Predicting Traffic Flow on Faulty Traffic Detectors Using Machine Learning Techniques

Alben Rome B. Bagabaldo[1] and Marta C. González, Ph.D.[2]

[1]Dept. of Civil and Environmental Engineering, Univ. of California, Berkeley, CA. Email: bagabaldo@berkeley.edu
[2]Dept. of Civil and Environmental Engineering and Dept. of City and Regional Planning, Univ. of California, Berkeley, CA. Email: martag@berkeley.edu

ABSTRACT

Caltrans' performance measurement system (PeMS) collects traffic data from thousands of detectors. These detectors could malfunction resulting in less than 100% to 0% observation at certain vehicle detection stations (VDS). Thus, this study aims to predict traffic flow when no data is available on faulty detectors using machine learning techniques such as linear regression and random forest. The entire length of I-680 South was considered using flow data over two weeks. The results of the prediction of flow show that using the eigenvectors from principal component analysis in a linear regression model gives very low accuracy. Meanwhile, simply using the select features from the data results in better performance which is then improved when the output from k-means clustering was added as a feature. Meanwhile, using random forest gives the best performance compared to the previous models but with a greater model training time.

INTRODUCTION

Among the primary elements of traffic stream are flow, density, and speed. These elements and their relationship to each other are being studied, mathematically, in traffic flow theory (Garber and Hoel 2020). Traffic flow, together with the other elements of traffic. is beneficial in understanding mobility and flow data are often used in planning and designing highways. When analyzed, it can help guide traffic management measures being implemented in a city. These can also help measure delay and congestion in the network to have an overall understanding of traffic. In some cases, flow is also being used to estimate energy consumption and pollution level.

Flow is usually measured in terms of the no. of vehicles per unit time (*e.g.*, vehicles per hour) and the data is usually gathered through empirical studies. Ideally, anything which allows us to determine the number of vehicles passing through a given location or section of a road at a given unit of time can be considered as a method to account for traffic flow. There have been many advancements in these methods. Today's methods include (but not limited to) the use of loop detectors, radars, cameras, or closed-circuit televisions (CCTVs), and unmanned aerial vehicles (drones).

Despite these advancements, any of these systems can be subject to failure. For instance, detectors (e.g., loops) occasionally malfunction, cease working or halt sending data just like in the case of one of the well-established systems of gathering data on the freeways of California called the California Department of Transportation (Caltrans) Performance Measurement System (PeMS). PeMS User Guide (Caltrans 2020) mentions that these errors can happen for many reasons including communications loss and faulty connections which then creates gaps in the

data set. If these gaps are not filled, PeMS wouldn't be able to serve its main purpose which is to give the condition of freeways at high accuracy. Given this, it is the aim of this study to predict traffic flow when no data are recorded by the sensors located at any specific vehicle detection station (VDS) at a given time.

As a case study, we randomly selected Interstate 680 Southbound direction (I-680 S). I-680 is a north to south interstate highway in the San Francisco Bay Area which runs from San Jose, CA in the south and Fairfield, CA in the north which are marked with stars in Figure 1.

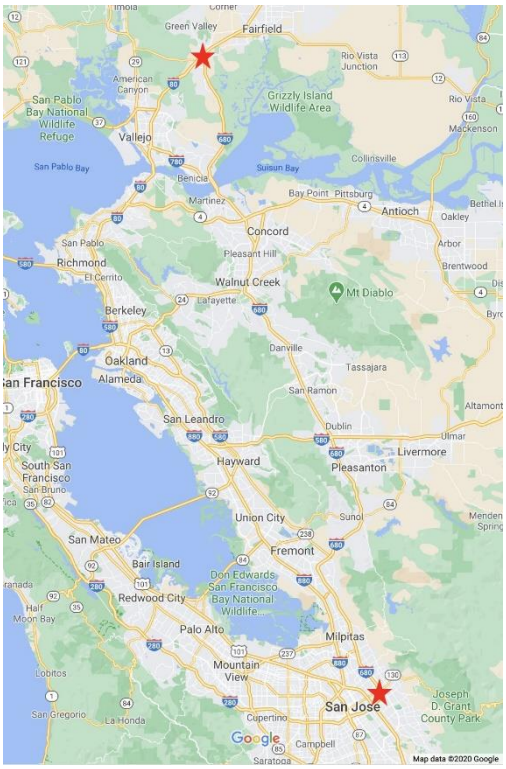

Figure 1. The start and end of Interstate 680 (I-680) as marked.

