CITY OF WINNIPEG ICB: MODELLING MOSQUITO POPULATIONS

ANDRII ARMAN, JONATHAN GALLAGHER, AND AIDIN ZAHERPARANDAZ

Abstract. The city of Winnipeg's Insect Control Branch (ICB) is interested in predictive models of mosquito population based on environmental data. In particular, ICB determines whether to spray or not a given region of the city based on a mosquito trap count.

In this project we collect and aggregate data provided by the city of Winnipeg and external sources, determine key weather factors that influence mosquito population, and provide a model (Random Forest Classifier) that determines whether mosquito count in a given region of a city is larger than a given threshold. As a part of data collection we develop a computer vision tool for precipitation data extraction from satellite images.

1. Introduction

The City of Winnipeg Insect Control Branch (ICB) provides services to public to control mosquito population. Mosquito control program includes helicopter larviciding program, ground larviciding program, and monitoring adult nuisance mosquitoes (New Jersey Light Traps).

Main challenges for ICB are: predictive modelling of rainfall/soil moisture, and predictive modelling of larval and adult mosquito population. In particular, a key factor for ICB on deciding whether to spray in a given location of a city is whether a mosquito count in a given location exceeds 25.

Our main contributions are in the following three directions.

To start with, a large part of our work is devoted to data collection and data aggregation. For instance, we use weather data from external source visualcrossing.com, where only average metrics were given for a city of Winnipeg. In a pursuit of more localised weather information (in particular precipitation), a computer vision tool is developed that can be used for weather information extraction from satellite images, weather maps, etc. We discuss data collection and a concept of data silo in Section 2.

Second, we determine key factors that influence mosquito population. For instance minimal temperature yesterday happens to be the most significant weather factor for today's mosquito population. The analysis is done via important features of linear regression model in Section 3.

Finally, we obtain a predictor of whether a $trap\_loc[date] > 25$, where $trap\_loc[date]$ is a mosquito count in a given location on a given date (ICB makes a decision to spray a given location if $trap\_loc[date] > 25$). Our predictor of $trap\_loc > 25$ is a Random Forest Classifier that has good accuracy ($> 89\%$ for all regions) and precision ($> 84\%$ for all but three locations). Description of this predictor is given is Section 4.

2. Towards distributed and accessible city-data

2.1. **Data silo.** The concept of a *data silo* refers to a situation where organisational data is not collaboratively accessible. Data silos occur for different reasons; for example, the data could be held by different groups within an organisation (perhaps unknown to each other), or stored in seemingly incomparable formats. Note: often data silos refer to informational systems that should be linked together, and often excludes insulated data warehouses that are necessarily incompatible – in other words, linking data between silos would not, in theory, break the principle of least access. Data silos present a significant obstacle to large institutions (governments, large corporations) [3] because they lead to

(1) replicated efforts in data analysis;
(2) unawareness of the available data;
(3) overly specialised and non-transferable knowledge;
(4) a lack of shared incentives and objectives [2].

Breaking down data silos can lead to more shared incentives as well as faster and more complete analysis [3].

An additional obstacle for city-level data is that often the data is collected over multiple decades and is potentially dangerous or disruptive to reformat. A general solution to this sort of age-layered problem is known as continual improvement [1]. In the remainder of this section, we describe our efforts to start continuous transformation of the disparate and sometimes lacking data sets using a lightweight approach to what is known as data virtualisation.

