| Problem Chosen | 2021 MCM/ICM Summary Sheet | Team Control Number |
| --- | --- | --- |
| C | | 2109976 |

# Confronting Vespa Mandarinia: Neural Network Based Prediction and Classification Models

**Summary**

We construct two neural-network-based models to resolve the problems of Vespa mandarinia spread situation and identification of the pest by images. We make several hypotheses to simplify the issues. For instance, we assume that the sightings are related in both dimensions, time and space, to make neural network applicable for the problems. Also, the living conditions are considered as similar to the natural pattern as the past to avoid outside disturbance. After investigating several models, we finalize that the long short-term memory (LSTM) model is to solve the problems of prediction while another model based on deep neural network (LSTM) is to solve the issue of classification.

Firstly, LSTM is a special branch of recurrent neural network (RNN). Traditionally, RNN is used to solve short-term dependencies problems. Its performance is not good when dealing with long-term dependencies problems which characteristics long time lag connection. The prediction of spread situation of Vespa mandarinia is a typical long-term dependencies problem. There are thousands of discrete reports of sightings distributed unevenly in a time span over one year. Consequently, we choose LSTM instead of RNN to resolve this problem. Then we divide the useful data into two main groups. The larger part is used as train set while the other one is used as testing set. Then LSTM model is developed to able to predict the spread situation in the future.

Furthermore, confronting the problem of classification of images, the first thought we have is to use logistic regression. It is because that the result of classification of images can only be two, it is or is not a Vespa mandarinia. As a consequence, the statistics will follow a binomial distribution. Then, what we consider is the classification problem. Neural network is a common method to identify and classify images with a lot of parameters. Due to the quantity of given pictures, we choose to use a deep 5-layer-neural network. With the pixels as input, the model will learn the characteristics of Vespa mandarinia in the pictures and will be able to classify a new image by judging if there are Vespa mandarinia within.

In conclusion, our models can be of use in predicting the spread situation and classify images by the existence of Vespa mandarinia. Moreover, by processing the results of the prediction model, we can obtain the knowledge of the distribution of locations of potential positive ID sighting. Such information will be of help in increasing the working efficiency of the relevant government department by prioritizing sightings with higher possibility to be positive ID. The dominant advantage of our model is its high accuracy, which is nearly 90% for the classification model. Though there are still some weaknesses such as the relative slow computing speed, there are lots of space for our model to be improved and updates.

**Keywords**: LSTM, Deep neural network, Image identific

# Confronting Vespa Mandarinia: Neural Network Based Prediction and Classification Models

February 9, 2021

## Contents

# 1 Introduction

## 1.1 Problem statement

Vespa mandarinia was witnessed in Washington State since the discovery of the pest in the adjacent region of Canada in September 2019. In total, 4440 sighting incidents took place in the following year. Only 14 sightings was confirmed as witnesses of Vespa mandarinia while thousands of cases were mistaken or cannot be verified without adequate information. Therefore, a problem arose. Compared to the numerous reported sightings, the state government agencies with limited power may not draw timely validated conclusions to the incidents. Moreover, the lack of efficiency in identifying the right species would lead to a imprecise or useless model for the spread of Vespa mandarinia.

The reason why the government needs to identify the Vespa mandarinia and figure out how it spreads in the district is that it is harmful for the local species, especially honeybees. Vespa mandarinia is also known as the Asian giant hornet. More specifically, we learn that it is the largest species of hornets across the world, originating in the tropical region of eastern Asia. As an alien species to the state, Vespa mandarinia not only occupies some local natural resources, but invades the native European honeybees. Its invasion activity reaches the peak in September and October when the population of a single colony also becomes the largest in a year [5].

In order to fulfill the requirement, we developed a model that can predict the spread situation of Vespa mandarinia based on reported sightings. Then with the given images, we derived a model to predict the correctness of classification. Finally, our model can be further improved with increasing sightings and be of use in reality.

## 1.2 Assumptions

### 1.2.1 Assumptions for the Prediction Model

- All the sightings that are positively identified are relevant in both dimensions of time and space.
- The cases before 2019 can be considered as irrelevant to the discovery of Vespa mandarinia in Canada and the Washington State in 2019 and 2020.
- The Vespa mandarinia found in the Washington State from 2019 to 2020 came from one incidents of invasion of alien species.
- Since the time when Vespa mandarinia was first found in the Washington State in 2019, the biological habits of the pest, including the breeding habits, retained unchanged.
- The unverified sightings are considered as positive identification.

### 1.2.2 Assumptions for the Classification Model

- Vespa mandarinia can be identified by its appearance.
- From 2019 to 2020, the appearance of Vespa mandarinia in the district is inhabited stably.
- The species of Vespa mandarinia in the district is only one kind and each has alike appearance.
- The local species have not developed highly similar appearances as Vespa mandarinia.

# 2 Model Construction

## 2.1 Prediction Model

### 2.1.1 Brief of Long Short-term Memory

Long short-term memory (LSTM) was developed to solve the long-term dependencies problems that usually have extended time lags in recurrent neural network (RNN). Take the identification of image as an example as an instance, ordinary RNN can predict that one pixel is blue if and only if a number of surrounding pixels are all blue, while LSTM can predict the color of a pixel by the information obtained previously or even far ago. For example, if the previous identified part of image has the mirror characteristic, LSTM model can predict that the next pixel may have the same color with the corresponding pixel by symmetry. LSTM performs well when dealing with long-term dependencies problems. Consequently, LSTM is the model that we use to predict the spread situation of Vespa mandarinia.

### 2.1.2 Notations

The notations used in the following part are listed in the table Tab. 1.

| Symbol | Definition | Comments |
|---|---|---|
| $f_{\mathrm{t}}$ | the forget information determinant | |
| $\sigma$ | the sigmoid neural network | varies from 0 to 1 |
| $W_f$ | the slope of regression | similar to $W_i$ and $W_o$ |
| $h_{t-1}$ | the signal from the previous block | |
| $x_t$ | the input information at current block | |
| $b_f$ | a linear coefficient | same as $b_i$ and $b_C$ |
| $i_t$ | the input information determinant | |
| $\tilde{C}_t$ | the candidate vector | |
| $o_t$ | the output information determinant | |
| $h_t$ | the signal that generates in the current block | |
| $C_t$ | the current cell status | |

Table 1: Notations

### 2.1.3 Structure of LSTM

Due to the special settings of LSTM blocks, this model is suitable for processing and predicting the data in a time sequence that are separated extensively. Here is the scheme of LSTM fig. 1.

The upper horizontal line is the kernel of LSTM, the cell status. It's like the carrying belt in a factory. The goods it carries is information and the machine arms that add or take the goods away are called gates. The gates are composed of a sigmoid layer, which can output a number varying from 0 to 1, proportional to the amount of information that is allowed to pass, and a bitwise multiplicative operation. There are in total three kinds of gates in LSTM, which are forget gate, input gate and output gate. They are used to change and protect the cell status [2].

