Name: Dulanjana Perera (2059757)
Course: CSC 578 / Final Project

1. Data Exploration

The dataset consists of several categorical and numerical variables; temperature, the volume of rainwater, snow volume, and weather conditions are a few of them. The goal is to predict the traffic volume (numerical) after 3 hours using the above variables. The data samples were recorded at 1-hour intervals from October 2012 to October 2018. Prior to the preprocessing, the dataset is examed to identify potential preprocessing methods. Since both types of variables are presented, different data visualization methods are followed.

A full statistical description is generated to understand the distribution of numerical values and categorical variables. The description indicates a few anomalies in numerical data and a significant amount of categorical classes in some categorical variables. Therefore, a box plot is plotted to identify the outliers in numerical variables.

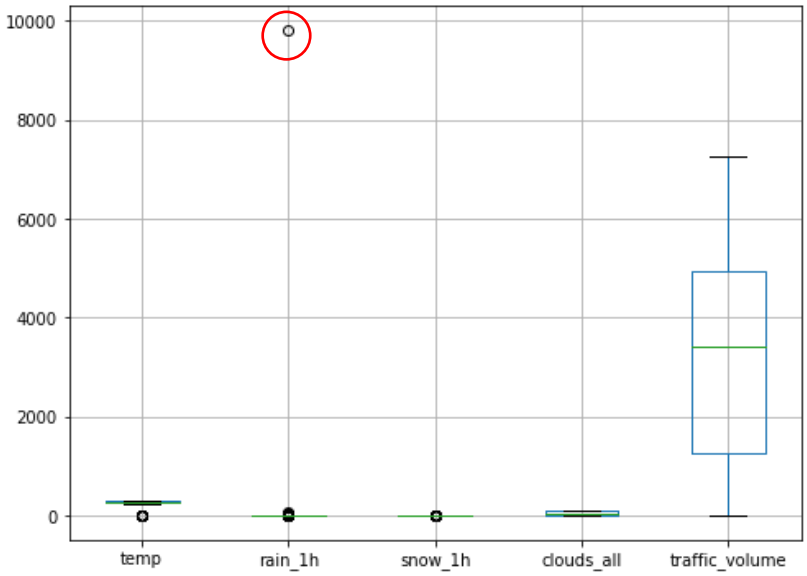

Figure 1: Box plot for numercal variables

The outlier is replaced by the mean of rainwater volume in the preprocessing section. As Fig. 1 illustrates, other variables are distributed normally, but the range of values of variables is substantially different from each other. Therefore, normalizing the data is the

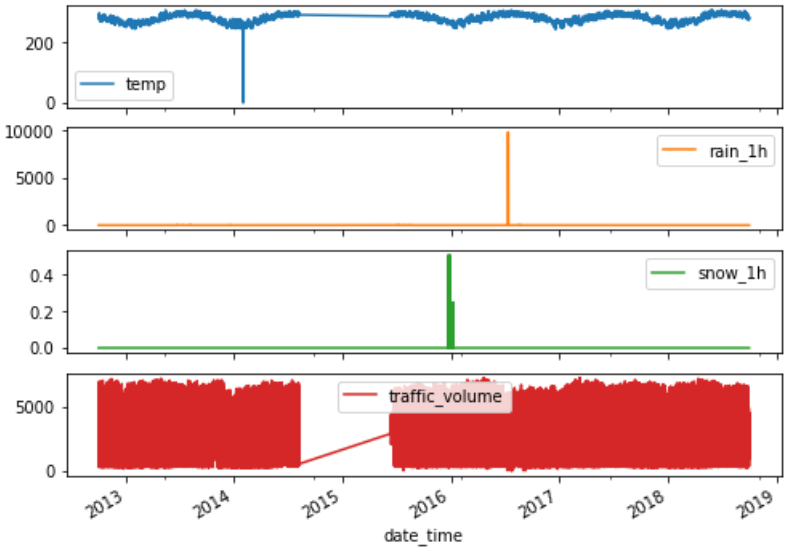

Figure 2: Raw Data visualization

first step before data is processed for training. However, as suspected in the statistical description, the temperature seems to have outliers (Fig. 2)

Since the temperature is in the Kelvin scale, it is impossible to find 0 K in nature. Hence, I rule out that this is an outlier in this variable. Similar to the rain variable, this is also replaced by the mean of the temperature. Notice there is a significant gap around 2015, which does not contain any values of all the numerical and some categorical variables. However, removing this data will result in poor prediction.