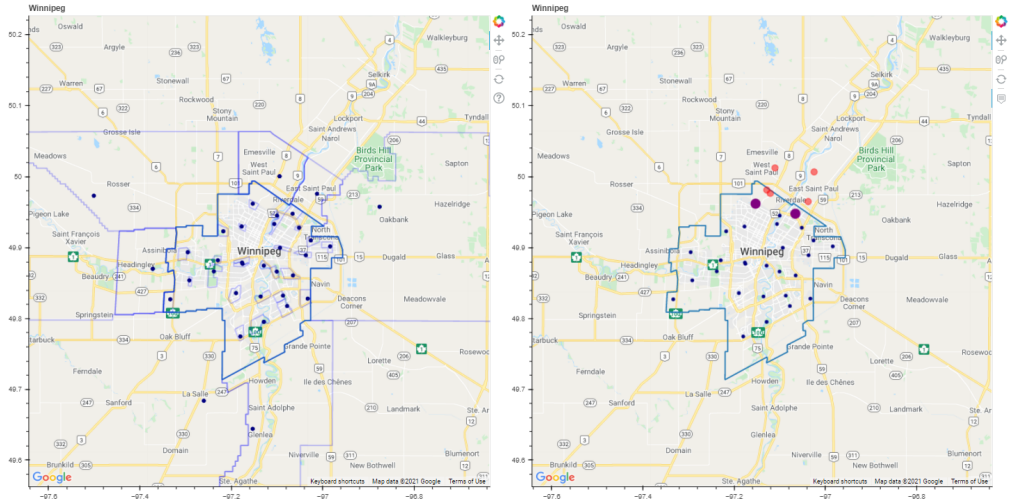
Data virtualisation refers to techniques that allow amalgamating data from disparate, heterogeneous, and discordant sources. Our data sources include:

(1) existing data sources owned by the ICB;
(2) data sources owned by other departments are added;
(3) relevant data sources owned by outside departments (e.g. the European Space Agency and visualcrossing.com) are added.

2.2. **Amalgamating and harmonising data.** The following data sources are amalgamated into a single data-frame:

- A database of daily mosquito trap counts for 2015-2021;
- A file containing mosquito trap locations (36 locations);
- 311 City requests (mosquito complaints) for 2014-2021;
- A database of helicopter larviciding data for 2020-2021;
- A database of ground larviciding data for 2003-2021;
- A listing of ground larviciding locations;
- Weather data for 2015-2021 (externally sourced from visualcrossing.com).

After a simple amalgamation of the databases, we added spatiotemporal dimension to the dataframe. For example, daily mosquito trap is indexed by date with columns being trap locations: 28 locations $NW_1, ..., SE_7$ and 8 locations outside city $AA, ..., HH$. This data is then cross-referenced with the mosquito trap location data. Mosquito trap locations is a JSON file that for each trap name provides a polygon in which trap is located. See Figure 1a, blue dots represent the center of a corresponding trap polygon. This means that within the



(A) Trap location areas and approximate locations

(B) Helicopter spraying and closest traps

FIGURE 1. Visualisation tool use example (for 2021/5/27)

amalgamated dataframe, mosquito trap counts can be looked up by date and location within the city, and also that date-location may be looked up to determine mosquito trap counts.

Mosquito complaints are indexed by date, and contain the location of a complaint. While we did not explicitly extend the dataframe with 311 data indexed by date, date-location indices can easily be extended to include 311 data.

Helicopter larviciding data is indexed by datetime and contains, principally, a polygon indicating where spraying occurred. We replaced spraying polygons with their centers, and for each spraying we found the closest trap to the spraying location. This data was categorical; for each day and each trap we recorded 'yes' or 'no' based on whether or not spraying occurred within a bounding box containing the trap. For example, on Figure 1b spraying locations (red dots) on a given day (2021/5/27) are presented; the closest traps (NW6, NE1) are highlighted in purple.

Ground larviciding data is similar design to helicopter larviciding data. We received ground larviciding data too close to the end of the project to incorporate it, but the same procedure as used for helicopter larviciding data could be used to ground larviciding as well.

Weather data includes temperature (average, min, max), precipitation, wind, etc (obtained from visualcrossing.com). This data contains only city averages, and is added to the dataframe by date, as each location is taken to have the same data. Initial analysis indicates that the most important weather features are temperature, precipitation and cloud cover.

To create Figures 1a and 1b we use the Bokeh Python library for creating interactive maps. The resultant maps and the ease of overlaying additional data on demand could be developed further into a powerful and flexible visualisation tool.