**Forget Gate** The first and most important step of a LSTM is to determine what to forget or drop from the cell status. This function is achieved by the forget gates. For example, when identifying an image, it detects a new block of pixels with the same color, the previous color ought to be forgotten and predict the present color with the only information from the nearby area. When the computer process this, it computes the value of $\sigma$ by

$$f_t = \sigma(P_f \cdot [h_{t-1}, x_t] + b_f). \tag{1}$$

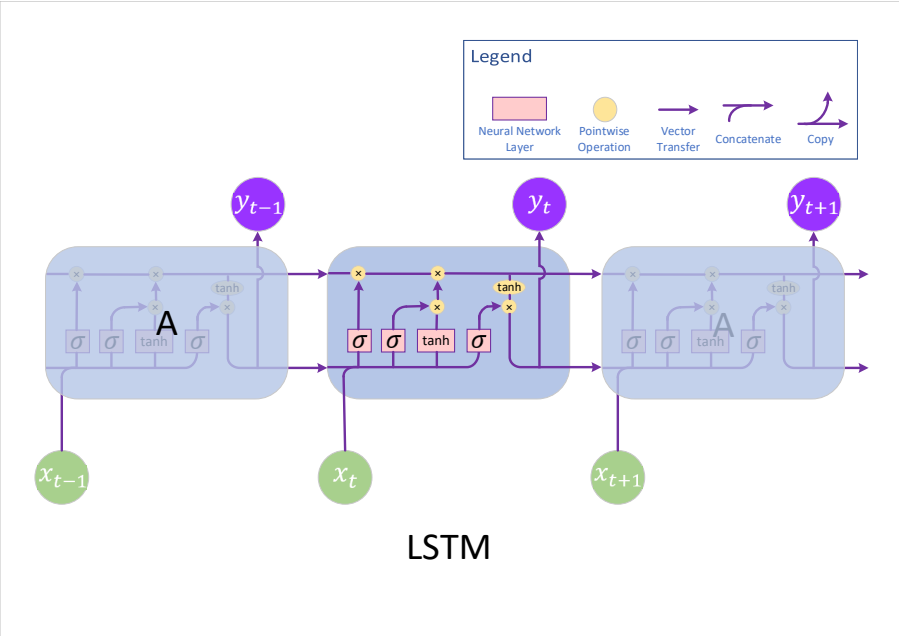


Figure 1: Basic Structure of LSTM.

Then a value varying from 0 to 1 will represents the amount of information that is retained. 0 means totally discard while 1 implies retain all.

**Input Gate** The next step is to determine what to add to the cell status. The sigmoid neural network will determine what values to update and a $tanh$ layer will create a new candidate vector. Then, the generation of such information will cause the update of the cell status. Still the case of image identification. The color of nearby pixels should be added to replace the previous color and then the prediction can be made. This update process can be described by the following formula.

$$i_t = \sigma(P_f \cdot [h_{t-1}, x_t] + b_i). \tag{2}$$

$$\tilde{C}_t = \tanh(P_C \cdot [h_{t-1}, x_t] + b_C). \tag{3}$$

After updating the information, it should be added to update the cell status.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

**Output Gate** After updating the cell status, the block will output a value based on the current cell status. This process is done in two steps. First, we run a sigmoid layer to determine which part of the cell status is to be output. The result of this step is calculated by Eq. (5). Then, $o_t$ is multiplied by a tanh. Finally, the output signal is determined by Eq. (6).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o). \tag{5}$$

$$h_t = o_t * tanh(C_t) \tag{6}$$

From the analysis above, we learn that LSTM is suitable for predicting the spread situation of Vespa mandarinia, which is a typical long-term dependencies problem.

### 2.1.4 Introduction of LSTM to the Real Case

The real case we were to deal with is to predict the spread situation of Vespa mandarinia with given sightings in 2019 and 2020, merely with 14 positive identification cases reported out of 4440 cases in total. With such a complicated data set, the first thing we did was the data cleaning. All the cases that were proved not to be the sightings of Vespa mandarinia (negative ID) and those were not processed were deserted.

The data we used were those positive ID and unverified cases. Then we used a part them as the primary input of LSTM. This part of data is clustered and visualized in fig. 2. Attention should be called here because one of the ordinary points of our model was to use only the cases before October of 2020, instead of all the cases. The aim was to separate the data deliberately as a train set and a testing set. The data before October of 2020 were used to train the LSTM to predict the spread situation of Vespa mandarinia till the end of 2020.

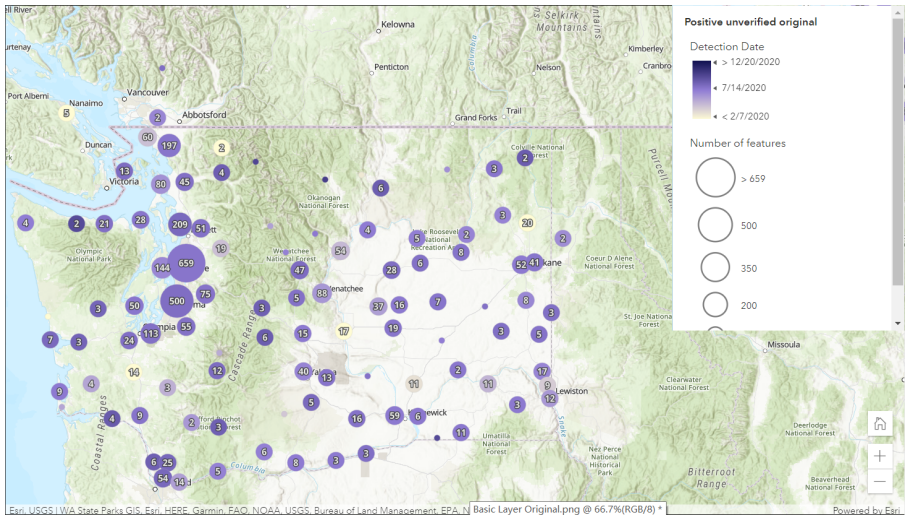


Figure 2: Original Clustered Data.

Once we obtained the prediction data, the testing set would be compared with the prediction set with feedback to the LSTM. With changing parameters, the error between the prediction set and the testing set would be minimized. The prediction results compared favorably with the testing data. At last, our model was constructed and was able to predict the future spread situation of Vespa mandarinia. More details of the results will be discussed in the Simulation Results Section.

## 2.2 Classification Model

### 2.2.1 Brief of Deep Neural Network

Given an image of a reported sighting, there are two possibilities, the animal in the image is a Asian Giant Hornet or not an Asian Giant Hornet . The assumption is that for a given picture of a reported sighting, which is regarded as in put X, the corresponding Y, which indicates weather or not it's an Asian Giant Hornet (1 for true and 0 for flase) follows a binomial distribution. To predict whether a new reported sighting is credible, we can use Logistic Regression to calculate the value of the possibility that the creature in the picture is an Asian Giant Hornet. The model can be explained by Eq.(7)

$$P(Y = 1|X) = \frac{exp(\omega X)}{1 + exp(\omega X)} \tag{7}$$

However, after applying the model we found out that the accuracy of this model is not high enough, which means we need more parameters in the model to increase the accuracy of our model.

Neural Network is often used for classification due to the large number of parameters, and also its accuracy. So we used a deep neural network with five layers to predict the possibility that a picture is an Asian Giant Hornet. From different deep neural networkds we choose the densely connected neural network. Generally, its mechanism can be described as the figure below shows.

