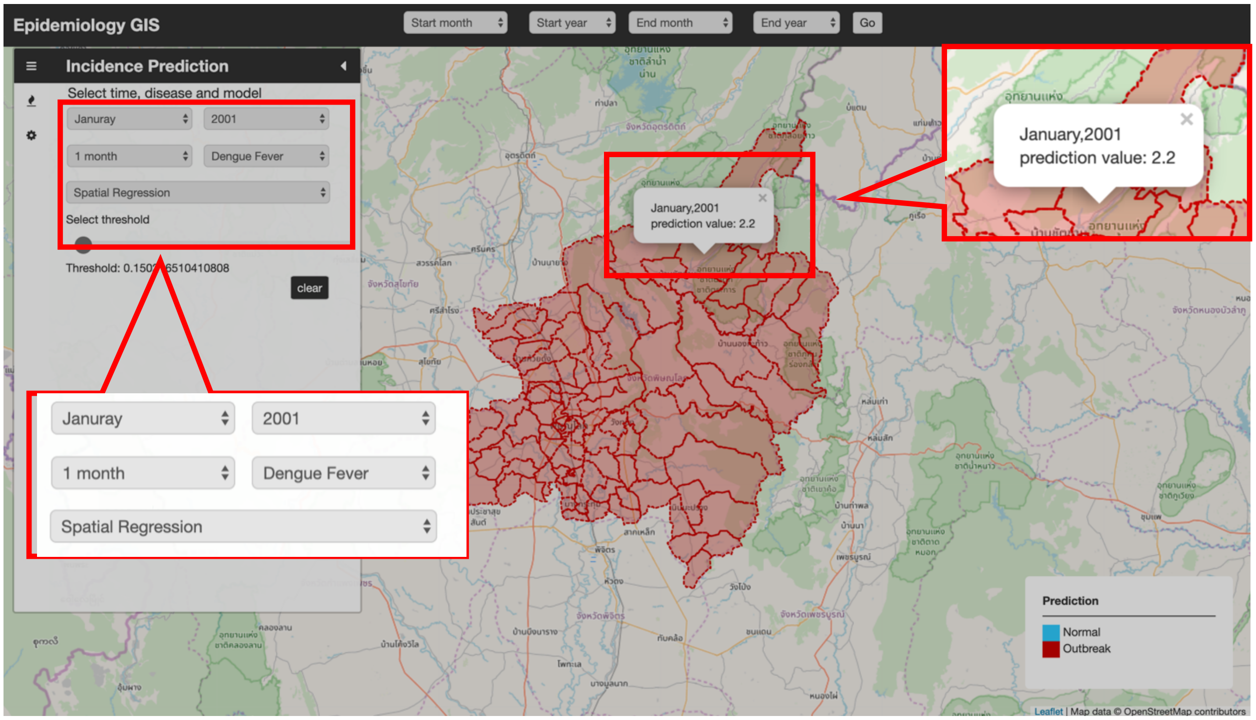

Figure 4.26 Example of incidence prediction result

Prediction value is number of the predicted cases. If the prediction is equal or more than 0.5, there is outbreak. In the figure 4.26, a pop-up of a sub-district shows that the prediction value is 2.2. Therefore, In January,2001, the sub-district is an outbreak area.

In addition, user is able to show hotspot area. When the user clicks on the fire ( ) button, the hotspot analysis window will be shown as in the Figure 4.27.

For measurement tools, there will be a threshold range slider and list of measurement tools/approach that user can select.

Threshold is radius that a sub-district will affect to nearby neighbors.