To further illustrate the issue of sensors that are malfunctioning, the percentage of good and bad detectors as visualized in Figure 2 was generated from the PeMS website based on the record on March 18, 2019, the chart is generated considering all the 514 detectors on I-680 S. Bad detectors are those that seem to malfunction based on the previous day's data.

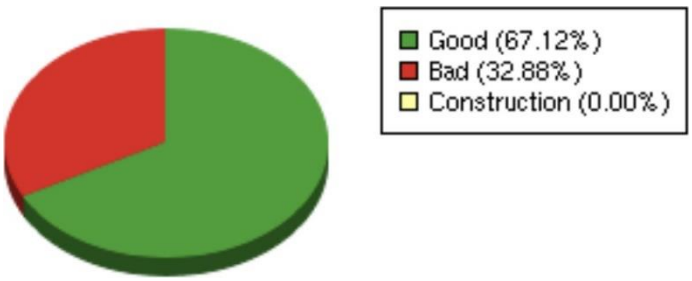

Figure 2. Percentage of good and bad detectors for I-680 S.

The work of Tsekeris and Stathopoulos (2006) inspired the initial method used in this paper. They analyzed traffic data from traffic detectors in several arterial links of an extended urban network. They are able to model the time series of these data using principal component analysis (PCA) and the technique has been proven to provide a "plausible and powerful tool in showing the common patterns of temporal variability". They recognized in their work that the "temporal trend thresholding of the given eigenflow together with the corresponding band of ±2SE is suitable for the long-term analysis of the expected or predictable variations of traffic flow" which can also capture outliers in cases of unexpected events in the network like bus breakdowns and accidents.

This is interesting to investigate given that the current methods being used by PeMS to fill in the gap between the data set is by the use of linear regression from neighbors based on local coefficients, linear regression from neighbors based on global coefficients, temporal medians wherein PeMS looks at data values at similar times and days of the week over a long period of time where the medians of those data values are used to fill gaps and cluster medians (Caltrans 2020; Bickel, et al. 2007).

Furthermore, given that the current imputation techniques require information from the neighbors, it has also been the objective of this study to introduce models that may not depend on the neighbors but features that could be inherent to the VDS itself such as its ID, postmile (which is the relative location/distance from the start of the freeway), and other features such as time and day of the week using linear regression and a more advanced machine learning technique like a random forest. The use of just a few eigenvectors from PCA as features in a linear regression model, if proven to result in high accuracy of the flow estimates can then help reduce the dimensionality of the data required to estimate traffic flow when no data is available at any VDS and timestep. Otherwise, if this does not meet our desired performance, other techniques introduced can be considered useful. Meanwhile, it is also good to note that this study has been limited to having predictions for the aggregated flow for every five (5) minutes.

The remainder of the article is organized as follows: Data which describes and includes the visualization of data used in the study. This is followed by Methods. Next are the Results and the last part is dedicated for Conclusion.

DATA

The data that used in this paper are five-minute aggregated flow data of each day on I-680 S over a two-week period from March 17, 2019 to March 30, 2019. Before downloading them from the PeMS website, which are filtered to consider only the data from the mainline excluding those from the on and off-ramps. The data downloaded from PeMS have the following columns: "Time", "Postmile (Abs)", "Postmile (CA)", "VDS", "AggFlow","# Lane Points", and "% Observed". Time is divided into five-minute time steps within the entire day starting from 00:00 to 23:55. Absolute postmile which has its column named "Postmile (Abs)" indicates the actual distance along a freeway from its beginning to its end. These postmile values increase monotonically with the length of the freeway. Absolute postmiles have been converted by PeMS from the Caltrans Postmiles. Then, it also has the column for the Caltrans Postmiles ("Postmile (CA)") which is then considered as jurisdictional because these are reset to zero at every county line and are assigned to physical boxes and geometric features on freeways when they are built. "VDS" column gives the ID of the vehicle detector stations. We also have the aggregated flow (AggFlow) at each VDS which is the number of vehicles for every five-minutes. Then, the

number of lane points ("# Lane Points") at each VDS location is also given within the data which represents the number of points used to generate the data. The last column is the "% Observed" which tells us whether the aggregated flow has been generated using actual data or partially/fully inferred from the points where there is collected data. The data is then restructured for use in the principal component analysis as described in Section 3.