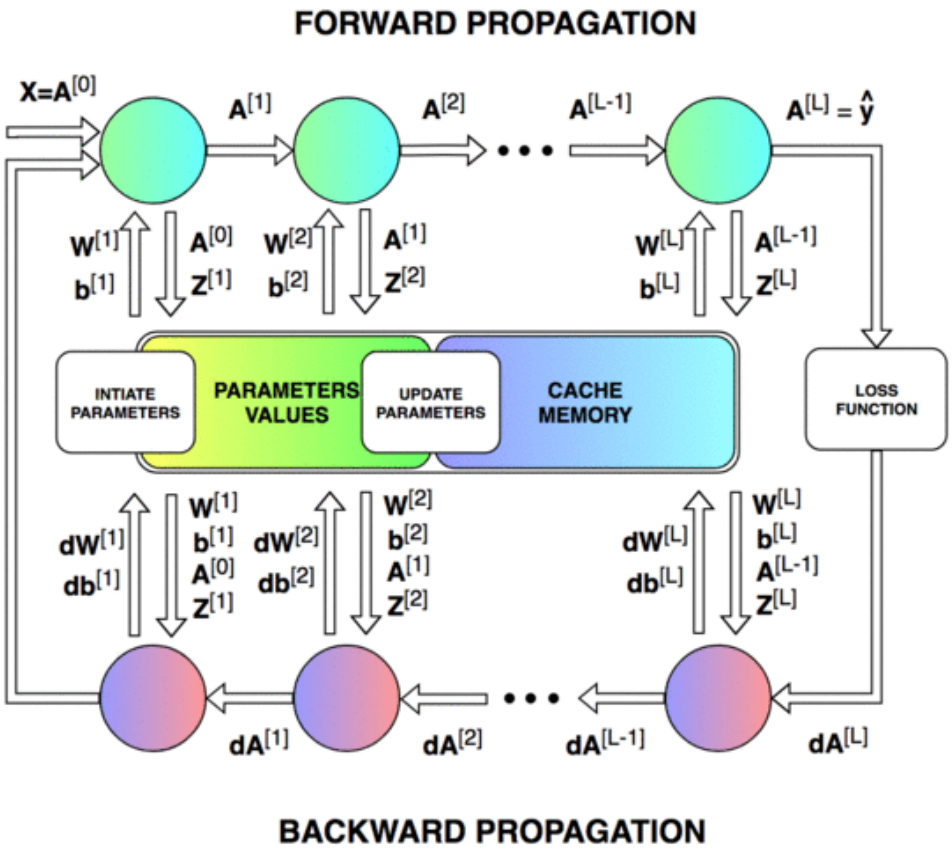


Figure 3: Forward and Backward propagation of DNN [4].

The Neural Network accept an input of an image with size m * n * 3 where n and m is the resolution of the image and 3 means the picture is of mode "RGB" and the neural network will randomly initialize the parameters and calculate the value of Loss function which is shown by Eq.(8) in each iteration, then use back-propagate algorithms to get the gradient of each parameters and update the parameters along its gradient to find the minimum of the values of loss function.

$$Loss = -\frac{1}{m}\sum_{i=1}^{m}(y * log(a^{[L]}) + (1-y) * log(1-a^{[L]})) \tag{8}$$

In this equation, y is the label of the picture and a is the input of some layer. After we finished training the neural network, for another reported sighting, we can calculate its possibility of being a credible sighting as well as predicts the likelihood of a mistaken classification.

### 2.2.2 Notations

The notations used in the following part are listed in the table Tab. 2.

| Symbol | Definition | Comments |
|---|---|---|
| $X_i$ | the i th image in the data set | |
| $y_i$ | whether the i th image is Vespa mandarinia | |
| $\hat{y}_i$ | prediction of whether the i th image is Vespa mandarinia | |
| $\sigma$ | the sigmoid function | varies from 0 to 1 |
| $ReLU$ | Rectified Linear Unit | |
| $a_i^{[L]}$ | the input in L th hidden layer for i th image | |
| $\omega, b$ | the parameters of the neural network | |

Table 2: Notations

### 2.2.3 Structure of Deep Neural Network

The structure of a common deep neural network can be described by the figure fig. 5

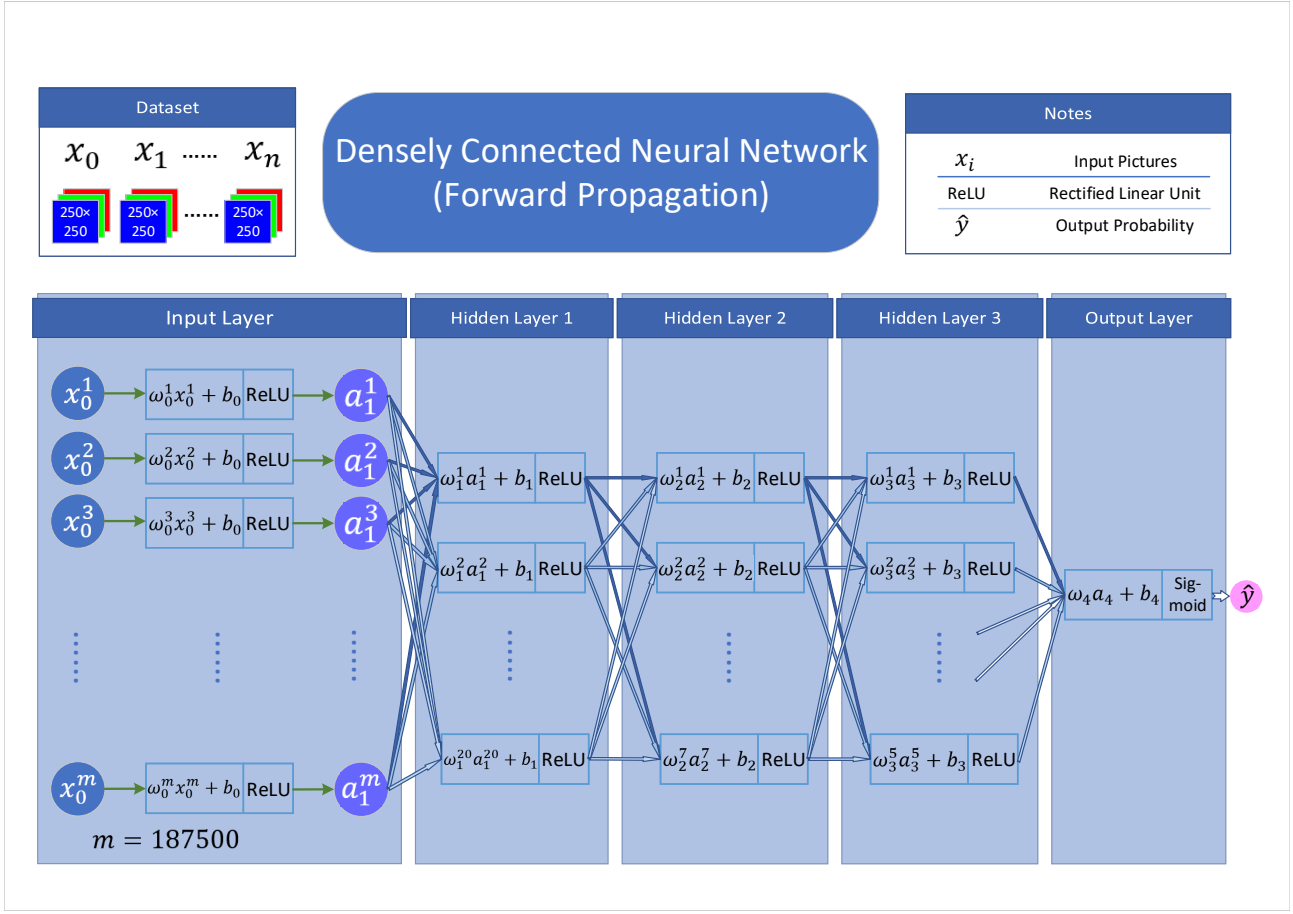


Figure 4: Forward Propagation of DNN.