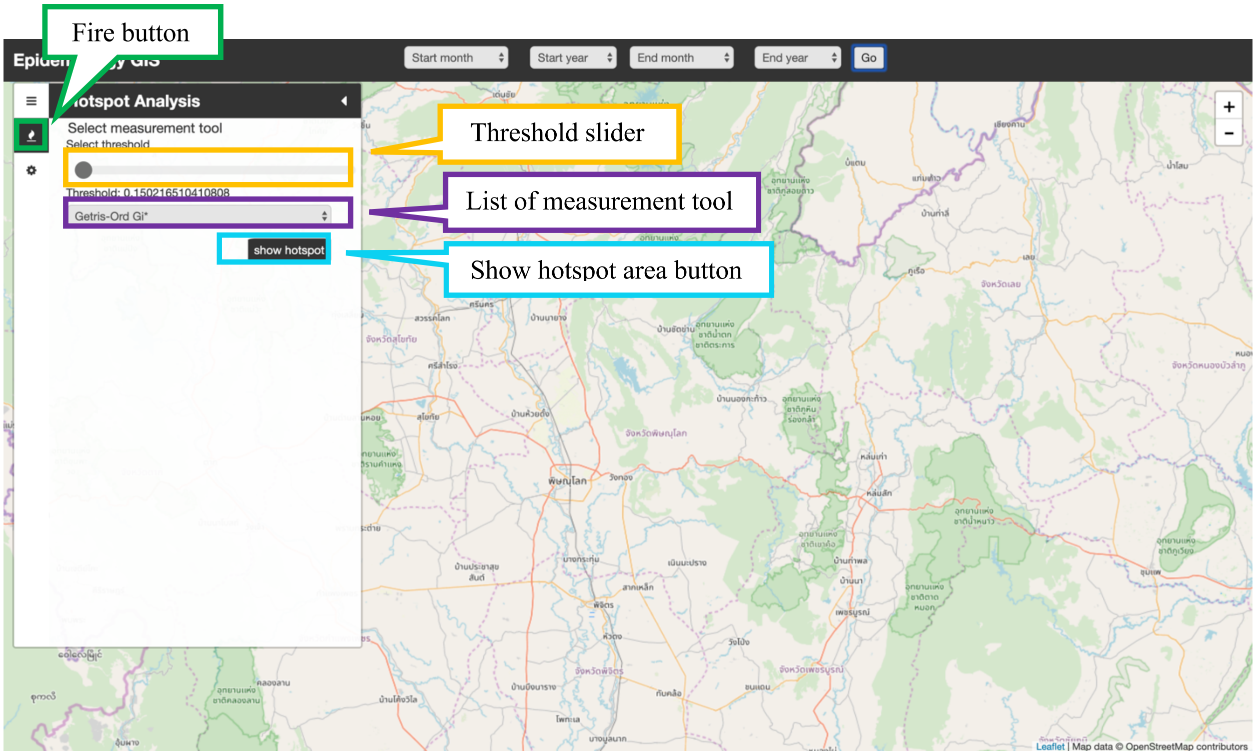

Figure 4.27 Hotspot analysis window

In Figure 4.28, when user click on the "show hotspot area" button, the hotspot area will be highlighted by using red color. The button will become clear hotspot which lets user clear the hotspot on the map.

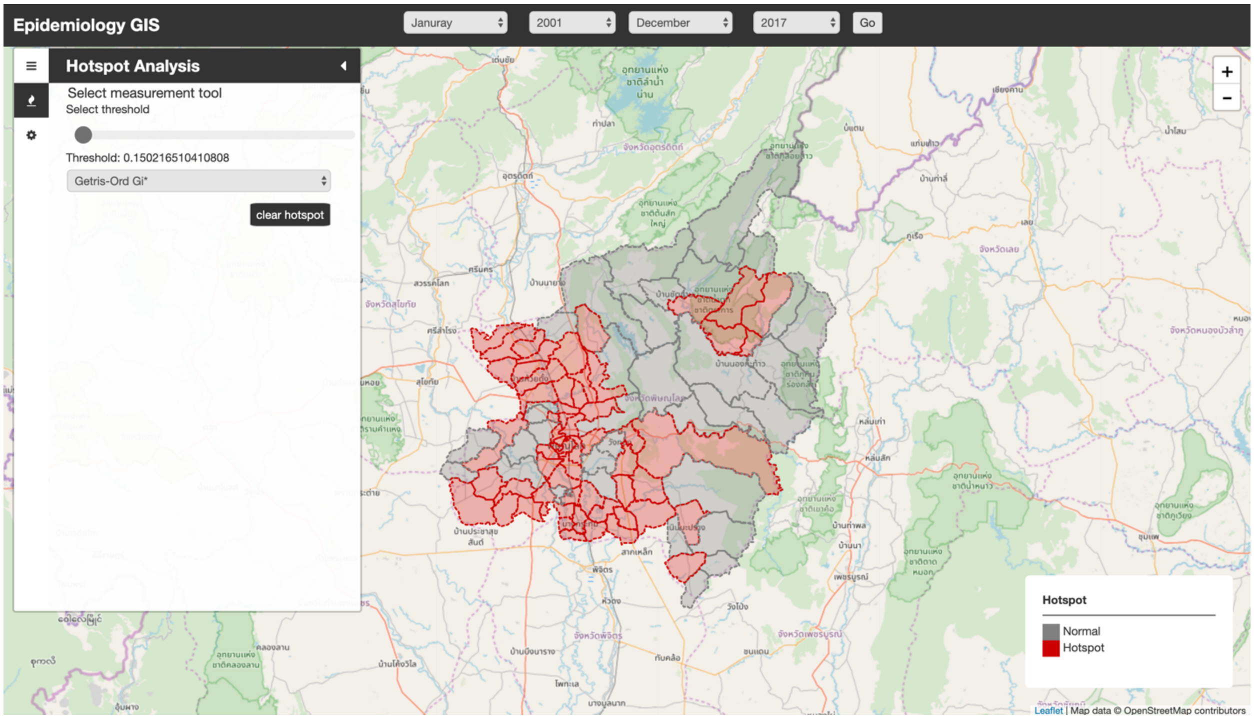

Figure 4.28 Example of hotspot analysis

S in the figure 4.29. These buttons will let user to display the layer(s) on the map.