This gives a total of 580,608 flow observations at five-minute time interval on 144 mainline VDS. But, among these observations, only about 63% is based on 100% observation, 27% was imputed after no observations were recorded, and the remaining 10% was also imputed after observations of less than 100% but greater than 0%. After filtering the rows with less than 100% observed flow at each VDS, we end up with 366269 rows out of the original 580608. Then, 19 out of the 114 sensors are not able to record any flow at all.

In addition, we can generate a time series of flow passing through any given VDS corresponding to a specific section of the freeway. The inventory of these VDS from PeMS website can help you locate this approximately on the map. An example of a time series plot is shown for VDS 418817 (Absolute Postmile 0.200) in Figure 3. The time series can easily help us identify the variation in terms of peak and off-peak hours of traffic and can later be used to compare the reconstructed flow.

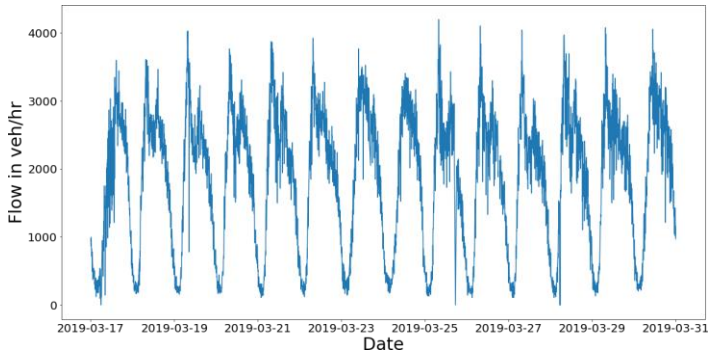

Figure 3. Flow time series at VDS 418817 (Absolute Postmile 0.200).

Also, we used k-means clustering to see if it can easily divide the data into clusters using only the two columns of the data: VDS ID ('VDS') and aggregated flow ('AggFlow'). The no. of clusters used is four (4) based on the elbow rule using the plot in Figure 4 of the variation of the sum of squared distances in the function of the number of clusters. Meanwhile, the results of k-means clustering can be seen in Figure 5. From among the clusters, it is only Cluster 1 which is very easy to interpret showing all the flow when they are below 4000 vehicles per hour, while the other three (3) clusters aren't giving any very intuitive interpretation, this might be because of the limited no. of inputs used in clustering. Despite this, the clusters were used to see if they can add a predictive power in the linear regression model which is almost similar to considering the clusters as described in Caltrans 2020.

METHODS

We first used principal component analysis on traffic flow as adopted from the method introduced in the work of Tsekeris and Stathopoulos (2006). They calculated for eigenflows which they defined as "a time series that captures a common pattern (or source) of temporal

variability in traffic flow at the network level. Each traffic flow time series is expressed as a weighted sum of eigenflows and the corresponding weights reflect the extent to which each source of temporal variability is present in the given traffic flow".

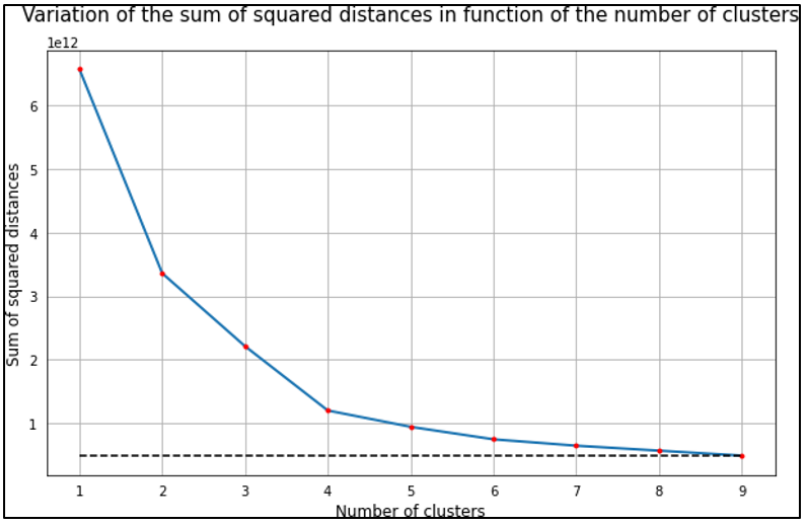

Figure 4. Variation of the sum of squared distances in function of the number of clusters.