In our Deep Neural Network model, there are five layers, the input layer, output layer and three hidden layers. The scale of input layer is 250 * 250 * 3 and output is 1, the reason is that we resized the pictures in the reported sightings to 250 * 250 and each picture can be splited into three matrix representing the value of R, G, and B, and the output is just the possibility with size 1. The size of the three hidden layers are 20, 7, 5 separately. The structure of the deep neural network is shown in the figure fig.5

Given the input $X_i$, which is the i th image of the training set, we can calculate the $z_0^1 = \omega_0^1 X_i^1 + b_0^1$ where $\omega_0$ and $b_0$ are parameters of input layer and then apply ReLU function to all the linear unit z, which is $a_1^1 = ReLU(z_0^1)$ to eliminate the part of output of the first layer that is negative. After that do the same thing to the output of the input layer, then get the out put of hidden layers 1, 2, and 3. For the last layer which is also the output layer, the activation function is still the one we used before

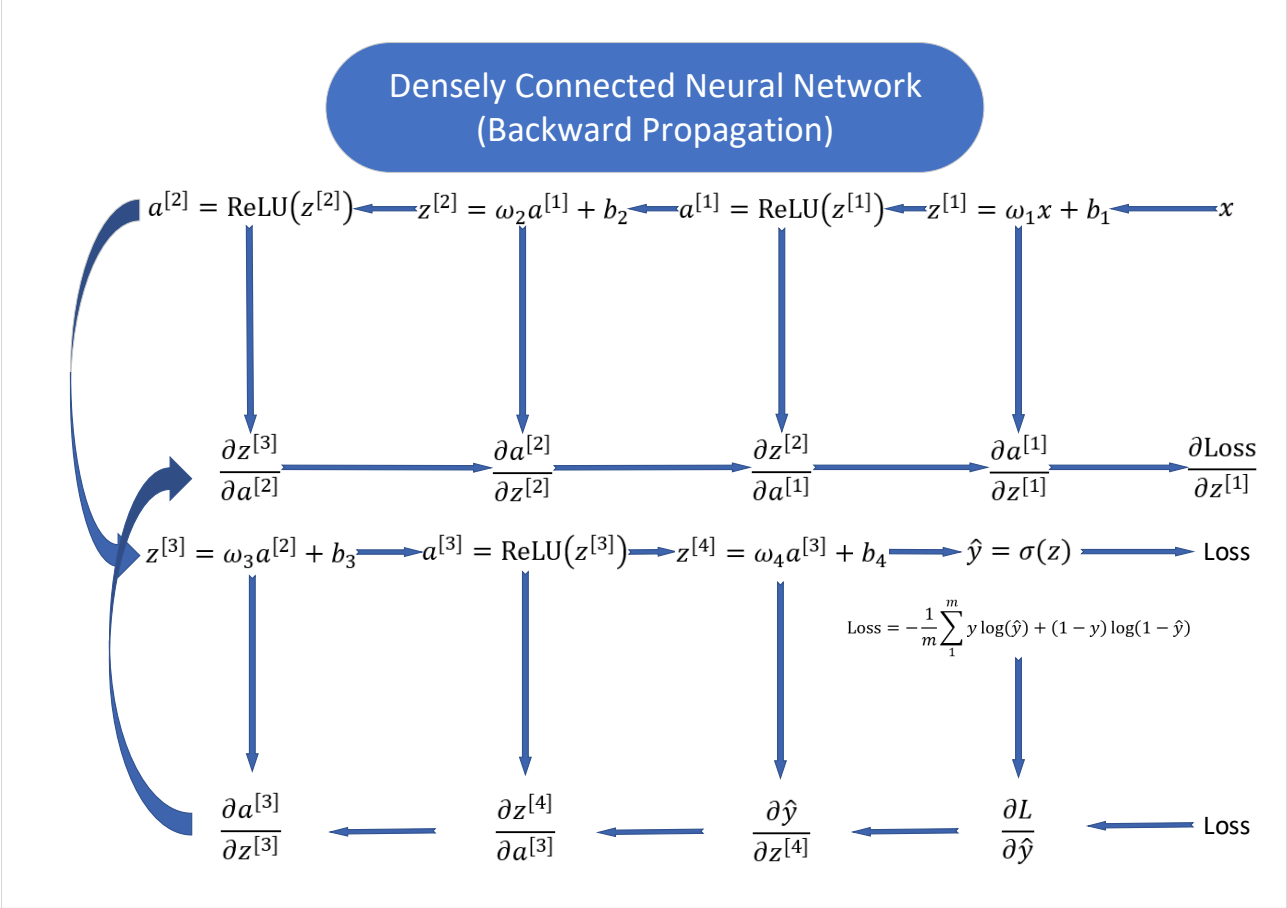


Figure 5: Backward Propagation in DNN.

- the simoid function to get a possibility, which is the likelihood of a mistaken classification.

The process of running the deep neural network:

1. randomly initialize the parameters of the deep neural network
2. take the input then calculate the Loss function
3. use back propagation to calculate the gradient of the parameters in the model
4. use gradient decent algorithm to decrease the parameters along its gradient and update them in the network
5. repeat the procedure until the loss function is small enough

The backward propagation is illustrated by fig. **??**. After the forward and backward propagation we will get a series of $\omega$ and $b$ to represent the neural network. Then try to use this model to classify a picture, the result $\hat{y}$ will be the possibility that the picture is a Vespa mandarinia.

### 2.2.4 Introduction of Deep Neural Network to the Real Case

To use the Neural Network model to predicts the likelihood of a mistaken classification, we choose to use the images provided in the dataset as input X, with some photos of Vespa mandarinia found from the Internet [5] [1] [6] [3], We used approximately 500 pictures with negative ID, and 200 pictures with real Vespa mandarinia as the training set, and 60 pictures as test set. After running the forward and backward propagation algorithm for 5000 times of iterations at a learning rate

$\alpha = 0.0025$, the Loss decreases from 0.6075114409935868 to 0.08984676281165943. The cost function with respect to number of iterations is shown in the figure fig.6

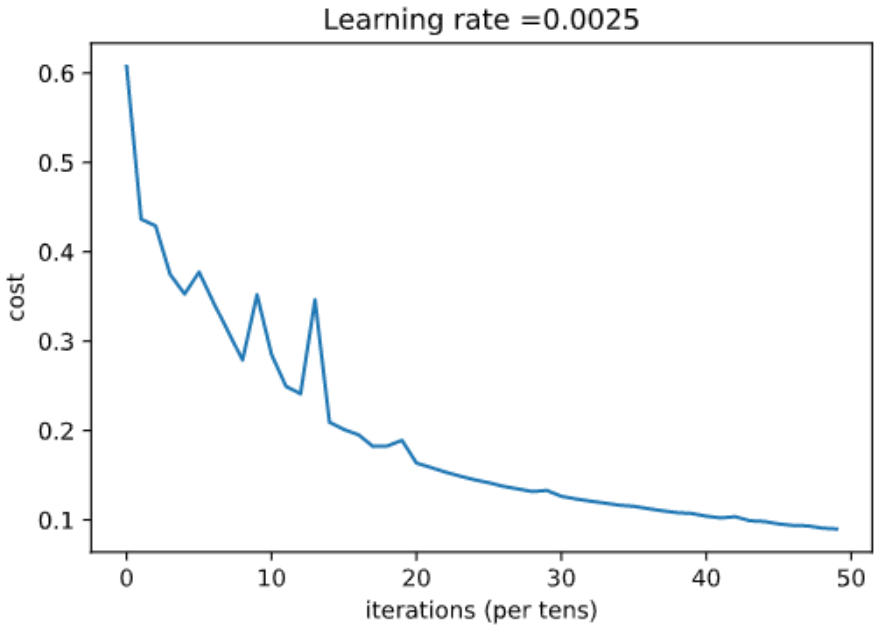


Figure 6: the curve of cost function with learning rate 0.0025.

By applying the model, we can predict the likelihood of a mistaken classification of the test set. The accuracy of this model to predict training set is 0.99857 while for the test set it is 0.80000. The result is satisfactory to predict the likelihood of a mistaken classification.

# 3 Simulation Results

## 3.1 Problem1: The Prediction of Vespa Mandarinia's Spread Situation

### 3.1.1 Forecast of Vespa Mandarinia's Spread Situation after September in 2020

Although the spread situation of Vespa Mandarinia has been given, our model still did this part of simulation for testing purpose. And the results compared favorably with the practical ones.