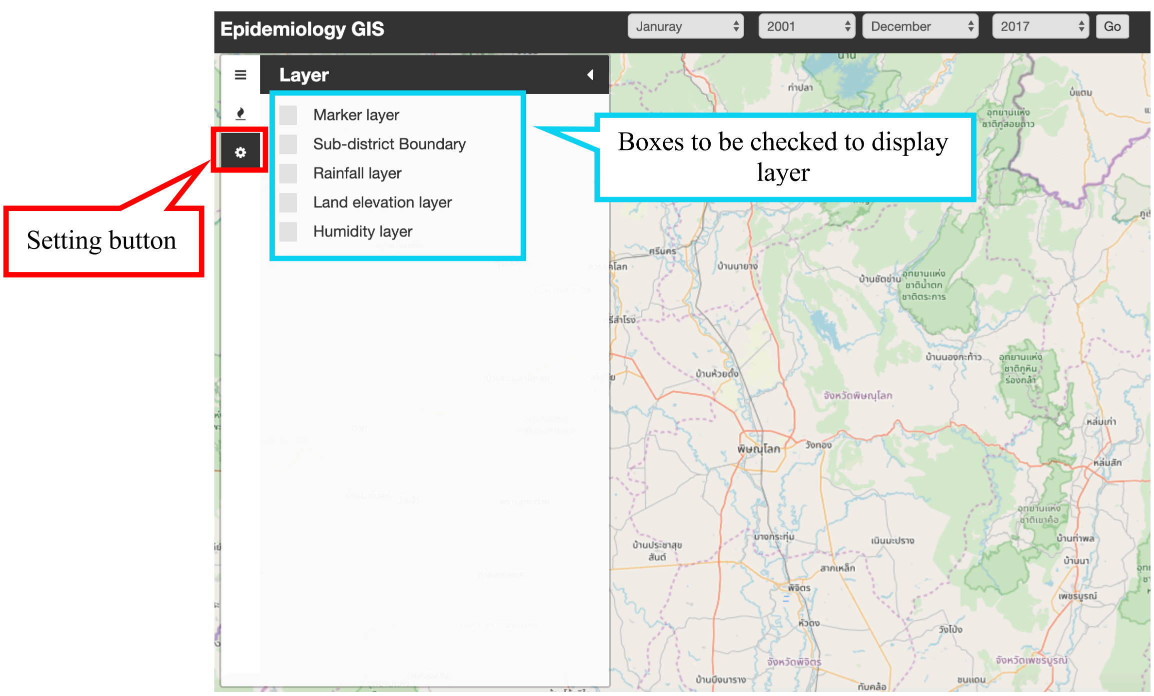

Figure 4.29 Layer window

Marker layer is a layer with pins. A pin refers to a sub-district that its number of incident cases is not null. The figure 4.30 is a good example of showing incident cases.