After cleaning the numerical data, a histogram is plotted to observe the distribution. All the variables are skewed heavily and normalization will reduce the skewness.

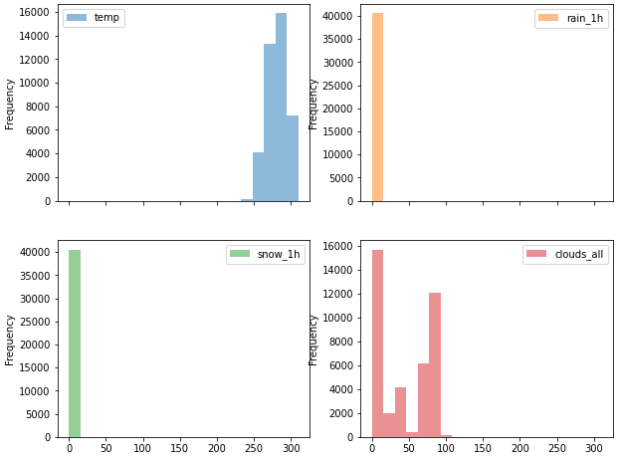

Figure 3: Histogram of numerical variables

The categorical variable examination is conducted. The frequency of each category class is computed.

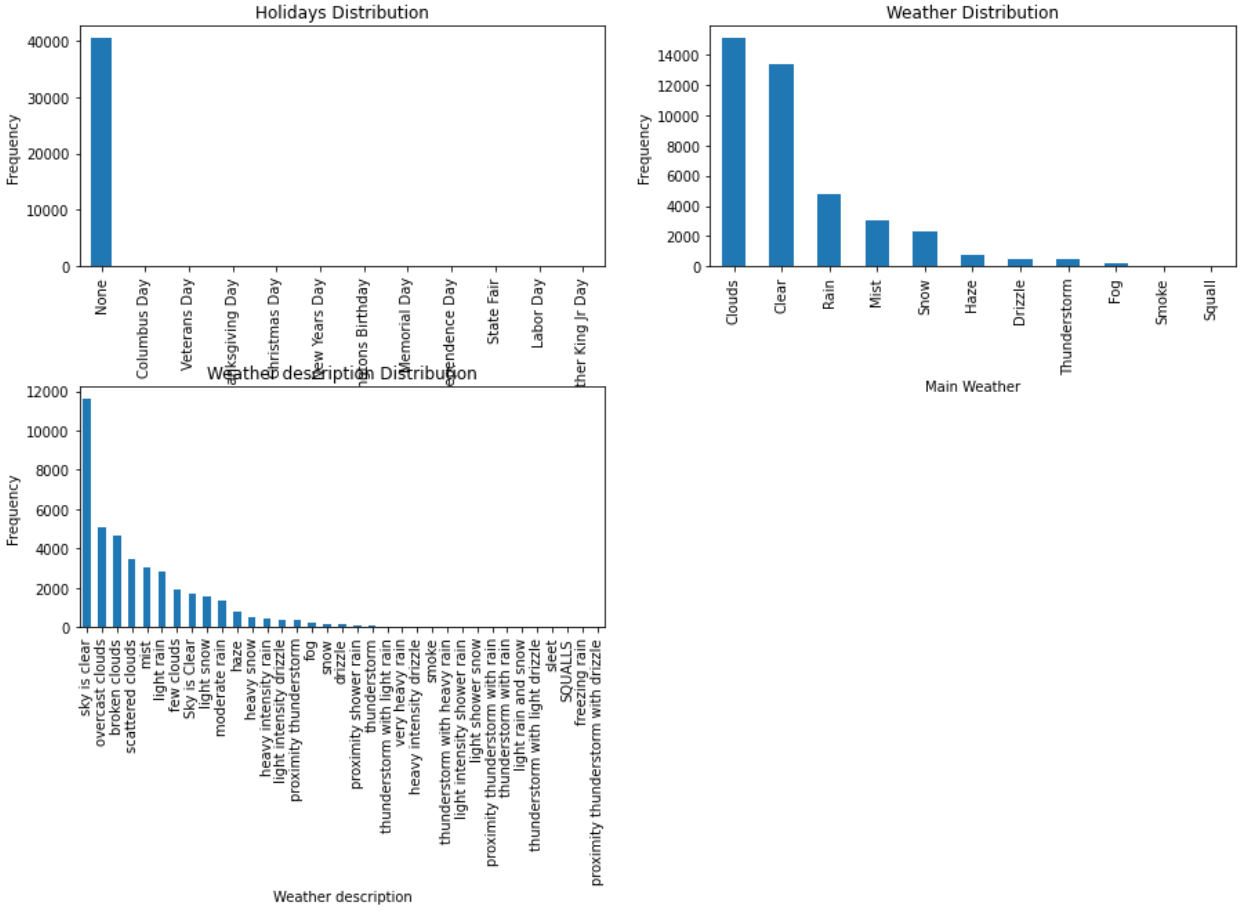

Figure 4: Frequency of each class in each variable

Figure 4 indicates that the holiday category has no meaning because a substantial portion of the class belongs to None. Other categorical variables also show highly imbalanced class distribution. Moreover, during the random check of the data, I noticed that weather conditions are inconsistent with the rain and snow volume. Hence, I decided only to use numerical values.

2. Data Preprocessing

The ‘rain_1h’, ‘temp’, ‘snow_1h’, ‘clouds_all’ and the target ‘traffic_volume’. In addition to these, the datatime variable is converted into a sinusoidal wave to keep the continuous time and provide periodic behavior to the dataset. Moreover, the FFT analysis shows that day as frequency has major effect on the ‘traffic_volume’ variable. The other numerical