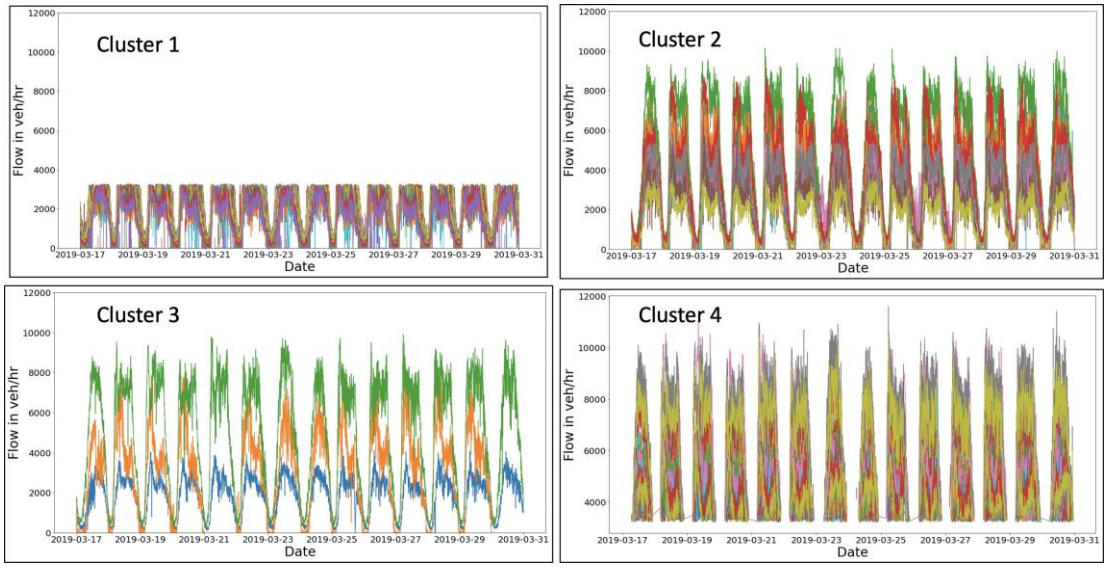

Figure 5. Four clusters with time series of flow.

Similarly, m is used as the number of columns corresponding to the mainline detectors/VDS of I-680 S. Then, t is the number of successive days in which the detector data are collected which in this case is 14 days (i.e. two weeks from Mar. 17 to Mar. 30, 2019), and τ is the number of time intervals where flow data was collected. Then, matrix X with dimension $p \times m$ is defined which is then referred to as measurement matrix where $p = no.of\ rows = \tau \times t$. From here, we have column vectors X_i which then represents the i-th traffic flow time series, and each row j denotes the particular point in the time series in which traffic flows have been collected at

interval *j* which in this case is at every five (5) minutes. In short, one data point (*i.e.*, row) is the flow on every road section for five (5) minutes for a given time of the day while one feature (*i.e.*, column) is the flow on one VDS. After rearranging the data conforming to the description above which results in a final dimension of $3{,}996 \times 95$ (there appear to be 32 missing time steps). This is then divided into train and test set at a ratio of 1:1 to make sure that I can use the eigenvectors from the train set as features into the linear regression to predict flow on the test set. So, the dimension for both subsets is $3{,}996 \times 47$. We needed to exclude the data from one VDS so they have the same no. of columns that address the compatibility issue when adopting the eigenvectors from the train set to the test set. Then, the reconstruction of the training set from a smaller number of eigenvectors is calculated and its overall accuracy is calculated using equation 1, where X correspond to the data, $\bar{X}$ is the mean, and $\hat{X}$ is the reconstruction value from PCA.

$$Accuracy = 1 - \sqrt{\frac{\sum_{i=1}^{96}(X - \hat{X})}{\sum_{i=1}^{96}(X - \bar{X})}} \tag{1}$$

Then, the histogram of the flow data is visualized in Figure 6. It can be seen from the histogram that there are many flow observations that fall within the range of lower values (skewed to the right). Hence, the data was standardized by removing the mean and scaling to unit variance which was done using the 'StandardScaler' module of sklearn (Pedregosa, et al. 2011). It calculates the standard score of sample *x* by having *z* = (*x* - *u*) / *s* where *u* is the mean of the training samples or zero if *with_mean* = *False*, and *s* is the standard deviation of the training samples or one if *with_stdev* = *False*. The resulting histogram after standardization of data is shown in Figure 7.