2.3. **More accurate rainfall data.** Since precipitation data is identified as a significant data point for mosquito population modelling, the accuracy of precipitation data is important. The

data from visualcrossing.com provides only city averages, and some accuracy issues have been identified. The City of Winnipeg, Water and Waste department maintains maps of rainfall as collected by multiple sensors spread throughout the city (see Figure 2).

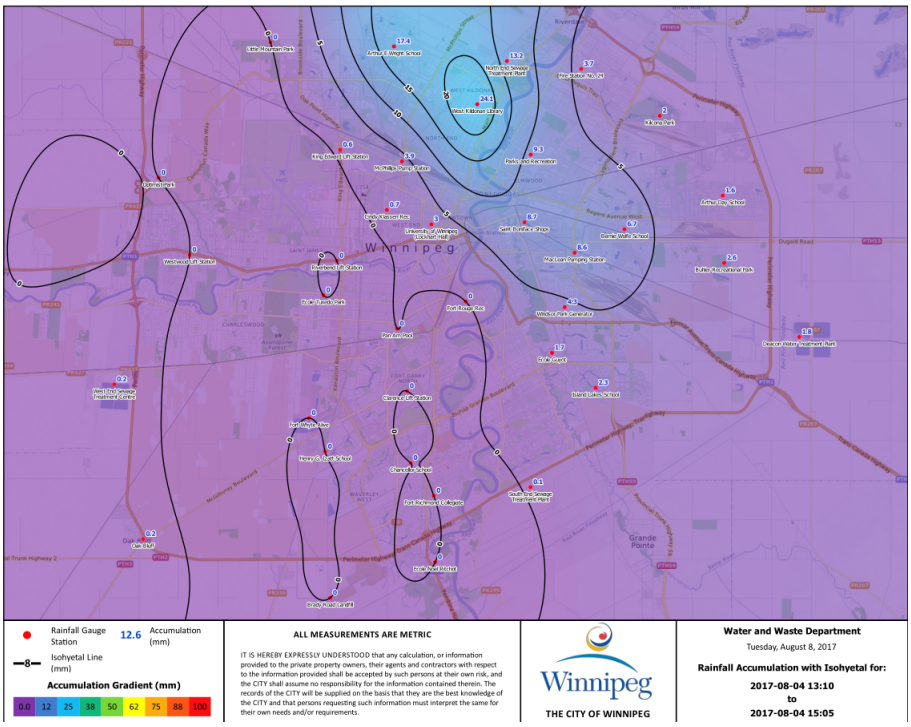


FIGURE 2. An example of rainfall map from Water and Waste department

The raw data of the rainfall amounts is not available, leading to a silo problem (this data cannot be directly integrated). To circumvent this issue, we created a tool that loads images in pdf format directly from the Water and Waste department website, converts them to png format, and uses machine learning to to reverse-engineer the rainfall amount.

The heart of determining the rainfall amounts from an image amounts to subdividing the image into a grid of smaller squares (see Figure 3 for an example); this is a configurable option of the tool. Then we remove certain features that have nothing to do with rainfall amounts: this includes the description and information block at the bottom of the image and the isohedral lines laid over the image. To perform this, we tie rainfall location amounts to rainfall gauge stations only; thus, only squares in the grid containing a rainfall gauge station are counted (the rest of the map is extrapolated anyway). To remove the isohedral lines, the colour black is consistently used for drawing the lines and never used for indicating rainfall amount. So we simply omit certain ranges in the RGB spectrum. We use the Open-CV https://opencv.org/ library for computer-vision to transform an image into a 2-dimensional array of RGB-values. We use the SciPy library https://www.scipy.org/scipylib/ to apply the k-means algorithm to each square in the grid and determine the k most dominant colours in that grid. For example, for the grid square in Figure 3 and $k = 2$ we get the two most dominant colours as a light-purple and a light-blue.