variables are normalized to zero mean. Before the normalization, the total dataset is grouped into *training, validation*, and *testing*. Then the training set means and standard deviations of each variable are used to normalize the other sets. This voids testing data incorporated with the training phase and in practice, we do not have the test's standard deviation and mean except only past data.

3. LSTM Parameter Tuning

The various recurrent neural networks are developed using Keras libraries. Before developing preliminary models, a baseline model is developed to compare the novel networks. This model only contains input and output layers and does not make any predictions except output the previous hour's traffic volume. If I provide 12 hours data set, then the baseline outputs the initial time step value. The baseline does not perform well with future values.

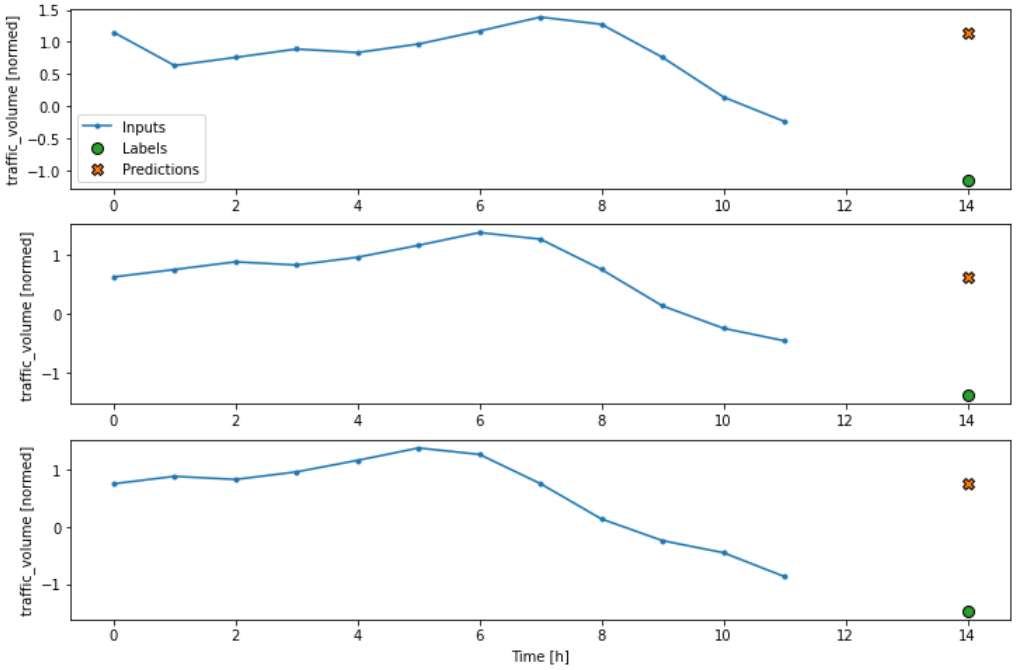

Figure 5: Baseline prediction of 3 hour window future

After defining the LSTM, as usual, the best epochs are searched. The test results suggested that, after 20 epochs, models tend to learn less. This test is conducted for a single lstm layer with 32 nodes.