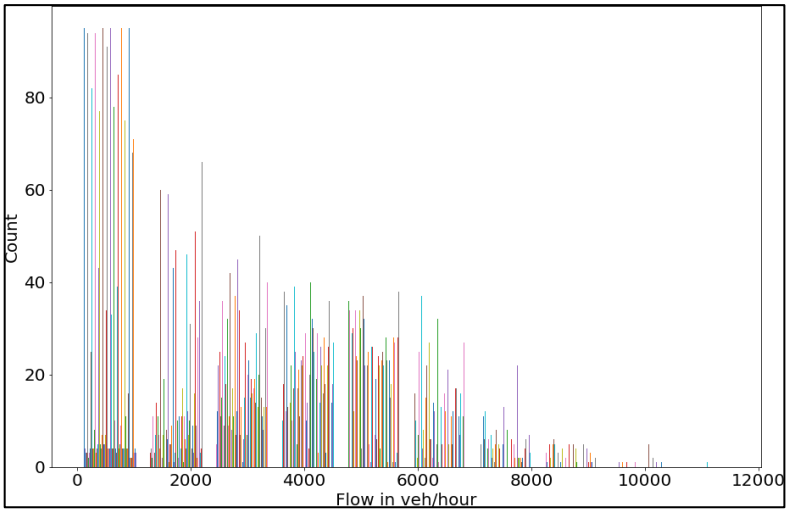

Figure 6. Histogram of flow data (no. of bins = 10).

Using only the no. of eigenvectors required to explain at least 90% of the variance, we trained a linear regression model using the eigenvectors and This might be considered as a naive approach but as a baseline, we are interested to see if this somewhat has a predictive power. The linear regression equation using PCA has its equation described in equation 2.

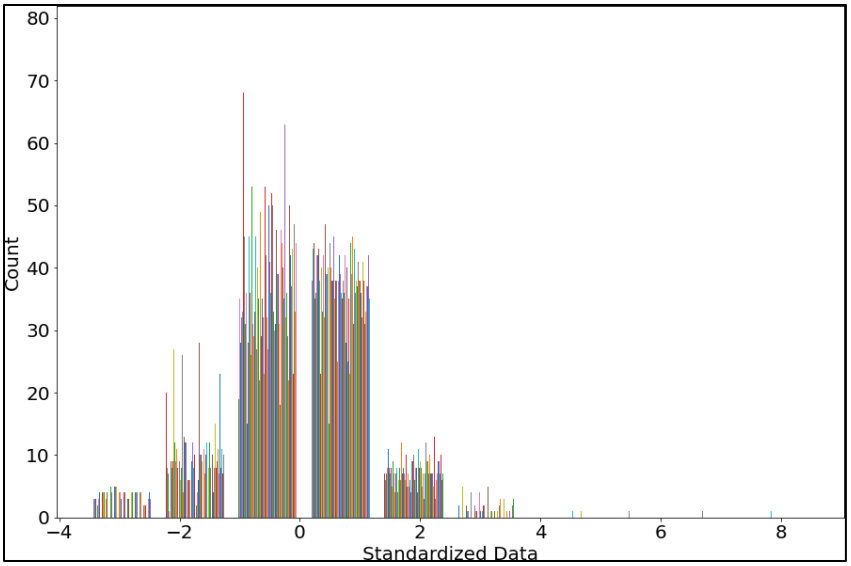

Figure 7. Histogram of standardized data (no. of bins=10).

$$AggFlow_i = \beta_0 + \beta_j \sum_{j=1}^{n} value\ from\ eigenvector_{ij} + \epsilon \quad (2)$$

In equation 2, *i* represents the row location of the value from the corresponding eigenvector based from *j* which represents the order of eigenvectors while *n* is the no. of eigenvectors used in the model.

Meanwhile, for the other machine learning techniques, the original structure of the data with columns described in section 2 was used with an added column corresponding to the day of the week as another feature. This time, 80% of the data was used for training while 20% for testing. We can note in the linear regression equation below that time and day of the week are both treated as categorical variables.

$$AggFlow_i = \beta_0 + \beta_1(VDS) + \beta_2(Abs\_postmile) + \sum_{j=3}^{299} \beta_j Time_j + \sum_{k=300}^{305} \beta_k Day_k + \epsilon \quad (3)$$

Similar form of the equation above was used in another linear regression model where we added the resulting clusters from k-means clustering that was previously applied as described in section 2 to see if it improves the performance of the model. The corresponding linear regression equation is shown in equation 4.

$$AggFlow_i = \beta_0 + \beta_1(VDS) + \beta_2(Abs\_postmile) + \beta_3(Cluster) + \sum_{j=4}^{300} \beta_j Time_j + \sum_{k=301}^{306} \beta_k Day_k + \epsilon \quad (4)$$

The regression type of random forest is also introduced to see if a more advanced machine learning technique would be able to generate a higher accuracy on both train and test set. The module 'sklearn.ensemble.RandomForestRegressor' (Pedregosa, et al. 2011) was used with its default parameters where we have the number of trees in the forest being equal to 100. This method uses random decision forests as an ensemble learning method, regression in this case, where we use the mean or average of the individual trees to generate prediction. This model is

known to be hard to interpret but usually results to high level of accuracy when applied on the test set. The features considered in random forest is similar to that of the second linear regression model with its features described in equation 3.

