Forecast of Dengue Incidence Using Temperature and Rainfall

Divyadarshini Thirugnanasambandan[1], Estee Ng Jia Ying[2], Ho Zong Sien[3], Ling Zhi Yu[4], Ong Ghim Hwee Claire[5], Pang Jia Hao[6]

National University of Singapore (NUS) CS3244 Group Number 10
{e0313477, e0335669, e0200944, e0203418, e0315944, e0053599}@u.nus.edu
A0184488L[1], A0191172J[2], A0171374E[3], A0173848N[4], A0186969A[5] , A0160717H[6]

Abstract

Dengue fever is a disease caused by the dengue virus. It is spread by the Aedes mosquito and is most widespread in tropical countries such as Singapore. Our main objective is to predict the number of dengue cases for different clusters around Singapore four weeks in advance, to allow sufficient time for public health officials to take the necessary actions such as neighborhood fuming, and also to allow health care providers sufficient time to prepare. According to various studies including a study published by *BioMedical Engineering Online*, which sought to elucidate the linkage between dengue fever incidence and climate factors (Lai YH, 2018), it was found that climatic conditions play a significant role in determining where Aedes mosquitoes thrive. Places with warm temperature, humidity and frequent rainfall often have high incidence of dengue fever because these climatic conditions are very favorable for Aedes mosquito breeding. Hence, our team has designed a model to predict the number of dengue cases four weeks in advance in the four different regions of Singapore (North, South, East, West), based on temperature and rainfall.

Background

According to the National Centre for Infectious Diseases, symptoms of dengue fever start to show about five to seven days after a person has been bitten by a viral mosquito. In most cases, people may experience mild symptoms such as mild fever, rashes or headaches. However, in more severe cases, known as Dengue Hemorrhagic Fever (DHF), victims can develop symptoms like high fever and bleeding of gums. Even though DHF rates are low (less than 1%), dengue cases can become fatal. There are four types of dengue virus (DENV1-4). When a person is infected with one type of dengue virus, he becomes immune to that type for the rest of his life. But this does not mean that he cannot be infected by the remaining three types of viruses.

Being very near the equator, Singapore experiences warm and humid weather all year round, with temperatures ranging from 23°C to 33°C and high amounts of rainfall. According to research (Reinhold et al., 2018), the time taken for Aedes mosquito larvae to hatch and develop into adult mosquitos is shorter at higher temperatures of around 30°C compared to lower temperatures like 21°C. Another research (Morin, et al., 2013) also mentioned that the spreading ability of viruses is positively correlated to climate temperature. The duration for the DENV-1 and DENV-4 viruses to be detected in the saliva of an Aedes mosquito from the time of its feed greatly decreased from 9 days (at temperatures of around 26-28°C) to 5 days (at 30°C). This means that an infected mosquito will be able to spread the virus to more victims during its lifespan.

This is an alarming concern to Singapore because, according to a news article ("Weekly dengue cases remain 'high'; total deaths this year rise to 7: NEA", 2020), over 6,000 people have become infected with the dengue virus and 7 have died from the virus in 2020 alone (as of April 27). This tells us that the dengue virus is spreading much quicker due to the warmer temperatures and humid weather, partly brought about by global warming. In addition, while most victims recover from the virus, severe cases may be fatal, which is a pressing reason to address the situation of the increasing number of dengue cases in Singapore.

As such, our model, to predict dengue incidences four weeks in advance, can give public health officials sufficient time to take action against the spread of the disease and formulate necessary policies to deal with the spread. It could serve as a friendly reminder to residents to check for and clear their homes of stagnant water so as to prevent mosquito breeding. The prediction will also allow health care providers to be better equipped and prepared for treatment of dengue fever.

RNN & LSTM

In recent years, the Recurrent Neural Network (RNN) model has become increasingly popular amongst organizations to predict data with some form of sequence to it. This works with the data that we have collected because both the weather and the cluster data were recorded weekly. Hence, We decided to consider RNN as our main prediction model.

The RNN are networks with loops in them (Figure 1), which allow information to persist. This means that unlike other neural networks that predict information only based on the current input, the RNN takes into consideration previous inputs in its decision-making.