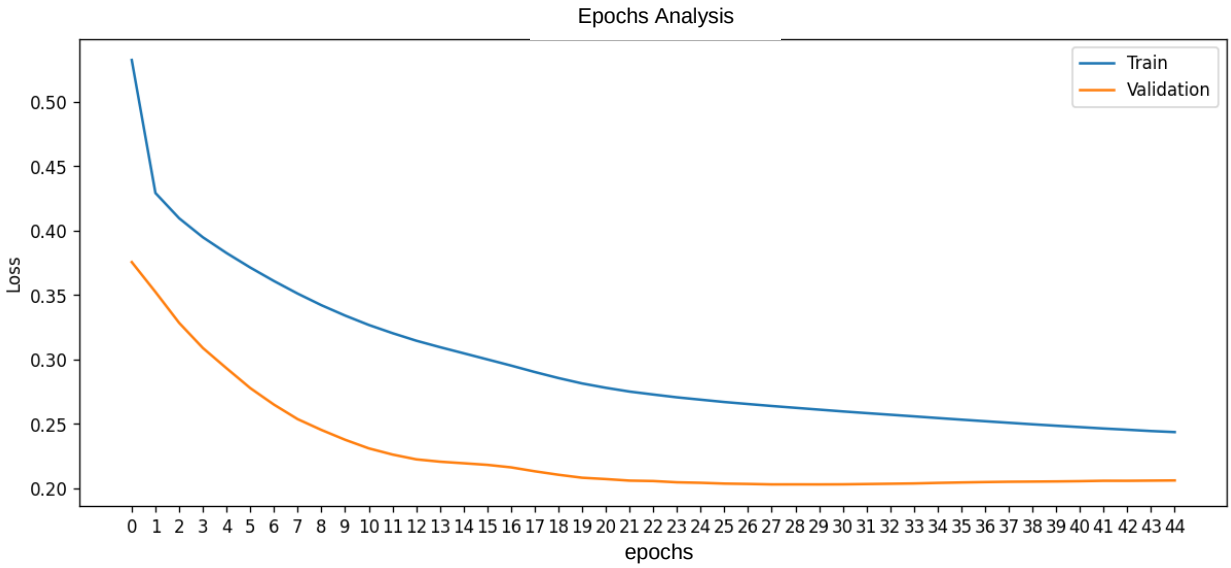

Figure 7: Epochs vs Loss

The second step of the tuning process is the batch size search. In this case, I searched the batch size and number of nodes in the lstm layer at the same time. Here, only a 1 lstm layer is used. I hope to see a trend in the direction of number of nodes. In this analysis, only selected nodes are evaluated with selected batch sizes.