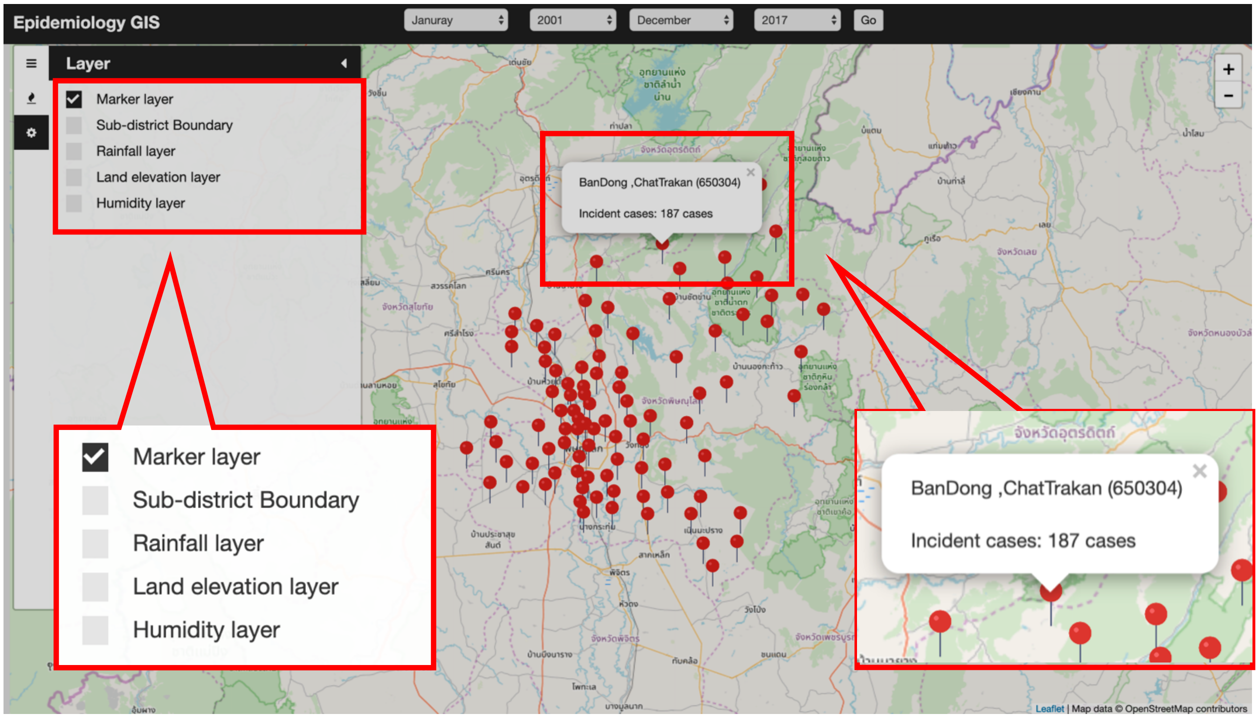

Figure 4.30 Example of marker layer

Figure 4.31, 4.32, 4.33 and 4.34 show example of displaying layer on the map. Note that checked box means layer is on, and unchecked box means display is off.

For rainfall layer, there are two size of lines. The thickness means the average rainfall of the sub-district is higher than median, and the thinness means it is lower than the median.