'Score' was the metric chosen to evaluate all the four models considered on this paper. The score of the models are calculated for both the train and test set for each model as the coefficient of determination (R^2) of the prediction. Root mean square error between the predicted and true values of each model on the test set are also reported in this paper.

RESULTS

On the left of Figure 8 is the cumulative sum of the variance explained as we increase the number of components used in PCA. It can be noticed that it takes at least 4 components to explain at least 90% of the variance, with principal component (PC) 1, PC2, PC3, and PC4 explain 79.82%, 4.76%, 4.44% and 1.88% of the variance respectively. On the right of Figure 8 are the corresponding plots of PCs 1, 2, 3, and 4. The plot gives us an idea how the detectors are compensating for the variations in traffic flow based on each eigenvector.

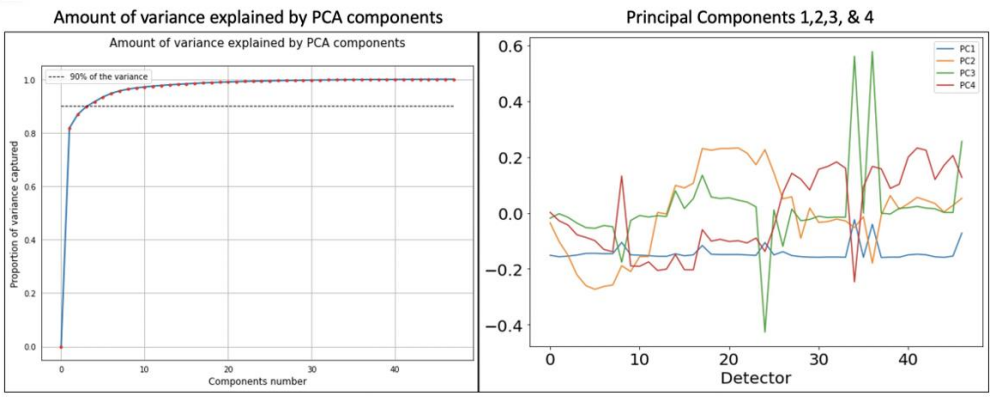

Figure 8. Cumulative sum of the variance explained at increasing no. of components *(L)*; Principal components 1, 2, 3, and 4 *(R)*.

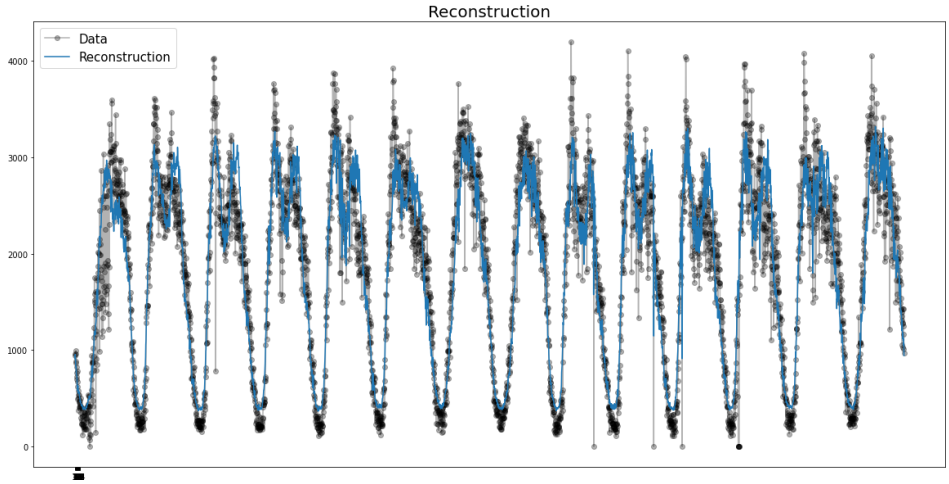

Figure 9. Example of flow reconstruction for one VDS using 4 eigenvectors.