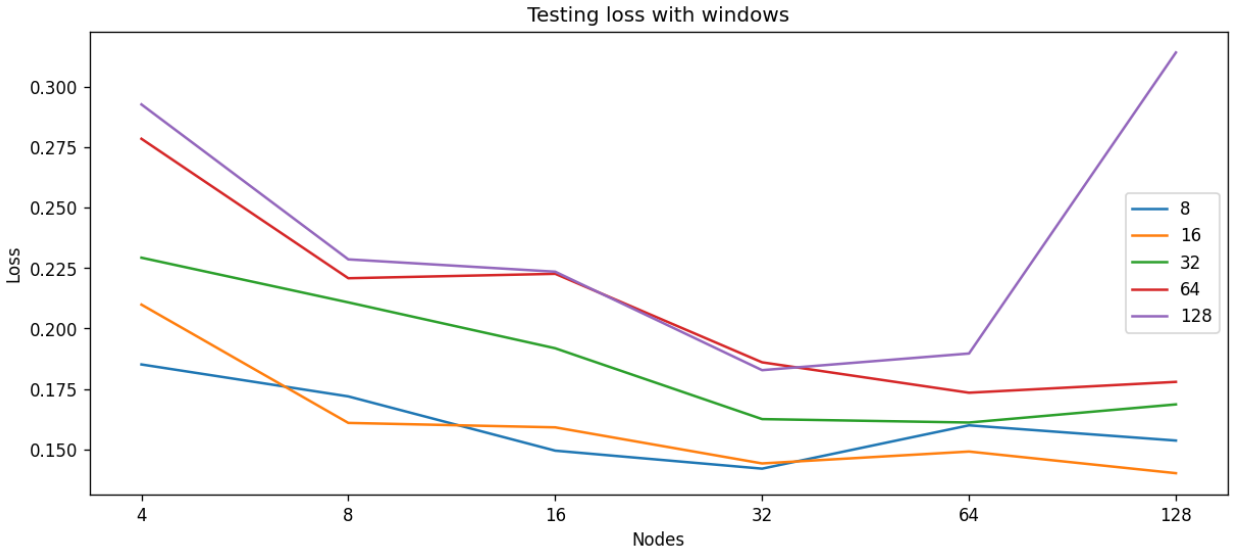

Figure 6: Nodes vs Batch size. Each line indicates the batch size

This shows that the smaller the batch size, the lower the loss. Furthermore, as expected, a downward trend is observed. Higher nodes generally produce a lower loss. The main reason because higher nodes tends to generalize the model and smaller batch size allows the model to train more iterations with different batches.

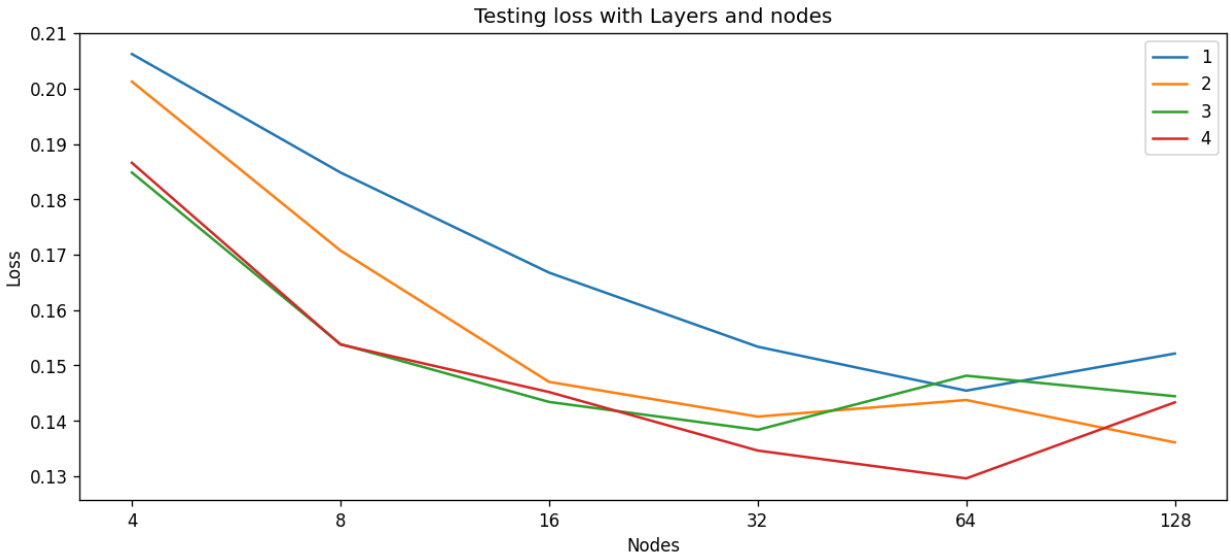

Figure 8: Layers vs. Nodes. Each line indicates number of LSTM layers.

The next experiment is to investigate the layer size. In this case, I used 16 batch size throughout the experiment as suggested by the previous results. I don not have any idea what model will result with different number of layers. Based on past experience, I can guess that the higher number of layers and the lower number of layers always produce inaccurate results due to the overfit and underfit. Hence, the sweet spot should be between 2 – 4 layers.

Although the initial guess is not entirely true, the 4-layer network shows lower loss at all nodes. This implies that the 4-layer network is generalized well for unseen data. Further tuning is conducted for 4-layer, 16 batch size 64-node lstm layer to investigate how recursive drop will perform.

The results in Fig. 9 indicate that the model surpasses the other variations and the baseline model. The testing Mean Absolute Error (MAE) is 0.1334 (normalized). All the other models produce MAE> 0.14**. The model obtained 267.83 in the submission portal, whereas the model without dropout obtained 270.60. Since it is not a significant improvement, I decided to go back and evaluate other models with 0.2-dropout. However, layers less than 4 always produce scores higher than 280.