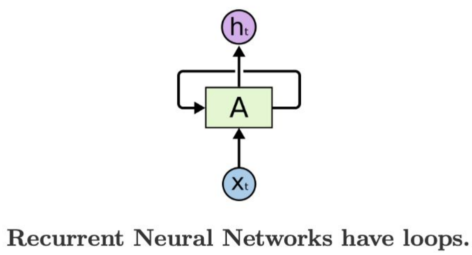

Figure 1: Overview of a RNN

This is useful for our proposed application because the dengue virus cannot be predicted solely based on the current input. Different factors trigger the development of the virus at different locations. The ability of the RNN model to consider former weather data (e.g. temperature and rainfall from previous years in a particular location) allows it to better predict the possible number of dengue cases during the same period in the current year in the same location.

One of the appeals of RNNs, is the idea that they may be able to learn the order of dependence over time. However, when the gap between the relevant information and the point where it is needed becomes very large, regular RNNs become unable to learn to connect the information. This issue of long-term dependencies is a problem in our situation, because the current dengue cases have long term dependencies on previous cases. As such, we decided to use Long Short-Term Memory (LSTM) networks.

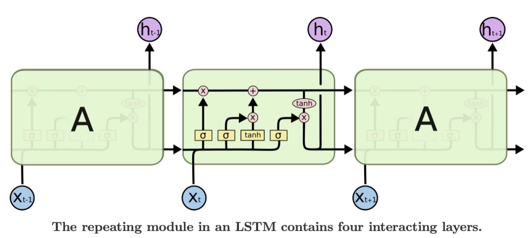

Figure 2: Inside a LSTM

LSTM networks are a special kind of RNNs, capable of learning long-term dependencies. Just like a normal RNN, the LSTM has a chain of repeating modules of neural networks (Figure 2). However, instead of a single neural network layer, the repeating module has four neural network layers interacting with each other. The LSTM also has the ability to remove or add information to the cell state, which is carefully regulated by structures called *gates*.

Gates are a way to optionally let information through. They are composed of a sigmoid neural net layer and a pointwise multiplication operation. The sigmoid layer outputs numbers between zero and one, describing how much of each component should be let through. An LSTM has three of these gates, to protect and control the cell state.

Experimental Setup

We will first describe our data collection process, followed by feature extraction and pre-processing of the collected data. We will then discuss our chosen model and the hyperparameters that yielded the best results for all four models. Lastly, we will discuss the limitations and improvements to our approach.

Dataset

We chose to use weather data from the Meteorological Service Singapore's (MSS) website which was obtained by web-scraping. This website provides accurate weather information for different parts of Singapore which suited our needs.

In addition to our weather data, dengue cases data were collected from the National Environmental Agency's (NEA) website on Dengue Clusters. However, due to limitations of the website, the case count was updated every week with no accessible historical data. Fortunately, screen captures of the same weekly dengue clusters from 2013 were found hosted on outbreaks.sgcharts.com. Manual data entry was then done to convert the screen captures into a suitable tabular format for training. As such only dengue data from 2016 - 2018 were chosen. The issue of limited observation points will be addressed in the upcoming sections.

Feature Extraction

To ensure the accuracy of our model we had to choose the right attributes. We decided to use temperature and rainfall along with the dengue case counts because, as discussed above, these features were the most impactful in Aedes mosquito breeding.

Data Preprocessing

Missing weather information was a major issue during the collection of data. Thus, the daily weather data was averaged over each week and across each region (North, East, South, West). We decided that using weekly data would still allow observation of some sort of trend while at the same time, maintain accuracy of the weather data. Similarly, the dengue case counts were grouped and summed over each region per week. The combined dataset was split into two-thirds training (2016 - 2017) and one-third validation (2018). This was done to capture any seasonality over each year that could have occurred.

The resulting feature values were standardised by subtracting its mean and dividing by its standard deviation. This was prefered over normalisation as the units in terms of dengue case count, rainfall and temperature were different. Next, the training and validation data were organised as a supervised learning problem where 16 weeks of past observation were used to predict observations four weeks into the future. For example, the training example would be an array of 16×3 (16 weeks consisting of three features) and the label would be the dengue case count at week 20. This is repeated for each week in our dataset (i.e. weeks 1-16 to predict week 20, weeks 2-17 to predict week 21, and so on). Thereafter, the resulting data were batched in sizes of 26, cached and shuffled.