Then, in Figure 9 is an example of flow reconstruction in one VDS from the training data using four (4) eigenvectors from PCA. Only four (4) was used since it's enough to explain at least 90% of the variation in flow. The overall accuracy of the flow reconstruction is calculated being equal to 0.7254.

Meanwhile, as discussed within section 3, we developed three (3) linear regression models and one random forest model. Their performance based on the corresponding score for both train and test set of each model is summarized in table 1 and the biplots of the predicted and true values of the test set is shown in Figure 10.

Table 1: Comparative summary of scores and RMSE of the models

Model	RMSE	Train score	Test score
Linear regression with PCA	1945.68	0.9120	0.1372
Linear regression w/ select features	1385.65	0.6130	0.6120
Linear regression w/ clustering	1166.22	0.7263	0.7252
Random forest (no. of trees = 100)	1089.06	0.9513	0.7595

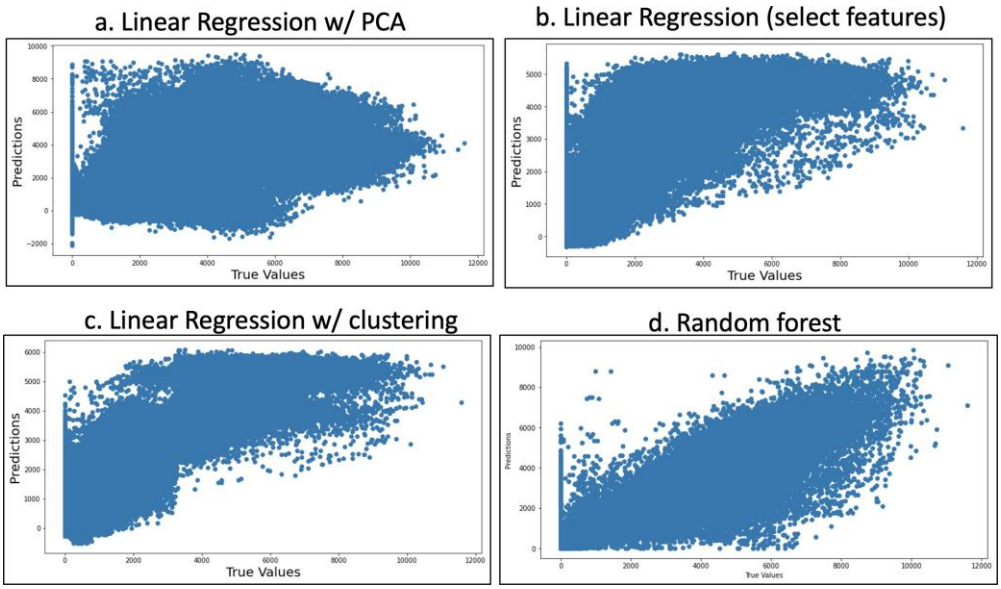

Figure 10. Biplots of predicted vs. true flow values of the test set.

It can easily be noticed that random forest stands out in terms of all the metrics considered as seen in table 1. Also, the biplot of random forest in Figure 10.d. is apparently the best in comparison with the three other models. But, its performance isn't too far from the linear regression model where we added the cluster from k-means as feature considering both the RMSE and test score making it the second best performing model among the four. Meanwhile, the model which performed the least on the test set is the linear regression where we used the values from the eigenvectors of the training as features but its train score comes second to random forest in terms of performance.

In addition, I also illustrated in Figure 11 the feature importance to see which among them matters the most based on the random forest model, making the the top five are postmile, VDS, when it's a weekend (Saturday and Sunday), and when the time is 01:55 AM.

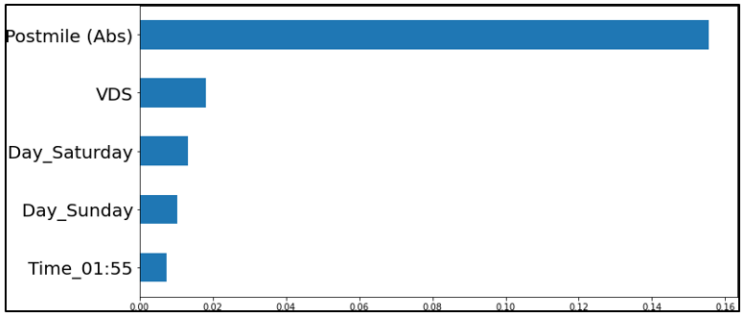

Figure 11. 5 Most Important Features.