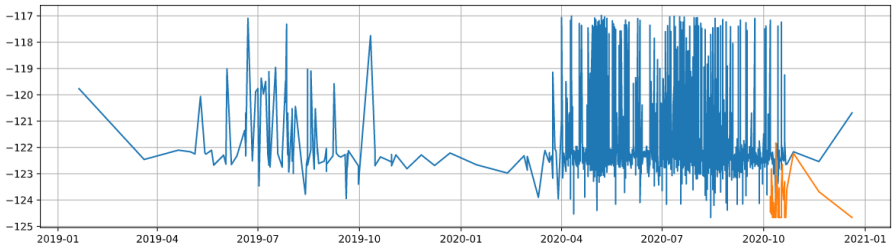


Figure 7: Forecast Data for Longitude.

Both figures show the original data and the forecast ones. The blue line represents the train set while the orange line shows the prediction results. Specifically, fig.7 gives the information with respect to longitude and fig.8 gives the information about the latitude. The value for the lines are provided in the appendix, by tab.4 respectively. From the figures, one can clearly tell that the forecast line matches the line that represents the train (reality) set with an acceptable error. This high-level correspondence implies the reliability of our model.

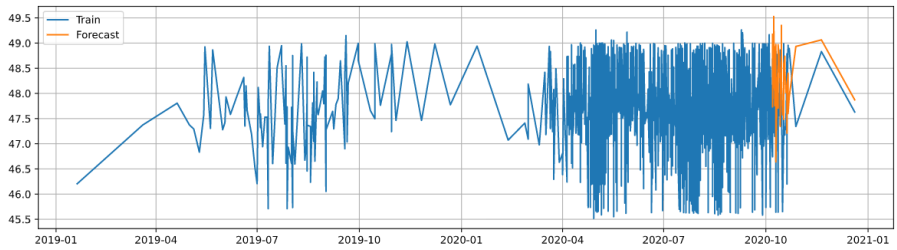


Figure 8: Forecast Data for Latitude.

### 3.1.2 Forecast of Vespa Mandarinia's Spread Situation in the Future

Due to the limitation of the amount of given data and time, we predict 50 possible upcoming positive sightings and their locations with respect to longitude and latitude. In order to visualize our results, we made a map (9) with all the positive and would-be positive locations marked as points.

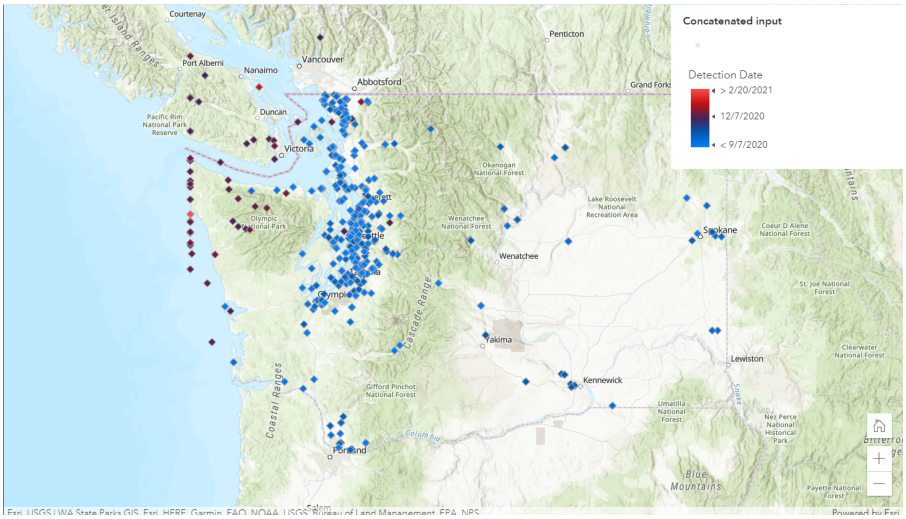


Figure 9: Forecast Data.

### 3.1.3 Level of Precision

This part is aimed at calculating the precision by terms of area in two dimensions, longitudinal and latitudinal. According to *University Physics with Modern Physics (14th Edition)*, the radius of the earth is $R$, 6370 $km$ [7]. Our assumption is that: the earth is a perfect sphere. Since the longitudinal and the latitudinal data have a precision of $10^{-3}$, we use the 0# prediction data as an example. The latitude is $49.174°$ and the longitude is $124.466°$.

The radius of the circle whose center passes the earth's axis and perpendicular to it can be calculated by Eq. (9)

$$r = R * sin(49.174°) = 4820km \tag{9}$$

So the precision of longitudinal length is calculated by Eq. (10)

$$p_{long} = r * 10^{-3} = 4.82km \tag{10}$$

And the precision of latitudinal length can be calculated by Eq. (11)

$$p_{lat} = R * \frac{2 * 49.174° * 10^{-3}}{360°} = 1.74km \tag{11}$$

The precision results are summarized in the table Tab. 3

| Item | Precision |
|---|---|
| Longitudinal length | 4.82 [km] |
| Latitudinal length | 1.74 [km] |
| Area | 8.388 $[km^2]$ |

Table 3: Precision of the Prediction Model

### 3.2 Problem2: The Prediction of the likelihood of a mistaken classification

As we mentioned in the model of Deep Neural Network before, we use the method of logistic regression and a neural network to predict the likelihood of positive sightings and use one minus that possibility will give the likelihood of a mistaken classification. To fit our model, a preprocessing is needed for a given picture of reported sighting.

We need to first transform the picture into a 250px * 250px size picture, then use some scripts to transform the matrix into a vector of size 187500, then it can be regarded as one input of the neural network and it will give the prediction based on the parameters trained using the training set.

This way the output will be a number between 0 and 1 and use $1 - p$ it will be the likelihood of a mistaken classification (recognize a creature as a Vespa mandarinia but it is actually not).

### 3.3 Problem3: Prioritizing Upcoming Positive Sightings

Due to the sudden invasion of Vespa mandarinia, reports of possibly positive sightings flows in as a cascade. It is not practical for the Agriculture Department of the government to process the information or even give feedback in a timely manner. Notwithstanding, this may give rise to the waste of time in dealing with the pest if we cannot figure out where they are virtually.

Consequently, it's of vital importance to prioritizing the valuable reports of sightings that are more possible to be positive ID. In order to achieve this purpose, we can take advantage of both the prediction model and classification model.

To make use of the prediction model, we can mark all the predicted locations on a map and compare these points to the additional locations of new sightings. If they are very near, then the reports of sights can be processed by the government staff first.

More directly, we can use the classification model. What we need to do is to let the model process all the pictures attach to the reports and calculate the possibility of positive ID. Then rank all the cases and start working on those with higher possibilities.

### 3.4 Problem4: Update of Model with New Reports

Our models are based on neural network, which needs a great quantity of data to train. With increasing reports, our models can be further improved and updated to a higher level of precision and accuracy. That's because as the data it learns increases, it's more possible for it to deal with new data with previous knowledge and "experience". Additionally, the parameters of the models are modified after each update, making the model more reliable and accurate.

When it comes to the frequency of update, we can first see a figure fig.10, which illustrates the number of sightings in different months.

From the figure, we can tell that the distribution of sightings has a clear season-relevant pattern. There are a lot of sightings reported in the spring, the summer and the fall (from April to October) and few sightings reported in the winter. We observe that the maximum number of reports is around 1,400 per month or 47 per day, thus we can set two update conditions.