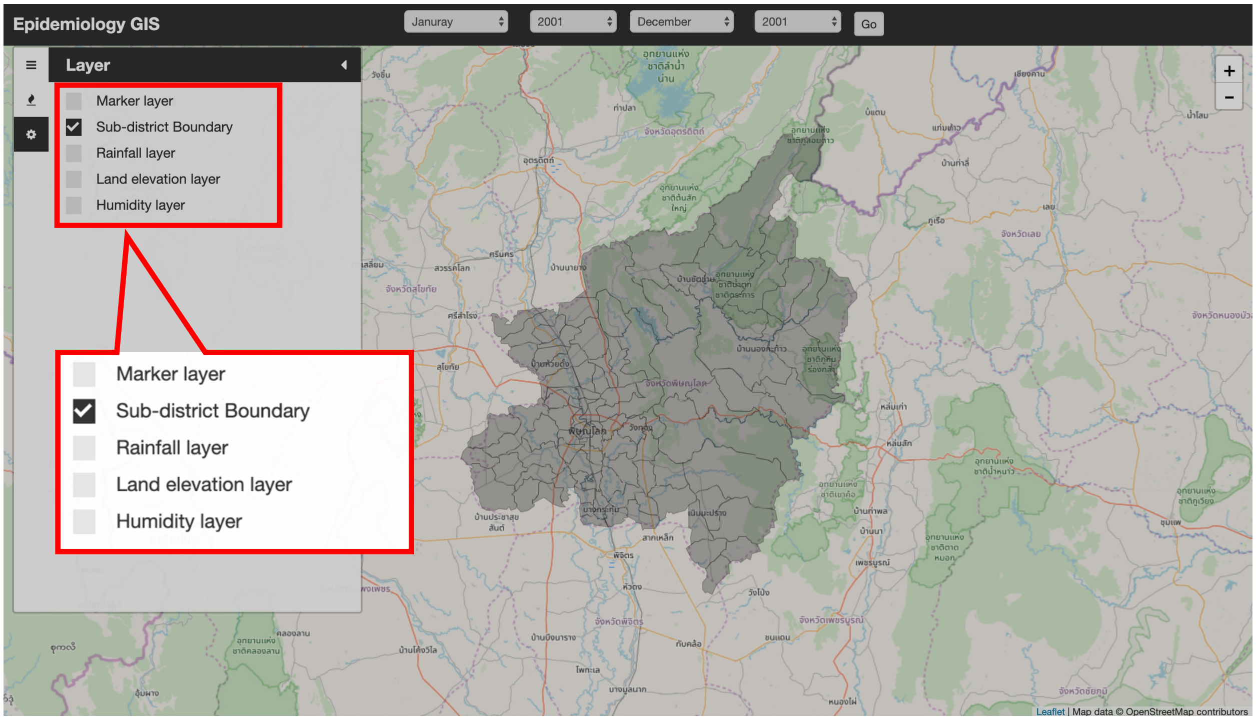

Figure 4.31 Example of displaying sub-district boundary on the map

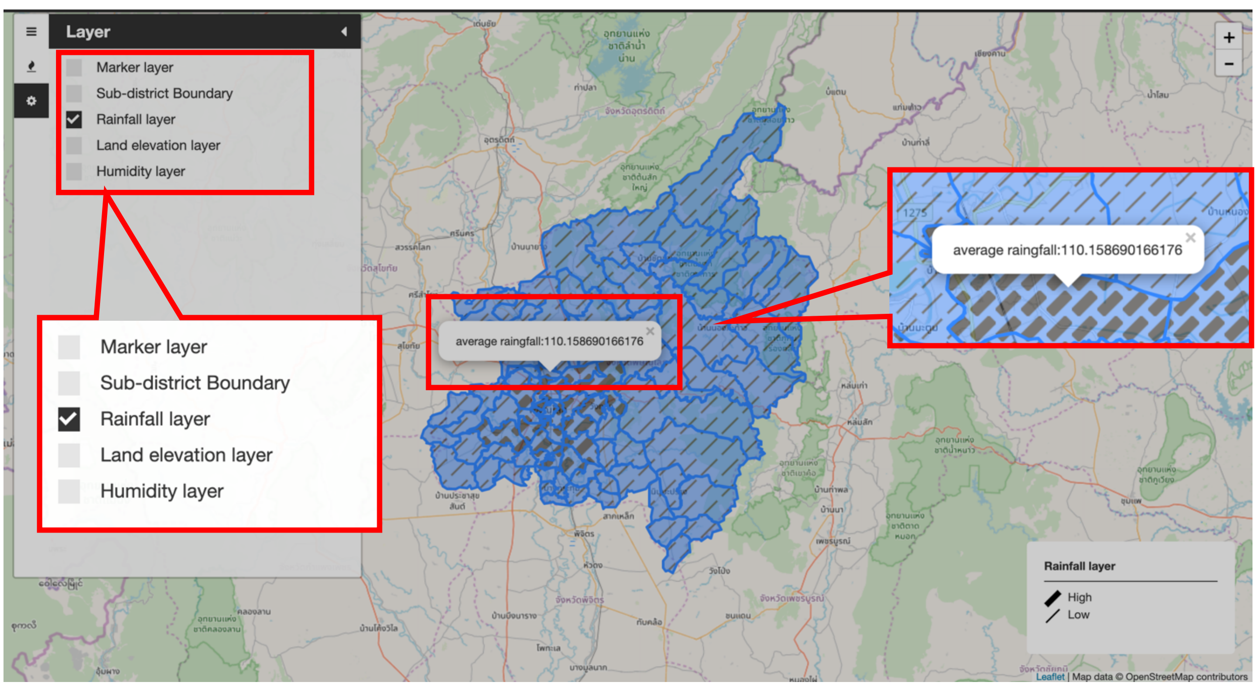

Figure 4.32 Example of displaying rainfall layer on the map

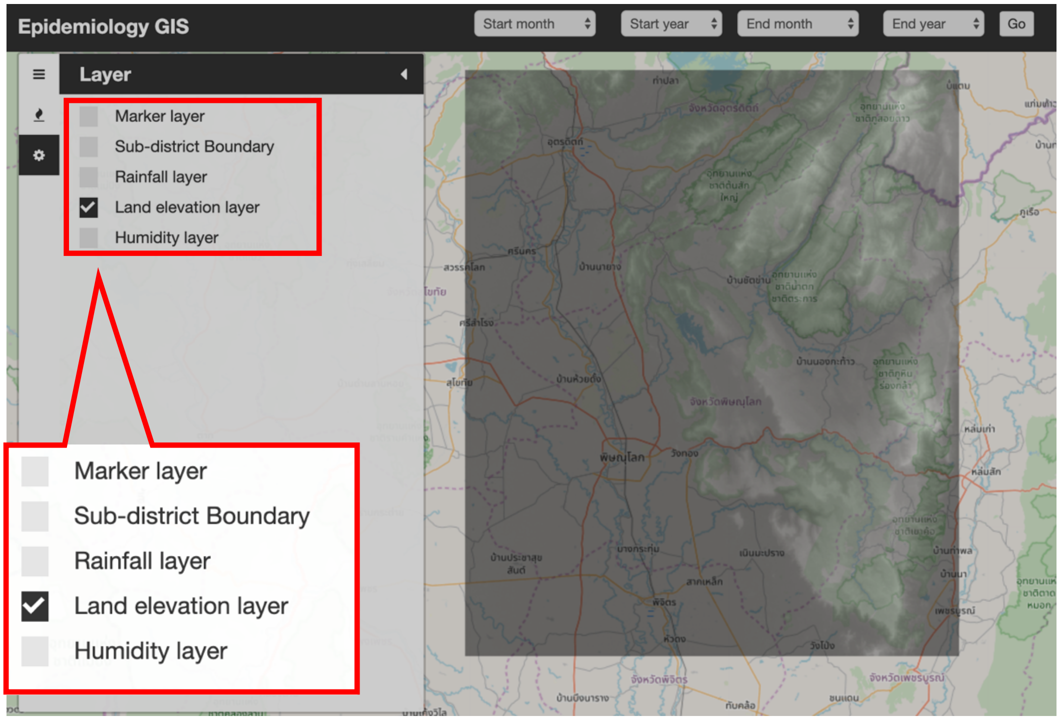

Figure 4.33 Example of displaying land elevation layer on the map

For the humidity in the figure 4.32, color of each polygon depends on average of humidity quantity of the selected period. We classified level of the average into 5 levels as shown in the table