Methodology

Due to the stochastic nature of LSTM, multiple runs were done during the search for the optimal model using the training and validation diagnostic plots. Therefore an average validation loss across 10 runs was computed and compared to the baseline model. Due to the limited data (154 observation points), overfitting was a salient issue. Therefore, we carefully tested hyperparameters that were aimed at minimising overfitting. Specifically, the use of different loss functions, increasing the model's capacity and the use of different regularisation techniques. Additionally, we chose to impose a constraint of reusing the same model with the same hyperparameters for all four regions in Singapore as we wanted to train a model that is capable of generalizing across each region in Singapore.

Baseline

The input example into our neural network is a 16×3 array produced by the preprocessing procedure above. We chose an LSTM model with a single fully connected hidden layer with 16 units and a single output unit. We chose 80 epochs to train the data, using RMSprop as the optimiser with initial learning rate of 0.001 and the loss function Mean Absolute Error (MAE) was used. We chose 80 epochs because it struck a good balance between training and validation loss across the four model predictions. RMSprop was chosen as an adaptive gradient descent algorithm alternative to the classical Stochastic Gradient Descent (SGD), since it provided a heuristic approach without requiring computationally expensive work in tuning hyperparameters for the learning rate schedule manually.

Loss Function

Here we compare the use of two other popular regression loss functions, Mean Square Error (MSE/L2 Loss) and Huber Loss. MSE was chosen as the classical loss function. The squaring means that larger mistakes result in more errors as compared to smaller mistakes, thereby punishing larger errors. On the other hand, Huber Loss was chosen as it combined the benefits of MAE and MSE. Firstly, Huber addresses the issue of large gradients of MAE where convergence to the local minima may be an issue. Huber attempts to rectify this by curving around the minima which decreases the gradient. Secondly, Huber Loss is less sensitive to outliers and thus, more robust as compared to MSE considering that our limited dataset might be prone to overfitting.

Model Capacity

Next, an alternative multi-layer (two hidden layer) model variant was tested for any possible improvement in validation loss. This variant had 32 output units in the first hidden (LSTM) layer and 16 output units in the second hidden layer.

Regularisation

Lastly, an L2 weight regularisation of value 0.0001 was used to produce a simpler model which aligns with Occam's Razor principle by minimising the weights learnt. To further minimise the problem of overfitting, a drop-out layer with a drop-out rate of 0.2 was used to randomly "drop-out" (i.e. set to zero) a number of output features of the hidden layer during training. These two regularisation techniques were tested individually and in combination.

Other notable behaviours that resulted in poorer performance were the addition of recurrent regularisers and recurrent drop-out layers. Both of which increased the loss. Additionally, an alternative activation function Rectified Linear Unit (ReLU) was tested with no significant improvement in validation loss.

Discussion of Results

Below, we present the results from our experiments in Table 1. First, we present the results from the use of different loss functions. To our surprise, the Huber Loss outperformed MSE, by a very small margin, and (not surprisingly) MAE. There was an average improvement of 79.9 % across each region with the East region's validation loss decreasing as much as 96.9%. On the other hand, MSE had an average improvement of 78.2% in validation loss.

Next, we present the results of adding an additional hidden layer. There was an average improvement of validation loss of 25.4%.

Finally, we present the results of using regularisation techniques, such as adding kernel regularisation and a drop-out layer. The addition of a L2 weight regulariser with value 0.0001 yielded an average of 22.8% improvement in validation loss. While the addition of a drop-out layer with a drop-out rate of 0.2 yielded average of 25.3% improvement.