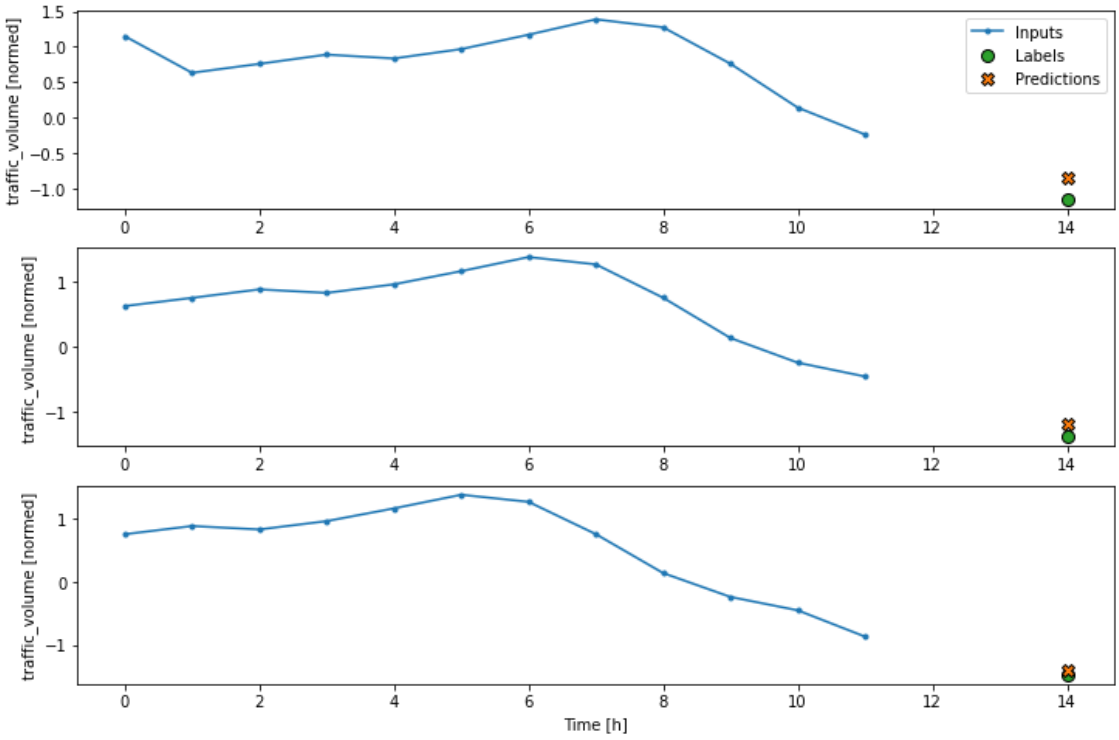

Figure 9: LSTM with dropout (0.2)

The other evaluations, such as 5-layer (with parameters of 64/32 nodes, batch 16/32) with/without dropout, did not surpass the best model. Only the with-dropout got the closest score, 272.05.

Statefulness

The statefulness definitely improves the performance of the LSTM. However, the improvement is not significant. Since [4-layer, 64-nodeEach, 0.2-drop] model performed better than other, I decided to try the statefulness of the model. It actually reduces the MAE of the test set to 0.1242. However, due to batch size restriction, this model was trained on 18-batchsize. This improvement aroused my curiosity and I tried several other variations of the model. Since I saw that the smaller batch size generate good results (refer Fig. 6), I used 6 and 9 sizes in the experiments. The following Table 1 shows the experiment results.

Table 1: Final experiment with statefulness (the values in the code file are different from this due to random initializations)

Batch Size	Nodes	Layers	MAE
18	64	4	0.12425
18	32	4	0.13826
18	64,32,16,8	4	0.13742
9	64	4	0.12185
9	64	2	0.12415
9	32	5	0.13224
6	64	4	0.13856
6	32	4	0.14526

The Best model is highlighted in red in the table. It produced 245.16 in the public leader board (Late submission). However, notice that model with green color text. It produced 248.75 in the public leader board (Late submission). Even though the red model performs better than the green model, it is more complex than the green model. Therefore, I would prefer the green model due to its simplicity.

4. Conclusion

LSTM is widely used to model periodic patterns. In this experiment, 6 numerical variables are used to predict the traffic volume at 3 hours in the future. The developed model can predict the 3-hours in future traffic volume with MAE of 260 (the traffic volume range 300-6000). In this experiment, only a few parameters were studied. The tuned parameters are the number of layers, nodes in each layer (same number of nodes at each layer), dropout, batch size, and statefulness. However, the model can be further improved by using bidirectional layers, a more rigorous dropout search, and a different number of nodes in each layer. Moreover, the usage of categorical variables may affect the performance if they are appropriately transformed into a useful numerical representation.