Table 4.2 Humidity Level

Color		Average of humidity(%)
		>75%
		>50%
		>25%
		<=25%

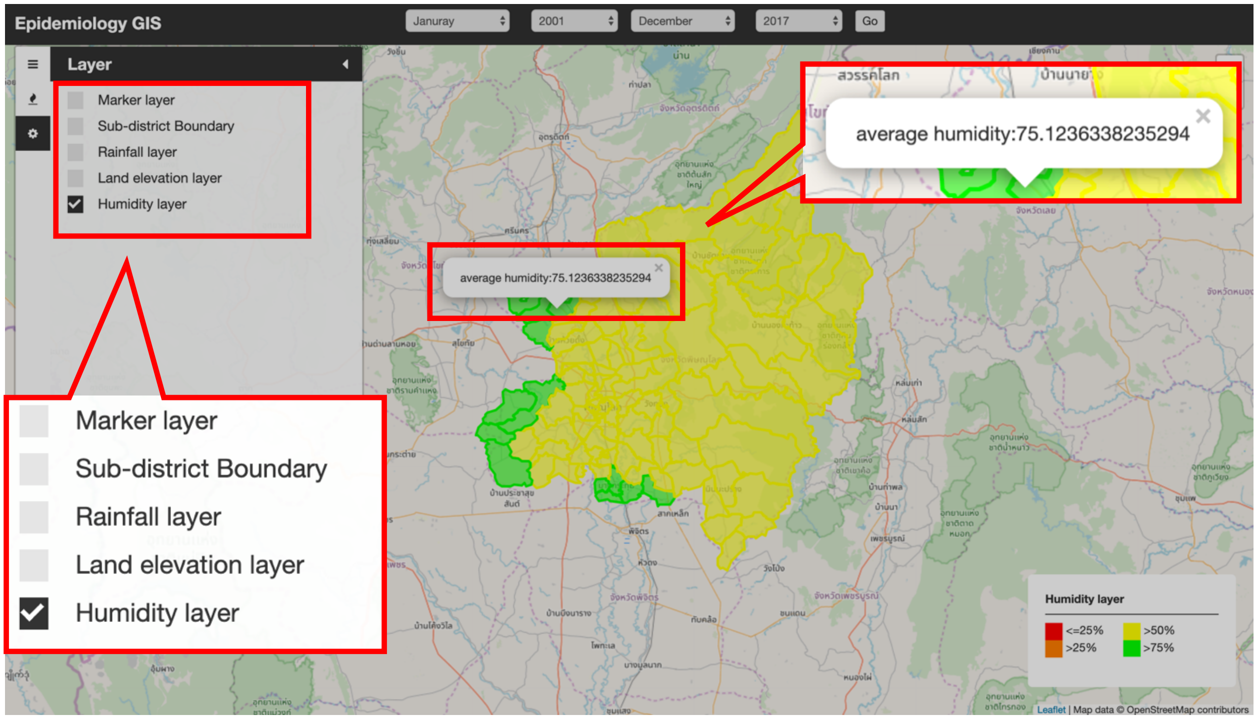

Figure 4.34 Example of displaying humidity layer

Chapter 5

Conclusion

5.1 Completion Status

There many major components in this project. They are listed and showed in the below table.

Table 5.1 Accomplished task

Component	Status	Note
Identified boundary of area	Complete	Phitsanulok
Select infection disease	Complete	Dengue fever
Disease data set	Complete	2001-2017
Architectural design	Complete	
UI design	Complete	
Schema for database	Complete	
Selected model	Complete	
Displaying layers	Complete	
Hotspot analysis parts	Complete	
A map software with map, prediction model, database, result of usability testing, and results of analysis testing/experiment.	Complete	
The software will have disease cases on the map, and also predict spread area by using data in the database	Complete	
The database will have infection disease cases and area details i.e. population, climate, etc.	Complete	

In this project, we have identified the boundary of area, selected the infection disease, found disease data set, created architectural design, designed the user interface, created the schema of the used database, displayed layers, and done with hotspot analysis part, layers and incidence prediction.

5.2 Problem

While we were working on this project, we had to face with many problems. Some can be fixed, but some cannot. The problems are listed in the table 5.2

Table 5.2 list of problems

Problem	Note
To find incidence case data.	Fixed by trying to contact many people. Finally, we could get the data.
To find climate data in sub-district level because there is only data in province level	Fixed by interpolating the data we got from TMD. But These data could be not accurate and leads unreliable result
The selected model focuses on only spatial. Therefore, it is not accurate as the model focuses on both temporal and spatial.	We fixed this problem by using average value of each factor to predict the future incidence cases.

This project, a main part is database. Therefore, the most of problems we have faced is about finding data.

5.3 Stakeholder's Comments

We have asked the stakeholders who are responsible to epidemiology about this software. They said this is a good project that will make their work easier, but there are many things that we need to improve:

- The user interface should be more attractive. For example, we should change the theme.
- It is a bit hard to understand. Therefore, it would be good to have a handbook for user.
- After calculation, it should show the summary of the result.

5.4 Suggestions

This project consists of incidence prediction, hotspot analysis and layer display. Those are what we expected first, but accuracy of the prediction is not good as it should be. The reason is the selected model and available data are not suitable for the project.

To apply this project to the different disease, the software should let users easier to able to manage database and prediction model directly because the different disease has its own characteristics and risk factor.