FIGURE 3. An example of a small square obtained from Figure 2

We then select the most dominant colour by applying a k-clustering algorithm (also from the SciPy library) to the k-means, and selecting the colour that has the largest cluster around its centre. This way we obtain the colour value that we associate to the current grid square. As the images produced by the Water and Waste department do not use a consistent colour scheme for rainfall amounts, we allow passing the colour encoding as a parameter, and then encode the different colour schemes used to allow determining rainfall amounts from a colour. For a piecewise linear colour gradients, we include a utility to automatically invert the colour scheme. Finally, given a mapping of dates to colour schemes, we process a range of dates to produce rainfall amounts over the city by location and date.

Since this data is now inherently made spatiotemporal, it may be readily added to our dataframe, though we haven't done this yet. It is worth noting that in the case of archived data, where the source data is not kept and only artefacts (such as rainfall maps) are kept, the technique of detecting specific features by combining the OpenCV library and a tool such as k-means can be reused. This technique may also be extended for non-specific features (such as shape of rainfall density over the city) by using neural networks.

2.4. **Moisture data.** While rainfall amounts are indicative of mosquito behaviour, soil moisture is seen to be more important for directly modelling mosquito populations. Rainfall does not directly translate to soil-moisture, as dry soil has different absorption properties compared to already moist soil.

One of the European Space Agency's (ESA) public projects is satellite data for soil moisture (and oceaninc salinity) https://earth.esa.int/eogateway/missions/smos. The soil-moisture data is provided in a general format though not immediately accessible for automated processing (it's accessible by ftps, and easy to access manually). To support obtaining soil-moisture data tied to geographic locations, we created a tool to automatically access the data and extract the soil-moisture data required. The only thing that is needed to run the tool is an ESA account (the sign-up form can be currently found https://eoiam-idp.eo.esa.int), and the desired start and end dates for obtaining the data. We automatically login to the ESA SMOS repository, access "level 2" science products (this includes the soil-moisture data), and download the data for the selected date range. The SMOS products contain soil-moisture data for nearly the entire planet, and includes earth-explorer data files that create global visualisations of soil-moisture (see Figure 4)

However, the SMOS data also includes raw data. We access this by downloading the NetCDF https://www.unidata.ucar.edu/software/netcdf/ formatted data, and then automatically filtering and extracting the soil moisture amounts by location.

This also makes the SMOS data fit consistently with our spatiotemporal dataframe. While we have not integrated the tool to update the dataframe, with the tool itself complete, this could be readily completed. As the processing we perform on this data is involved, we can

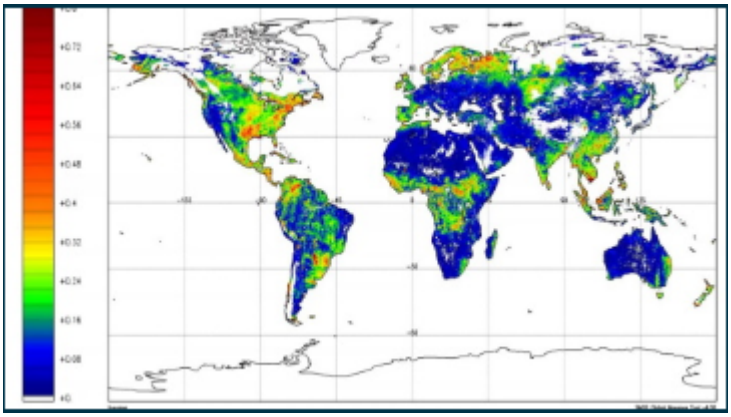


FIGURE 4. An example of soil moisture map from SMOS

also make the virtualized data faster to use by locally caching the SMOS data and preferring cached data if it exists. Thus, after a first run, accessing this data should be fast and reliable.

Much of the processing required is already performed by the SMOS project, including neural networks for determining soil moisture from the raw satellite image data. This could be seen as disadvantageous if one desires using a neural network that allows for more accurate moisture amounts to be determined using more locally available contextual information. However, the SMOS project does provide the raw satellite data in their "level 1" products section, and integrating these data points for more fine-grained analysis is feasible.