	North	East	South	West
Baseline (MAE)	0.242624	0.242624	0.572018	0.312457
MSE	0.035896	0.011561	0.262287	0.073300
Huber	0.030130	0.019320	0.198469	0.088192
Multi-Layer	0.106159	0.193160	0.571485	0.322334
(L2) Kernel Regularisation	0.223551	0.089556	0.529578	0.312919
Dropout	0.189193	0.128635	0.586894	0.261811
(L2) Kernel Regularisation + Dropout	0.211322	0.211322	0.632688	0.273112
Final	0.006164	0.005917	0.184670	0.065077

Table 1: Validation Loss Comparison

Final Model

In our final model, we tweaked the hyperparameters based on the best performing additions to the model to obtain an average of 0.065 validation loss across all regions. Specifically, a two fully-connected hidden layer model, with RMSprop as the optimiser and Huber Loss, was used. A dropout rate of 0.2 and L2 kernel regulariser of value 0.0001. Figures 3.1 and 3.2 shows the prediction of our model against the ground truth for the North region in Singapore and the associated change in loss respectively.

Our prediction will help government officials make the optimal decision on how to move forward with regard to the rise of dengue cases. If more cases are predicted during a certain period, government officials can be better prepared to make timely necessary decisions so as to limit the spread of dengue fever. In addition, our predictions would also provide important information to healthcare providers, should they need to increase or reduce their purchase of equipment or medication according to the possible number of cases in the coming months. This will not only help them cope with any surge in dengue cases so that they will be better able to provide effective and efficient treatment to affected individuals, but it will also help to save costs on medical supplies when there are fewer incidences of dengue in certain times of the year.

As seen above, our predictions will be able to improve the efficiency of the government and healthcare providers towards alleviating the problem of dengue in Singapore. It would not only improve the quality of life for residents but will benefit the economy with a healthy and productive workforce.

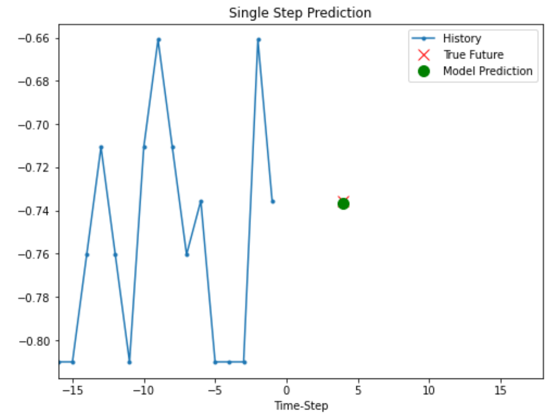

Figure 3.1: Model Prediction (North Region)

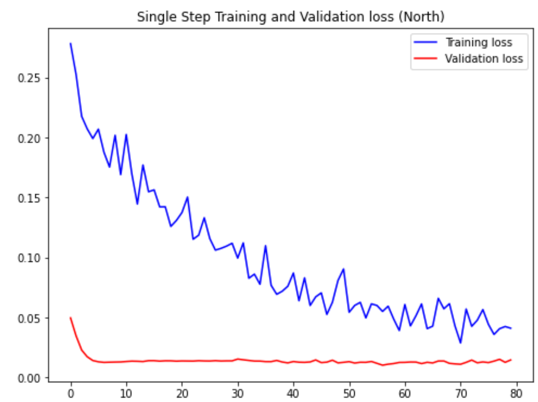

Figure 3.2: Model Loss (North Region)

Limitations

We had to individually key in the number of dengue cases for each location based on the data provided in an image in a PDF format found online. Since the data collection process was very labour intensive, we had very few data points to work with. This was a limitation that affected the performance of our model.

The manual entry process could have also led to some inaccuracies such as keying the wrong number due to a misread. Furthermore, we had to manually determine which locality a specific location belonged to using our visual interpretation. This was so that we can classify the different locations according to the localities present in our weather data. For example, we had to manually determine if the location “Woodlands Ring Road” belonged to “Admiralty” or “Admiralty West” using visual comparison of the respective locations because our weather dataset did not contain “Woodlands” as one of its localities. The process of splitting our dataset into four regions also meant that inaccuracies would arise during the splitting process. These are some limitations that almost certainly resulted in inaccuracies because of human error.

Due to the lack of domain knowledge in Aedes mosquitos and the spread of dengue fever in general, we had to do extensive research to decide which weather features greatly impacted the number of dengue incidences. However, we do feel that we might have left out other factors (besides weather and temperature) that could significantly affect mosquito breeding and rise of dengue cases, as a result of our unfamiliarity with the topic. Examples of these factors would include the level of urbanization, or the type of housing in the region.