And, Thai Language is essentially need for this software because the main users are Thais.

References

[1] Retrieved from http://einstein.sc.mahidol.ac.th/~bionanotech/7_Research_Page/ResProj_10/H1N1(2009)/05/no52.pdf

[2] Rogers, S. (2013). *John Snow's data journalism: the cholera map that changed the world*. Retrieved from https://www.theguardian.com/news/datablog/2013/mar/15/john-snow-cholera-map

[3] *How Hot Spot Analysis (Getis-Ord Gi*) works*. Retrieved from http://pro.arcgis.com/en/pro-app/tool-reference/spatial-statistics/h-how-hot-spot-analysis-getis-ord-gi-spatial-stati.htm

[4] *Buffer*. Retrieved from http://desktop.arcgis.com/en/arcmap/10.3/tools/analysis-toolbox/buffer.htm

[5] *Vector vs Raster: What's the Difference Between GIS Spatial Data Types?*. Retrieved from https://gisgeography.com/spatial-data-types-vector-raster/

[6] *Discrete raster*. Retrieved from https://support.esri.com/en/other-resources/gis-dictionary/term/discrete%20raster

[7] Thammapalo, S., Chongsuwiwatwong, V., McNeil, D., & Geater, A. (2005). *The climatic factors influencing the occurrence of dengue hemorrhagic fever in Thailand.* Retrieved from https://pdfs.semanticscholar.org/79dc/f2e007941e32938eb3cfbaa283939dcea742.pdf

[8] Cazelles, B., Chavez, M., & McMichael, A.J. (2005). *Nonstationary Influence of El Niño on the Synchronous Dengue Epidemics in Thailand.* Retrieved from https://journals.plos.org/plosmedicine/article?id=10.1371/journal.pmed.0020106

[9] Wu, P., Lay, J., Guo, H. et al. (2009). *Higher temperature and urbanization affect the spatial patterns of dengue fever transmission in subtropical Taiwan.* Retrieved from https://linkinghub.elsevier.com/retrieve/pii/S0048969708011509

[10] Roslan, N. S., Latif, Z. A., & Dom, N.C. (2016). *Dengue cases distribution based on land surface temperature and elevation.* Retrieved from http://ieeexplore.ieee.org/document/7813307/

[11] Cummings, D.A., Iamsirithaworn, S., Lessler, J.T. et al. (2009). *The impact of the demographic transition on dengue in Thailand: insights from a statistical analysis and mathematical modeling.* Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/19721696

[12] Latif, Z. A., &Mohamad, M. H. (2015). *Mapping of Dengue Outbreak Distribution Using Spatial Statistics and Geographical Information System*. Retrieved from http://ieeexplore.ieee.org/document/7371016/references

[13] Diah, I. M., Aziz, N., & Ahmad, N. (2016). Tuberculosis Disease Mapping with Poisson-Gamma Model in Malaysia. Retrieved from http://docsdrive.com/pdfs/medwelljournals/rjasci/2016/822-825.pdf

[14] Chinmayee, M., Siddharth, S. R., & Manjusha, P. (2017). *Prevention of infectious disease based on big data analytics and map-reduce*. Retrieved from https://ieeexplore.ieee.org/document/8117889/

[15] Nor, A. S., & David F. P. (2017). *Dengue disease mapping in Malaysia based on stochastic SIR models in human populations*. Retrieved from https://ieeexplore.ieee.org/document/6396640/

[16] Kevin, B., & Edward, S. (2011). *Mapping an epidemic outbreak: Effective analysis and presentation.* Retrieved from https://ieeexplore.ieee.org/document/6102486/

[17] Duane, J.G. *Dengue and Dengue Hemorrhagic Fever.* Retrieved from https://cmr.asm.org/content/cmr/11/3/480.full.pdf

[18] Banu, S., Hu, W., Hurst, C., Tong, T. (2011). *Dengue transmission in the Asia-Pacific region: impact of climate change and socio-environmental factors.* Retrieved from https://onlinelibrary.wiley.com/doi/epdf/10.1111/j.1365-3156.2011.02734.x

[19] Agafonkin, V. *Using GeoJSON with Leaflet.* Retrieved from https://leafletjs.com/examples/geojson/

[20] Jeefoo, P., Tripathi, N. K., & Souris, M. *Spatio-Temporal Diffusion Pattern and Hotspot Detection of Dengue in Chachoengsao Province, Thailand.* Retrieved from https://www.mdpi.com/1660-4601/8/1/51/htm

[21] Agafonkin, V. *Leaflet Quick Start Guide*. Retrieved from https://leafletjs.com/examples/quick-start/

[22] *GeoJSON draft version 6.* Retrieved from http://wiki.geojson.org/GeoJSON_draft_version_6

[23] *Using GeoJSON with Leaflet.* Retrieved from https://leafletjs.com/examples/geojson/

[24] *Dengue haemorrhagic fever: diagnosis, treatment, prevention and control. 2nd edition. Geneva : World Health Organization.* Retrieved from http://www.who.int/csr/resources/publications/dengue/Denguepublication/en/