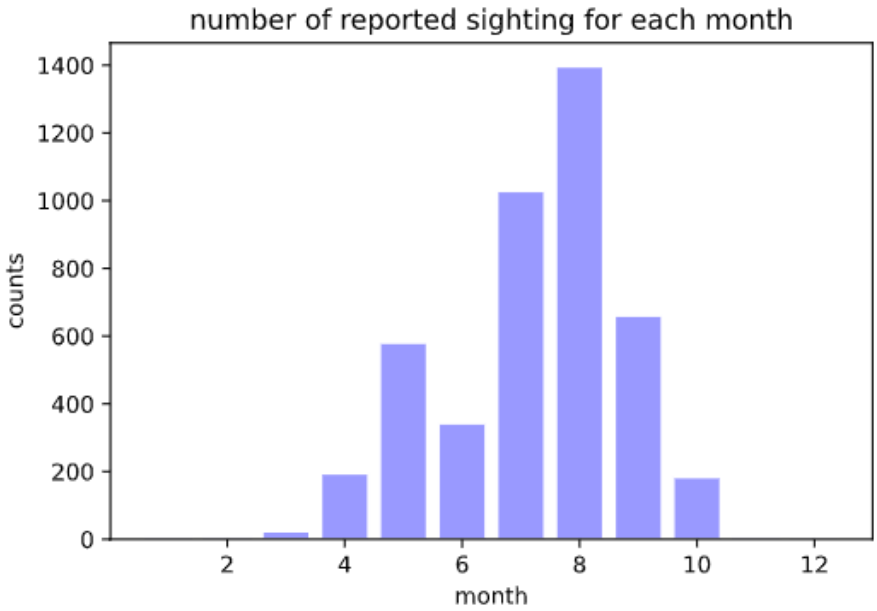


Figure 10: Distribution of Sightings in different months.

- Every 7 days (one week) past between April and October or every month between November and March next year.
- Every 100 sightings increased.

Once one of these conditions are satisfied, our model is updated one time with the data obtained during the period. Only in this way could we make conductive progress in our model under the natural rhythm without the influence of season fluctuation.

Based on this update mechanism, we may update our model approximately every two days in the worst case, which is far more time than the time needed to train our model sufficiently (with 5,000 iterations in about 1 hour).

### 3.5 Problem5: Evidence of the Eradication of Vespa Mandarinia in Washington State

By clustering the points we obtained with the prediction model, we can have an even clearer view of the distribution of new possible locations by fig.14.

From the figure, we can tell a tendency that the sightings are moving from the middle part of Washington State to the western coast of America or even arrive the ocean as time goes by. This is a solid evidence of how Vespa mandarinia spreads towards the outside of Washington State. It further indicates that less or no Vespa mandarinia is still in Washington State. In other words, it eradicated in the district.

What's more, by applying our classification model on the reports that labeled with 'Unprocessed', the results are all negative, which means that the possibility of mistaken recognizing Vespa mandarinia in the newest reports is high, It also indicates the district of Vespa mandarinia.

## 4 Strengths and Weaknesses

### 4.1 Strengths

- **Accuracy.** High accuracy is one of the dominant advantages of our model. For instance, the classification model has an accuracy of 80% for testing set and a even higher accuracy of nearly

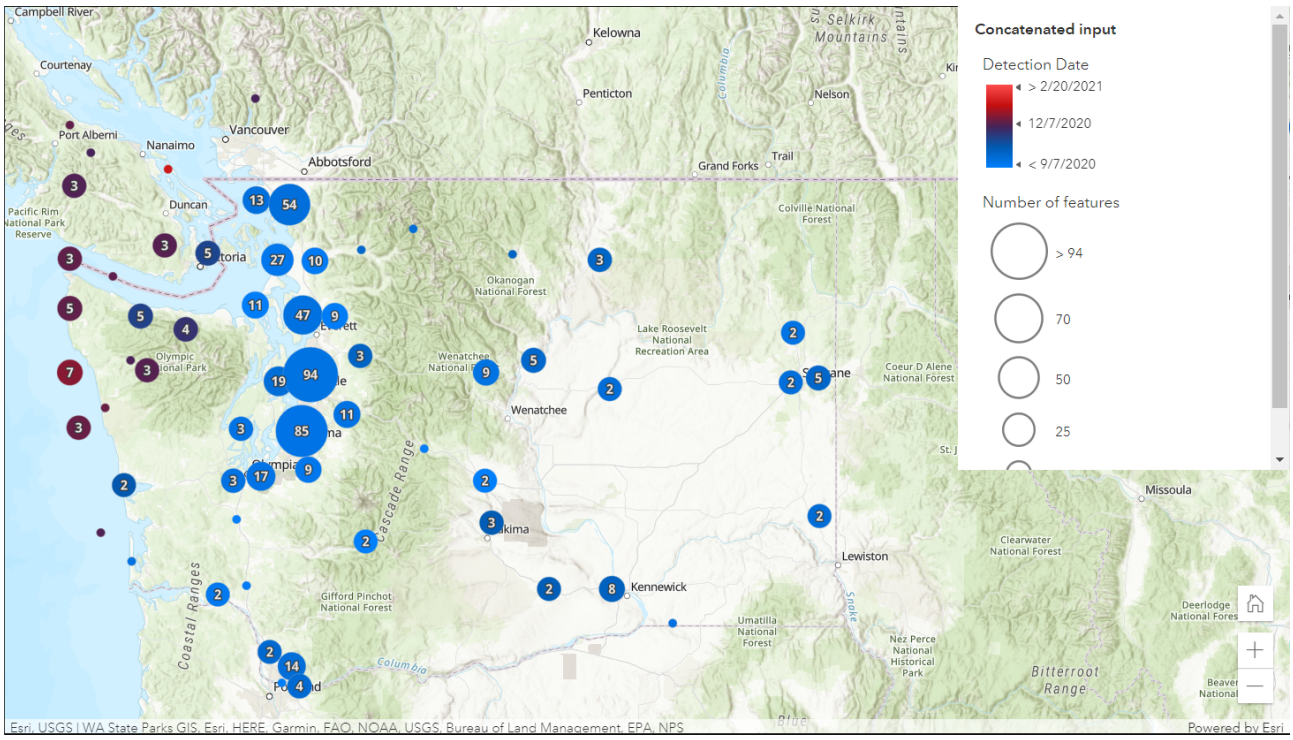


Figure 11: Clustered Prediction.

99% for train set. This is consistent with the high reliability of our model.

- **Suitability.** The LSTM prediction model suits the problem we need to solve very well. We've tried lots of other models. For example, the VAR model will only calculate the average number and output a horizontal line instead of the zigzag line created by LSTM model. The prediction made by VAR model is shown in figure fig.12 and fig.13

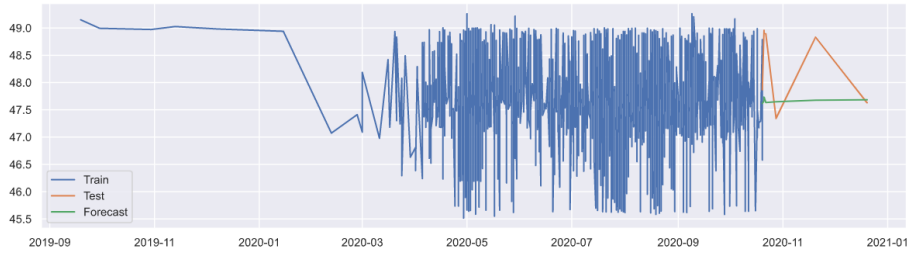


Figure 12: Latitude prediction for VAR model.