3. Identifying the most significant features for mosquito population

In order to determine the relationship between the acquired weather data, and the available data from helicopter spraying program and mosquito count from traps in 28 parts of the city, we have made predictions based on a linear regression model in three phases.

3.1. **A sight from regression analysis.** Once we had pre-processed all the gathered data from our datasets and the daily weather information, we ran a linear regression model to see the effect of current day weather information including precipitation, temperature, cloud cover, humidity, and wind speed on average trapped mosquito count across all over the city for each year from 2015 to 2021, as well as all years to determine the significance of each feature.

In order to check the importance of each feature in our regression model and to prevent multicolinear features, we took advantage of a statistical tool called "Variance Inflation Factor" or VIF, which provides an index to reflect the increase of variance of an estimated regression coefficient due to multicolinearity. We have dropped the features with VIF > 10 as it may not add to the descriptive power of our model.

Through this step, we essentially drop the most correlated features and identify most relevant features as the minimum temperature, cloud cover and precipitation.

This initial linear regression model was a foundation for developing further regression models through our analysis; in spite of the fact that it was not considerably successful at making highly accurate predictions for the average mosquito count in the city.

3.2. **Seven-days window of features.** After developing our first working linear regression model, we decided to consider a window of seven days of weather conditions to predict mosquito count. We also localised our model so that we could make predictions for each individual location of the city instead of the average mosquito count city wide. In this model we also included the helicopter spraying of chemicals for the past seven days.

These considerations led us to build a model with 26 features including helicopter spraying, precipitation, cloud cover, and minimum temperature for the past seven days.

The resulting model is able to explain more than 90% of the variability of the data with a better accuracy for all regions. However, there was a risk of overfitting, slightly higher VIFs for some features, and auto-correlation of features during the seven days period are serious drawbacks of this model.

For example, the model for the region NW1 in the year 2020 is making predictions for mosquito count using the above-mentioned features and provides an explanatory power of 96%. The evaluation of the targets and predictions using the model with 26 features is shown in Figure 5 to demonstrate a visual interpretation of prediction accuracy and it helps us in upgrading our model in making better predictions.

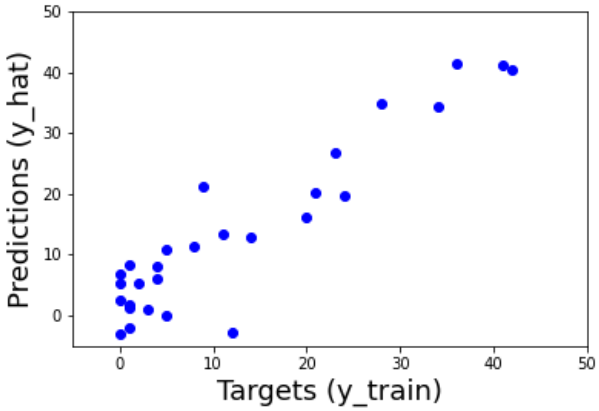


FIGURE 5. Targets vs. Predictions for mosquito count - distribution around 45° line as an illustration of model accuracy. Second linear regression model with 26 features and 96% explanatory power for NW1-2020.

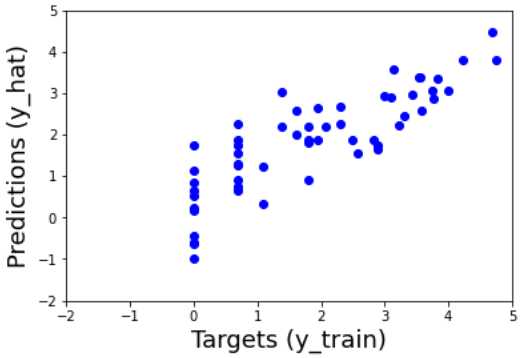
3.3. **Picking the best describing and most significant features.** In the last step of improving our linear regression model we narrow down our focus on only ten most important features including the minimum temperature, cloud cover, and precipitation for the previous day, as well as helicopter spraying in the past seven days. We also consider log of mosquito count as a target variable. These features have the least VIF scores, representing their least multicolinearity, and provide us with an improved balance of predictive power and explanatory power.