[25] *GeoServer is an open source server for sharing geospatial data.* Retrieved from http://geoserver.org/download/

A Brief Introduction to Spatial Regression. Retrieved from http://www.unsiap.or.jp/e-learning/el_material/Agri/1507_Literacy_KOR/M3_4_Country_Presentation_Korea.pdf

[26] *What is jQuery?*. Retrieved from https://jquery.com/

Andrea, Z. *Spatial Autocorrelation*. Retrieved from http://userwww.sfsu.edu/efc/classes/biol710/spatial/spat-auto.htm

Buffer. Retrieved from http://desktop.arcgis.com/en/arcmap/10.3/tools/analysis-toolbox/buffer.htm

Chapter 1. What is epidemiology?. Retrieved from http://www.bmj.com/about-bmj/resources-readers/publications/epidemiology-uninitiated/1-what-epidemiology

Claudia, D., Dante S., Robert, W., & Samuel, W. (2016). *Spatial Queries.* Retrieved from http://www.gitta.info/SpatialQueries/en/text/SpatialQueries.pdf

Elliott, P., & Wartenberg, D. (2011). *Spatial Epidemiology: Current Approaches and Future Challenges.* Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1247193/

Epidemiology. Retrieved from https://en.wikipedia.org/wiki/Epidemiology

Epidemiology. Retrieved from http://www.scientificspine.com/spine-dictionary/what-is-epidemiology.html

Epidemiology: some basic concepts and definitions. Retrieved from http://www.fao.org/Wairdocs/ILRI/x5436E/x5436e04.htm

Dengue Fever. Retrieved from https://www.webmd.com/a-to-z-guides/dengue-fever-reference#1

Dengue Viruses. Retrieved from https://www.nature.com/scitable/topicpage/dengue-viruses-22400925

Demers, M.N. (2005). *Fundamental of Geographic Information System*. Hoboken, NJ: John

Wiley & Sons, inc.

GEO 465/565 - Lectures 11 and 12 - "Spatial Analysis". Retrieve from https://dusk.geo.orst.edu/gis/lec11_12.pdf

Geographical Information Systems (GIS). Retrieved from http://www.manage.gov.in/studymaterial/GIS.pdf

How to Build Spatial Regression Models in ArcGIS. Retrieved from https://gisgeography.com/spatial-regression-models-arcgis/

Introduction to Hotspot Analysis. Retrieved from https://www.cdc.gov/dhdsp/maps/GISX/training /module3files/3_hotspot_analysis_module.PDF

Kumar, B. (2008). *Basic concept of Epidemiology*. Retrieved from https://www.gfmer.ch/Medical_education_En/Afghanistan_2008/pdf/Basic_concept_epid emiology_Kumar_Afghanistan_2008.pdf

Leaflet. Retrieved from https://leafletjs.com/index.html

Overlay Layers. Retrieved from https://doc.arcgis.com/en/arcgis-online/analyze/overlay-layers.htm

Pattarakavin, P., & Piromsopa, K. (2017). *Development of epidemiology data map visualization system*. Retrieved from http://ieeexplore.ieee.org/document/8166388/

Rouse, M. (2006). *Bayesian Logic.* Retrieved from https://whatis.techtarget.com/definition/Bayesian-logic

Spatial analysis. Retrieved from https://en.wikipedia.org/wiki/Spatial_analysis

Spatial Data. Retrieved from https://www.techopedia.com/definition/871/spatial-data

Spatial Interaction. Retrieved from http://geography.name/spatial-interaction/

The GIS Spatial Data Model. Retrieved from https://courses.washington.edu/gis250/lessons/introduction_gis/spatial_data_model.html

The Standard Regression Model and its Spatial Alternatives. Retrieved from www.utdallas.edu/~briggs/henan/13SpatReg.ppt

Ward, M. Spatial Epidemiology: *Where Have We Come in 150 Years?*. Retrieved from https://link.springer.com/chapter/10.1007/978-1-4020-8507-9_13

What is a coverage. Retrieved from http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=What_is_a_coverage

What is Hotspot Analysis?. Retrieved from https://glenbambrick.com/2016/01/21/what-is-hotspot-analysis/

What is PostGIS?. Retrieved from https://www.gislounge.com/what-is-postgis/

Xiaomeng, H., Tian, P., Huabin, R., Haohuan, F., & Guangwen, Y. (2013). *Parallel Buffer Generation Algorithm for GIS*. Retrieved from https://www.omicsonline.org/open-access/parallel-buffer-generation-algorithm-for-gis-2329- 6755.1000115.php?aid=14031

GeoJSON. Retrieved from https://en.wikipedia.org/wiki/GeoJSON

GeoJSON. Retrieved from http://geojson.org

Web Map Service (WMS). Retrieved from https://www.techopedia.com/definition/876/web-map-service-wms

Web Map Service. Retrieved from https://en.wikipedia.org/wiki/Web_Map_Service