Model Evaluation

An alternative solution to the vanishing gradient problem would be the Gated Recurrent Unit (GRU), a close variant of LSTMs. GRUs utilise two gates to solve the problem: the reset gate decides what past information to forget and the update gate chooses the information to be passed along to the future. While GRUs and LSTMs share similarities, several differences exist within the architecture of the units. The structural differences make for varying effectiveness, which are heavily dependent on the datasets and tasks (Chung, 2014).

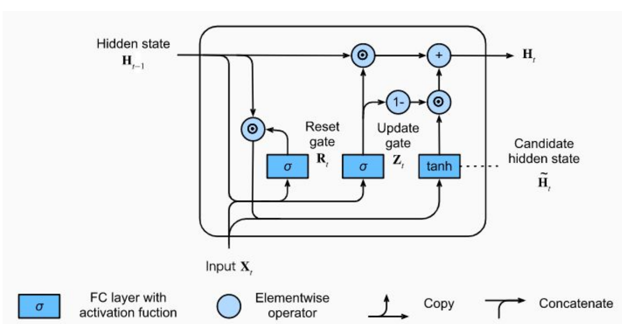

Figure 4: GRU Unit

The memory content of LSTMs which are seen or used by other units in the network is controlled by the output gate while GRUs expose its full content. Another difference is that LSTMs control the amount of new information flow from its previous hidden state independently of the *forget gate* while the GRUs's new information from the previous hidden state is regulated by the *update gate* (Chung, 2014).

In terms of qualitative advantages, GRU uses less training parameters (which also means less memory used), executes and trains faster than LSTM but LSTM is more accurate on a dataset using longer sequences.

To make a comparison between using LSTMs and GRUs in our model, their respective units are calibrated such that the number of parameters are approximately equal according to Table 2. The remaining model hyperparameters are unchanged as part of the controlled experiment. The 'Final LSTM' model in Table 1 is used as the control, and 'Final (GRU)' is the experiment.

	Units	Params
LSTM	32	4608
GRU	37	4662

Table 2: Number of LSTM and GRU units and the corresponding model parameters.

By comparing the plots of validation loss against the epochs of the models of a single arbitrary run (Figure 5), it is observed that both models perform similarly well when trained with limited datasets. The learning curves for the GRU appear to be better than those derived from the LSTM, which appear to exhibit overfitting as validation loss is higher than its corresponding training loss. However, these observations may not consistently hold on multiple runs.

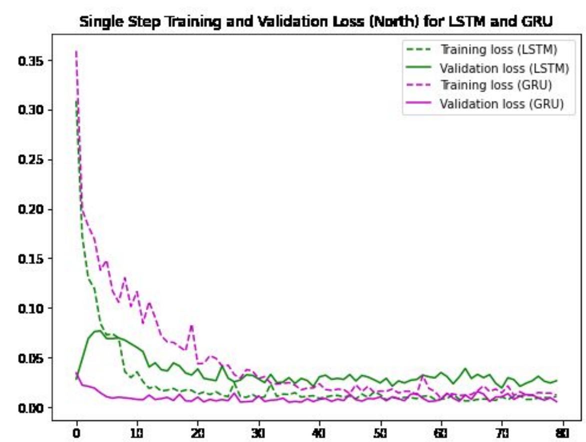

Figure 5: LSTM vs GRU Model Loss Comparison

	North	East	South	West
Final (LSTM)	0.026595	0.040172	0.166777	0.093481
Final (GRU)	0.014497	0.026316	0.242855	0.207666
Percentage Improvement (%)	45.49	34.49	-45.62	-122.18

Table 3: Validation Loss Comparison between LSTM and GRU models

The validation losses for both models were computed for the divided regions and averaged over 10 separate runs, as per Table 1's derivation methods. From Table 3, the GRU performed better than LSTM for the North and East regions, but performed worse for the South and West regions, with its performance for the West region disimproving by -122.18% as seen in Table 3. Therefore, the LSTM appears to be better suited for this particular dataset and task. However, whether its performance edge in terms of validation loss is generalizable over a large dataset is still inconclusive.

Improvements

Below are two other methods we could use to improve our model.