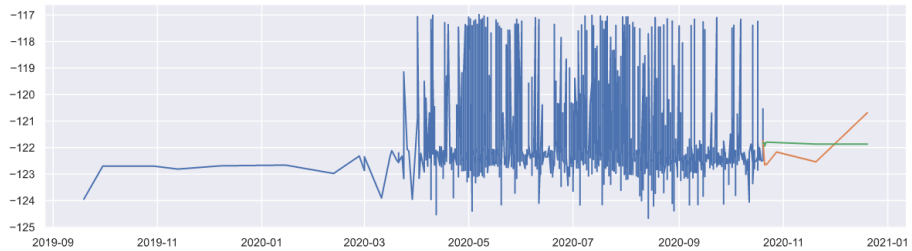


Figure 13: Longitude prediction for VAR model.

- **Stability.** Our models are very stable. This property guarantees the reliability when processing a large amount of information.

### 4.2 Weaknesses

- **Efficiency.** The computing speed of the model is relatively slow, which implies there might be some space for improvement of our algorithm.
- **Overfitting.** Potential overfitting issues happen in our classification model. We found that because the accuracy of train set is much higher than that of the testing set. The accuracy for train test is over 99 percent while for the test set it is only 80 percent, meaning that we gives too much weight for something unimportant in the train set, making the model overfitted.

## 5 Conclusions

In this paper, we build a model to predict the spread of Vespa mandarinia based on LSTM and determine its level of prediction. Then, we build a model to classify the characteristics of Vespa mandarinia based on a kind of Deep Neural Network (DNN) called densely connected neural network and determine its reliability. Finally, we apply our models in prioritizing investigation of reports and determining the eradication of the pest in Washington State and provide the way to update our models. Here we conclude our findings.

- The spread of Vespa mandarina did relate to locations and seasons. They were active in summers and inactive winters. Besides, their area of activity is moving northwest away from the Washington State towards Canada.
- To classify different individuals of Vespa mandarina is possible just according to their appearances. For a well-designed neural network model, the precision can reach 80 percent or maybe higher.
- Our model can be easily updated so that experts can identify reports more efficiently and submit reports more quickly. A potential risk of invasion of alien species could be eliminated before it diffuses.

Eventually, our work could contribute to the ecological balance and the environmental protection.

## 6 Memorandum

To the Washington State Department of Agriculture:

The news that the sudden invasion of Vespa mandarinia in Washington State has called for our attention. It's our honor to be of some help in investigating the spread situation of Vespa mandarinia and processing upcoming sightings. We sincerely hope that Washington State will get rid of the pest as soon as possible.

Vespa mandarinia is an agricultural pest, not only a threat to native species but to people. We learnt that Vespa mandarinia feeds on honeybees particularly in the fall, when Vespa mandarinia reaches its population peak every year. It is known to all that honeybees play an important role in the natural transfer of pollen. It's hard to imagine what will happen to the plants in the spring with a lack of honeybees. Consequently, Vespa mandarinia should be eradicated totally soon in Washington State.

In order to achieve the purpose, we make use of the data in 2019 and 2020 and develop our own prediction and classification models. Our prediction model can predict the spread situation of Vespa mandarinia with relatively high precision. For instance, the error of longitudinal length can be as small as 4.82 $[km]$, and that for latitudinal length can even be smaller, 1.74 $[km]$. As the data increases, the prediction will become more and more accurate.

We then further process the prediction data, and place the potential locations on a map. After that, we perform the clustering process. The results are shown in the following figure fig.14.

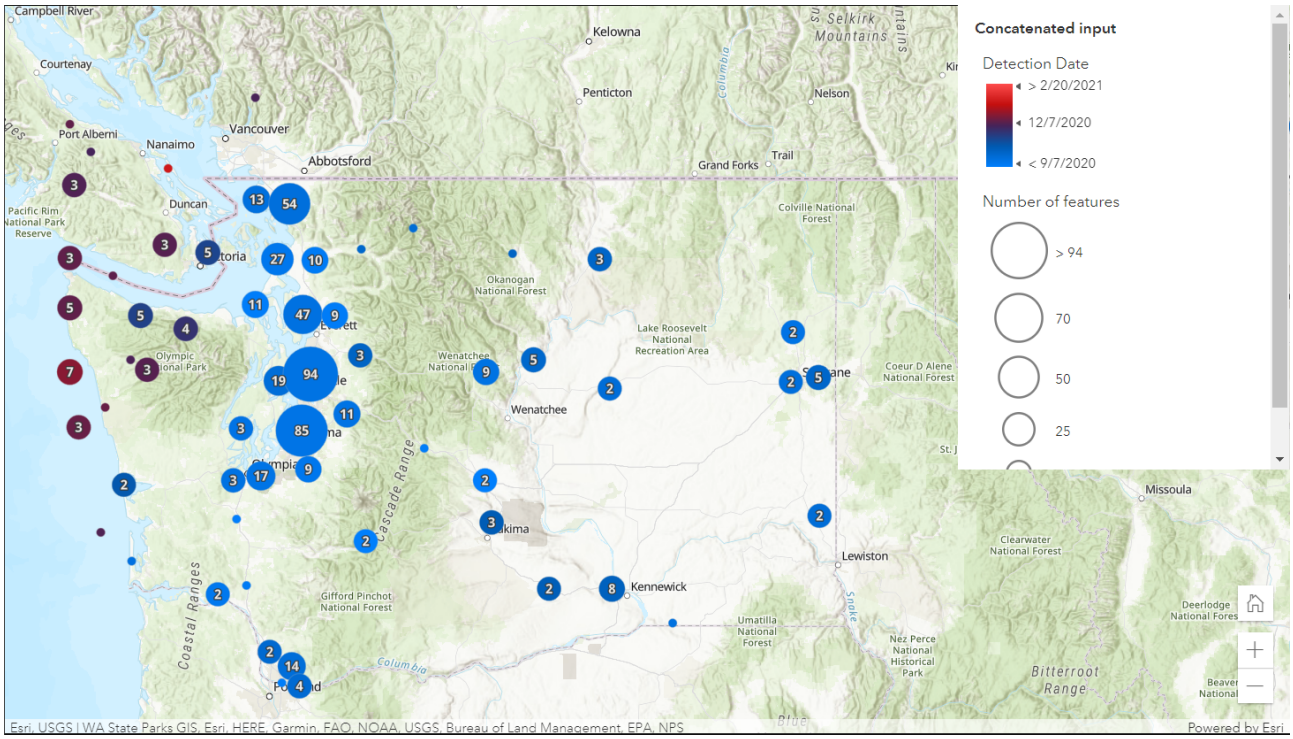


Figure 14: Clustered Prediction.

From this figure, we can clearly tell the spread tendency of Vespa mandarinia, which is from the middle part of Washington State to the western coast. This is a good news for Washington State because this is a piece of evidence that Vespa mandarinia is moving outside the district.

Our classification model can be used to do image identification. By learning the pictures of Vespa mandarinia, this model can process a photo and give a possibility about whether the photo contains Vespa mandarinia. With this model, you can prioritize the sightings with higher possibility to be Vespa mandarinia by letting the model process the images.

What's more, one of the advantages of our model is its accuracy. The accuracy of the classification can be over 80%, contributing a lot to the reliability of our model. Using our model will pre-classify most of the pictures reported and it will save a lot of manpower and material resources to classify manually.

In a nutshell, we sincerely hope that our models will be of some help in dealing with Vespa mandarinia.

Yours Respectfully, MCM Team #2109976

## References

[1] Wikimedia Foundation. Wikimedia commons, 2020. `https://commons.wikimedia.org/wiki`.

[2] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

[3] Hanna Royals Karla Salp. Asian giant hornet vespa mandarinia smith, 1852. `https://www.invasive.org/browse/subthumb.cfm?sub=85203`.

[4] Piotr Skalski. How to use numpy to build a neural network, 2018. https://zhuanlan.zhihu.com/p/47475047.

[5] Michael J. Skvarla. Asian giant hornets. *Pennstate Extension*, 2021. https://extension.psu.edu/asian-giant-hornets.