As per our analysis, we obtain an explanatory power of more than 70% for most regions using this model. While this model yields to a relatively strong explanatory capability, it had pretty low prediction accuracy. In each region of the city as this model has shown an accuracy

of 30% to 40% in making predictions. However, this linear regression model robustly shows the relationship between the best describing features and the target and makes it available for us to build stronger models.

	Features	Weights
0	Min_T_(-1)	0.951498
1	Precipitation_(-1)	0.168455
2	Cloud Cover_(-1)	-0.190795
3	NW1_h_(-1)	0.120021
4	NW1_h_(-2)	-0.006794
5	NW1_h_(-3)	0.109404
6	NW1_h_(-4)	0.074084
7	NW1_h_(-5)	0.221765
8	NW1_h_(-6)	0.184849
9	NW1_h_(-7)	0.192217

(A) Coefficients table



(B) Targets vs. Predictions - log mosquito count shown - distribution around 45° line as an illustration of model accuracy.

FIGURE 6. Final linear regression model with 10 most significant features and 71% explanatory power for NW1-2020.

In Figure 6 we illustrate our results using this model for station NW1 in the year 2020 where we obtain a 71% of explanatory power with the provided ten features and their corresponding coefficients in Figure 6a, but making accurate predictions is what we need to improve for this case as it is shown in Figure 6b.

4. Random Forest classifier for windowed, mosquito threshold predictions

We use Random Forest Classifier (RFC) to obtain a predictor for $trap\_loc[date] > 25$ for each of 28 trap locations.

RFC is a classifier that is based on decision tress. In a decision tree classifier, we want to decide whether $trap\_loc[date] > 25$, based on a series of simple yes/no questions. For example, we may ask if minimum temperature exceeds $15°C$, or if there was more than 1in of precipitation to determine if mosquito trap count exceeds 25. The problem with a decision tree classifiers is that they tend to overfit data by having large depth. Hence decision trees will likely perform well on training data, but might underperform on testing set.

One way to prevent overfit, is to consider a random forest: an ensemble of tress that use random subset of features and a random subset of training data. An oversimplified view of random forest is that we have many 'expert' trees that can fit a given data really well, but to each 'expert' we disclose only some portion of features and some portion of data. As a final prediction we take a majority of all of our many 'expert' decisions.

We use weather data (minimal temperature, precipitation, humidity) and trap count data as initial data for a given location. We use $\sim$ 1000 data points (years 2015-2021). We use back-filling for missing entries in trap count data.

Our classifier has 24 features as input: minimal temperature, precipitation, humidity and *trap_loc* over 6 days window (from date-2 to date-7), and target is a boolean $trap\_loc[date] > 25$.

We use 20 trees in each forest and 5 features per each tree. The split between training and test data in 70%/30%. We used RandomForestClassifier provided in sklearn Python library.

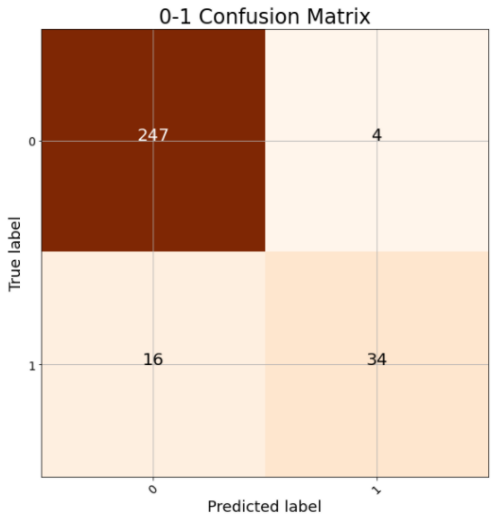


FIGURE 7. Confusion matrix for NW1 location

For illustration purposes, the confusion matrix of performance of RFC on testing data for NW1 region is presented in Figure 7. This matrix illustrates performance of our model on test data, with off-diagonal numbers indicating misclassified instances (true-negative and false-positive). We use accuracy and precision to estimate RFC performance. From the matrix in Figure 7 we deduce that RFC for NW1 location has accuracy 0.93 and precision 0.89.