Firstly, we could use a nested cross validation method known as Forward Chaining (Tashman, 2000) for our time series forecasting.

> In a real-world forecasting environment, we stand in the present and forecast the future (Tashman, 2000)

As such, this method of validation is preferred to the traditional K-Fold cross validation, to account for temporal dependencies within our dataset. This method splits each week of observation as a test set and utilises all data before it as the training set. The averaging of error over each training test split will thus produce a more robust estimate of the model error.

The second method is Early Stopping. Early Stopping is a form of regularisation used to avoid overfitting when training a learner with an iterative method, such as gradient descent. Too little training will mean that the model will underfit the training and test sets. Too much training will mean that the model will overfit the training dataset and have poor performance on the test set. A compromise is to train on the training dataset but to stop training at the point when performance of a validation dataset starts to degrade. Basically, during training, the model is evaluated on a holdout validation dataset after each epoch. If the performance of the model on the validation dataset starts to degrade (e.g. loss begins to increase or accuracy begins to decrease), then the training process is stopped. (Brownlee, 2019)

Conclusion

In our project, we used the LSTM model to help us predict the number of dengue cases four weeks in advance for the four different regions (North, South, East and West) in Singapore, given 16 weeks of historical observations.

Firstly, our multivariate time series forecasting based on the aforementioned factors - dengue case counts, mean temperature, mean rainfall - showed promising results. However, there may be other hidden attributes not captured by our model that may help produce a more robust prediction. Indeed, more domain knowledge with regards to Aedes mosquitos can greatly improve the accuracy of the model.

The main goal of our project was to train a model that could predict the number of dengue cases within each region in Singapore. Since we had imposed a constraint of reusing our model for each region in Singapore, some region's data yielded better results than others. Thus, a trade-off had to be made when deciding on the optimal hyperparameters.

Finally, with more data of dengue case counts that could be used as training and validation sets, the evaluation of the model would be much more accurate, and the prediction might be more certain.

The prediction would be very useful for the respective healthcare and government officials to take necessary timely actions. Our model would allow government officials to design better policies proactively, that are data-driven, thus allowing healthcare providers to better allocate resources. Given the sudden spike in dengue case count at the time of writing this report, our model has become even more relevant. Therefore, we believe that our project has the potential to improve the quality of life for the residents of Singapore.

References

Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola, (2020, May 01) *Dive into Deep Learning. 9.1. Gated Recurrent Units (GRU)* https://d2l.ai/chapter_recurrent-modern/gru.html

Brownlee, J. (2019, August 06). *A Gentle Introduction to Early Stopping to Avoid Overtraining Neural Networks*. https://machinelearningmastery.com/early-stopping-to-avoid-overtraining-neural-network-models/

Channel News Asia (2020, April 27), *Weekly dengue cases remain 'high'; total deaths this year rise to 7: NEA*. https://www.channelnewsasia.com/news/singapore/two-more-dengue-deaths-singapore-weekly-cases-remain-high-nea-12681434

Chung, Junyoung & Gulcehre, Caglar & Cho, KyungHyun & Bengio, Y. (2014). *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*.

Colah's blog. (2015, August 27). *Understanding LSTM Networks* [Blog post]. https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Lai YH. (2018, November 6); *The climatic factors affecting dengue fever outbreaks in southern Taiwan: an application of symbolic data analysis*. *Biomed Eng Online*. 17(Suppl 2):148.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, et al. (2015) *TensorFlow: Large-scale machine learning on heterogeneous systems.* Software available from tensorflow.org.

Morin, C. W., Comrie, A. C., Ernst, K. (2013). *Climate and Dengue Transmission: Evidence and Implications*

Reinhold, J. M., Lazzari, C.R., Lahondere, C. (2018). *Effects of the Environmental Temperature on Aedes aegypti and Aedes albopictus Mosquitoes: A Review.*

L. J. Tashman. (2000) Out-of-sample tests of forecasting accuracy: an analysis and review. International Journal of Forecasting, 16(4):437–450

Roles

Data Collection and Processing : Claire, Divyadarshini, Estee, Zong Sien

ML Model: Jia Hao, Zhi Yu, Zong Sien

Report : Claire, Divyadarshini, Estee, Zhi Yu, Zong Sien,