[6] Jayla Whitfield. Another invasive species..., 2020. https://exposingthebiggame.wordpress.com/tag/invasive/.

[7] Roger Freedman Young, Hugh D. University physics with modern physics. Global Edition, 14th Edition, 2015.

# Appendices

## Appendix A Code Samples

### A.1 Sample Code for LSTM Model

**Algorithm 1:** LSTM Model

**input :** Data for Latitude and Longitude with detection date, trainX and trainY
**output:** the LSTM time series model for Latitude and Longitude
initialization;
**while** *The Loss is higher than 0.03* **do**
    model initialized as Sequential;
    add one layer to the model with shape(trainX.shape[1], trainX.shape[2]);
    adjust the parameter in the model to avoid gradient increasing;
    add another layer;
    add another densely connected layer with shape (trainY.shape[1]);
    add Activation("relu");
    compile model using loss function MSE;
    fit the model ;
**end**
**return** *model*

### A.2 Sample Code for DNN Model

**Algorithm 2:** DNN Model

**input :** Input images trainX and label vector trainY, number of iterations, learning rate, test set
**output:** the DNN model
resize trainX into 250px * 250px;
flatten the matrix;
set random seeds;
initialize parameters $\omega$, b;
**for** *i from 0 to number of iterations* **do**
    calculate cost function using forward propagation;
    calculate gradient of each parameters using back propagation;
    update parameters using gradient decent;
    calculate cost function again;
    record cost
**end**
record parameters;
use parameters to predict set;
**return** *model*

## Appendix B Data for The Prediction of Vespa Mandarinia's Spread Situation

| Detection Date | Latitude | Longitude | Detection Date | Latitude | Longitude |
|---|---|---|---|---|---|
| 2020/10/7 | 49.17027352 | -124.4655896 | 2020/10/15 | 49.34792642 | -124.665014 |
| 2020/10/7 | 47.7265537 | -123.8828997 | 2020/10/15 | 48.27407842 | -124.665014 |
| 2020/10/8 | 48.92301776 | -124.5513251 | 2020/10/16 | 47.56839839 | -124.665014 |
| 2020/10/8 | 49.52368699 | -122.8407972 | 2020/10/17 | 47.69475221 | -124.665014 |
| 2020/10/9 | 47.80090322 | -124.0750809 | 2020/10/17 | 47.49047502 | -124.6443257 |
| 2020/10/9 | 47.75666786 | -123.9994218 | 2020/10/17 | 47.80283574 | -124.665014 |
| 2020/10/9 | 48.39045254 | -123.485545 | 2020/10/17 | 47.69630331 | -124.665014 |
| 2020/10/9 | 48.53449737 | -123.8621588 | 2020/10/17 | 47.94263189 | -123.7554979 |
| 2020/10/9 | 48.64531332 | -124.665014 | 2020/10/17 | 47.4908341 | -124.3248868 |
| 2020/10/10 | 48.96574578 | -124.665014 | 2020/10/17 | 48.18923916 | -124.1377971 |
| 2020/10/10 | 46.94210441 | -124.0993169 | 2020/10/17 | 47.79149832 | -124.665014 |
| 2020/10/10 | 46.63864391 | -124.3687582 | 2020/10/17 | 48.56991871 | -123.4814023 |
| 2020/10/11 | 48.12809481 | -124.665014 | 2020/10/18 | 48.73790543 | -122.6697473 |
| 2020/10/12 | 48.57361984 | -123.7789703 | 2020/10/18 | 48.51033619 | -123.477667 |
| 2020/10/12 | 47.71539582 | -122.4873235 | 2020/10/19 | 48.01412082 | -123.9929558 |
| 2020/10/12 | 47.79557506 | -121.8377628 | 2020/10/20 | 48.07388288 | -123.3061067 |
| 2020/10/12 | 48.97493541 | -124.665014 | 2020/10/20 | 47.72954649 | -123.8207859 |
| 2020/10/13 | 48.74424921 | -123.1520272 | 2020/10/20 | 47.20622806 | -124.42206 |
| 2020/10/14 | 48.1082895 | -124.1282838 | 2020/10/20 | 48.15576865 | -124.665014 |
| 2020/10/14 | 48.18329198 | -124.665014 | 2020/10/21 | 48.400179 | -124.665014 |
| 2020/10/14 | 47.34359327 | -124.665014 | 2020/10/21 | 47.60545078 | -124.665014 |
| 2020/10/14 | 48.36803272 | -124.2302224 | 2020/10/22 | 47.92866969 | -123.5968858 |
| 2020/10/14 | 48.00750908 | -124.665014 | 2020/10/28 | 48.93044592 | -122.2387128 |
| 2020/10/15 | 48.37186873 | -124.665014 | 2020/11/20 | 49.06401576 | -123.7006217 |
| 2020/10/15 | 48.57671937 | -123.5548522 | 2020/12/20 | 47.8645205 | -124.665014 |

Table 4: Corresponding Prediction Data for the Spread of Vespa Mandarinia in 2020 after September

| | Latitude | Longitude | | Latitude | Longitude |
|---|---|---|---|---|---|
| 0 | 49.17403896 | -124.4538782 | 25 | 49.35229792 | -124.665014 |
| 1 | 47.73191804 | -123.8656101 | 26 | 48.27739322 | -124.665014 |
| 2 | 48.92776934 | -124.5379701 | 27 | 47.57461443 | -124.665014 |
| 3 | 49.52857255 | -122.820109 | 28 | 47.69417562 | -124.665014 |
| 4 | 47.8089629 | -124.0617151 | 29 | 47.49565803 | -124.6359059 |
| 5 | 47.75334323 | -123.9821008 | 30 | 47.8024512 | -124.665014 |
| 6 | 48.39041011 | -123.4646326 | 31 | 47.70182486 | -124.665014 |
| 7 | 48.54064151 | -123.8544132 | 32 | 47.94729325 | -123.7408797 |
| 8 | 48.65106622 | -124.665014 | 33 | 47.48572791 | -124.3141035 |
| 9 | 48.97113468 | -124.665014 | 34 | 48.19347846 | -124.1223144 |
| 10 | 46.94173059 | -124.093804 | 35 | 47.79948029 | -124.665014 |
| 11 | 46.63927408 | -124.3544156 | 36 | 48.56901654 | -123.4709612 |
| 12 | 48.12898984 | -124.665014 | 37 | 48.73887906 | -122.6501231 |
| 13 | 48.57397937 | -123.7606421 | 38 | 48.51365724 | -123.4670535 |
| 14 | 47.7065885 | -122.4762115 | 39 | 48.01706091 | -123.9848345 |
| 15 | 47.79404092 | -121.8308073 | 40 | 48.0727181 | -123.2949466 |
| 16 | 48.96925442 | -124.665014 | 41 | 47.72756975 | -123.8165881 |
| 17 | 48.74060346 | -123.1334024 | 42 | 47.210642 | -124.4159169 |
| 18 | 48.11014386 | -124.1108326 | 43 | 48.15805177 | -124.665014 |
| 19 | 48.18721999 | -124.665014 | 44 | 48.40140542 | -124.665014 |
| 20 | 47.3429613 | -124.665014 | 45 | 47.599517 | -124.665014 |
| 21 | 48.36574067 | -124.2340001 | 46 | 47.9300033 | -123.5793421 |
| 22 | 48.01000657 | -124.665014 | 47 | 48.93275896 | -122.2284755 |
| 23 | 48.38053269 | -124.665014 | 48 | 49.06355217 | -123.6829364 |
| 24 | 48.58172284 | -123.5452854 | 49 | 47.8714681 | -124.665014 |

Table 5: Prediction Data for the Spread of Vespa Mandarinia