Finally, illustrated in Figure 12 an example time series of the predicted flow for one of the faulty VDS which in this case is for VDS 407194. It recorded 0% observation for the two-week period covered. We can see that the prediction is able to capture almost the same variability or pattern in flow when compared to Figure 3.

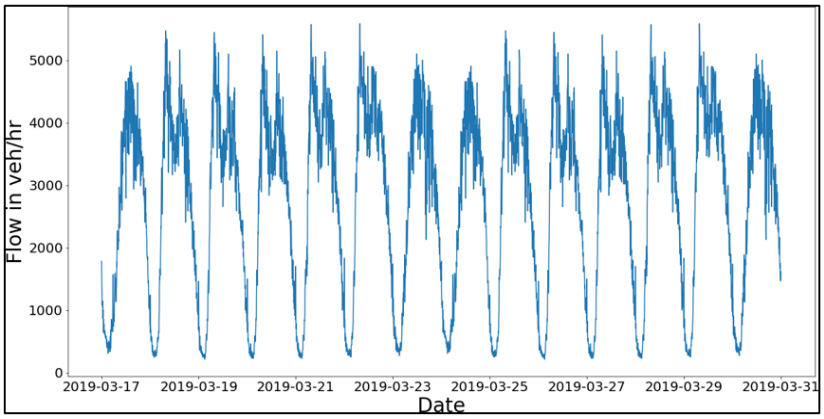

Figure 12. Time series of predicted flow for VDS 407194.

CONCLUSION

The linear regression using the eigenvectors from the training set as features to predict flow on the test set was found to overfit the training set. Meanwhile, the linear regression model using time, absolute postmile (relative location of VDS from the start of the freeway), VDS ID, and day of the week as independent variables performs much better on the test set compared to the former. This is further improved when the clusters from k-means were added as feature to the model.

But, random forest with just the default parameters performs the best among the models indicating that 'Postmile (Abs)' and 'VDS ID' as the two most important features. Meanwhile, it appears that distinguishing that the flow is gathered on a weekend, Saturday and Sunday, would be an important feature as well. This makes sense given that in most cases, flow distribution on weekends differs from weekdays.

Despite performing the best on both train and test set, we noticed that the random forest has the longest training time. It took about 37 minutes 48 seconds for it to finish running the cell

containing the random forest model (training and testing) while it was just 5.11 seconds for the linear regression with clustering to finish using a 2018 Macbook with 2.6 GHz 6-Core Intel Core i7 and 16 GB RAM with default settings on Jupyter Notebook ran in Google Chrome. So, it may be worthy to revisit if we care for the training time which could depend on how often we may want to train a new model from the data especially the difference in terms of accuracy on the test set between the two models are not too far.

Some future work after making this project could be addressing the lengthy training time for random forest and evaluating the performance of the model, which if we consider the best performing model here would be the random forest, using other freeways. Also, it might be of interest for many to consider other ways of using the features; like instead of having the days of the week, we may try classifying whether it's a weekday or a weekend; also try other machine learning algorithms to see if they can further improve the accuracy especially on the test set.

Lastly, among the interesting things we discovered is that PeMS generate so much data for all the freeways across California. Scaling this project to all freeways can be a big challenge. In that case, one must really be familiar with its design architecture to be able to successfully implement this kind of study and include it within the features of PeMS.

ACKNOWLEDGEMENTS

Alben Bagabaldo is thankful for the support of his Ph.D. adviser, Prof. Alexandre Bayen. He is also thankful to the Philippine Commission on Higher Education – Philippine-California Advanced Research Institutes (CHED-PCARI) and Department of Science and Technology - Science Education Institute (DOST-SEI) for funding his studies at UC Berkeley. Alben is also thankful to Mapua University for the additional financial support extended to him for his Ph.D. studies.

REFERENCES

Bickel, P. J., Chen, C., Kwon, J., Rice, J., Van Zwet, E., and Varaiya, P. (2007). Measuring traffic. *Statistical Science*, 22(4):581–597.

Caltrans. (2020). PeMS User Guide. Technical Report February.

Garber, N. J., and Hoel, L. A. (2020). *Traffic and highway engineering, enhanced* 5th ed. Cengage Learning.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Tsekeris, T., and Stathopoulos, A. (2006). Measuring variability in urban traffic flow by use of principal component analysis. *Journal of Transportation and Statistics*, 9(1):49–62.