loc	acc	prec	loc	acc	prec	loc	acc	prec	loc	acc	prec
NW1	0.93	0.89	NE1	0.89	0.79	SW1	0.96	1.00	SE1	0.96	1.00
NW2	0.95	0.95	NE2	0.92	0.89	SW2	0.90	0.84	SE2	0.94	0.90
NW3	0.97	1.00	NE3	0.95	1.00	SW3	0.91	0.93	SE3	0.94	1.00
NW4	0.90	0.84	NE4	0.99	1.00	SW4	0.94	1.00	SE4	0.93	0.84
NW5	0.96	1.00	NE5	0.95	1.00	SW5	0.98	0.00	SE5	0.92	1.00
NW6	0.96	1.00	NE6	0.92	1.00	SW6	0.95	0.96	SE6	0.99	1.00
NW7	0.93	0.90	NE7	0.94	1.00	SW7	0.99	NA	SE7	0.93	1.00

TABLE 1. Accuracy and precision of RFC for different locations

In Table 1 we list accuracy and precision of RFC for all 28 trap locations. We note that for SW5 and SW7 locations there were less than 1% of data points with $trap\_loc[date] > 25$

(for other regions this value is above 10%), hence high accuracy, but low precision for these locations. For these two locations we suggest using lower thresholds (15, or 10) for *trap_loc*.

For a future work, our classifier can be modified to obtain a predictor (i.e. use classifier for a tens digit of mosquito count) for mosquito population. Additionally, we do not use helicopter spraying data, as the corresponding dataset is to small, however ground treatment data is extensive and could further improve performance of the model.

5. Next steps and possible future studies

Within our analysis on predictive modelling of mosquito population in Winnipeg, we tried a number of ways to approach this problem from different perspectives and our team has noticed that there might be a possibility to work on this project further.

City data was slightly disorganised, for instance we were pulling weather data from external sources, instead of using internal data (for example, Water and Waste department has precipitation data, but we were not able to obtain it). Even for the datasets available, the fields did not always match and we had to do some data patching. We identified critical steps to design a more efficient city data portal with more harmonised data frames. For instance, used for city portal, our methods allow to request all available data within a certain radius of a given geolocation.

Also, having a control study (with no spraying) on mosquito larval development in Manitoba can potentially make an improvement in our mosquito population models. Such a "pure" model for a mosquito population growth can help us with filling out the gaps and missing values in our data more accurately.

By incorporating time series analysis we can make predictions based on mosquito population patterns and eliminate the auto-correlation of the features and targets, including weather data and trapped mosquito count.

Gaussian Process Regression would be another useful approach for this project that allows a regression to take into account prior knowledge and perceived conditional probabilities to describe the future and its uncertainty.

6. Acknowledgements

To Allen, Kristine and Ruth: thank you for organising this workshop. Thanks to Julien Arino for providing useful references. And a special thanks to the ICB branch staff: Scott, David, Ken and Jennifer.

References

[1] K. Fryer, J. Antony, and A. Douglas, *Critical success of continuous improvement in the public sector: a literature review and some key findings*, TQM **19** (2007), no. 5, 497–517. ↑2.1

[2] B. Gleason and M. Rozo, *The silo mentality: How to break down the barriers*, Forbes (2013Oct). ↑4

[3] A. study group on functional organization, *Organizational renewal: Tearing down the functional silos*, Association for Manufacturing Excellence, 1988. ↑2.1, 2.1

Email address: andrew0arman@gmail.com

Email address: jonathan@infinitylab.io

Email address: aidinzp@gmail.com