Analysis of delays on the New Jersey railway network

Rami Azouz, Linah Charif, Jasso Espadaler Clapés, Lynn Fayed
Network Tour of Data Science
January 10, 2020

I. Introduction

Evaluating delays and identifying their causes in railway networks are very critical topics given the detrimental cost that delays have on the economy. In analogy with the epidemiological spreading model, existence of hubs in every railway network results in a faster propagation of delays, mainly because the majority of network lines converges or diverges from the central stations. As a consequence, the first objective of this project is to assess whether values of delay at stations radially vary as we move further towards and away from the main hub of a railway network. The basic assumption behind this proposition is that trains can compensate delays on terminal stations by increasing their speeds. This is nevertheless impossible to achieve at central stations where high train frequencies imply a greater speed restriction. Furthermore, the values of delays are also dependent on the hour of the day, and are assumed to significantly vary between the morning and evening peaks.

An overwhelming amount of datasets in the field of transportation is available on open platforms. After a thorough research of the multiple existing datasets, we chose to base our project on the NJ Transit + Amtrak (NEC) Rail Performance dataset available on Kaggle. The New Jersey commuter network (Figure 1) is the second largest railway network in the United States in terms of number of passengers. It serves both the area of New Jersey state and New York, and allows commuting between the two zones. The dataset allows to assess the delays in the network and to fulfill the eventual objective; to predict its value at discrete moments in time by resorting to machine learning tools.

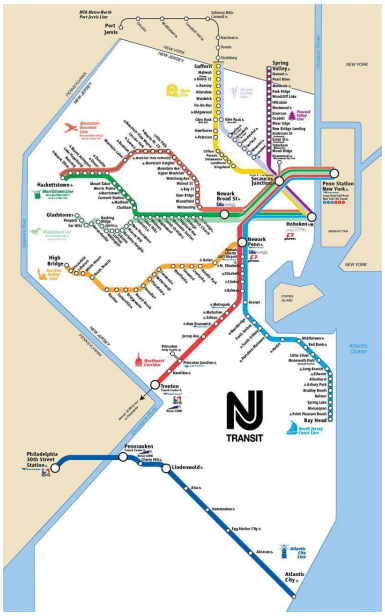

Fig. 1: Nj transit map

II. Preprocessing

A. Data Structure

The structure of the dataset provided has the sufficient tools required to assess the overall performance of the network. This assessment is essential given the great importance of the railway network for the New York area. The dataset used for the purpose of this assessment contains detailed information on the trains operating in the rail network [1]. Indeed, more than 287,000 train trips for the NJ transit network are available at stop-level and with minute resolution for the schedules (expected arrivals) and delays. The data covers train trips from March 1, 2018 to April 30, 2019. Table I gives an insight about some general observations in the network only for the month of March 2018 (each observation is a train journey between two stations).

No. of Observations	# of Stations	Lines	Trains
243028	165	11	1319

TABLE I: Summary of the data structure for a given month

One major challenge to assess the delays distribution is to find a metric capable of reflecting the variations of delays along stations, and along the time of the day. The original data structure provides information about point delays, that is delays for each train at each station of the journey. These delays are expected to immensely vary along two different dimensions: direction of trains at a certain station, and the hour of the day for which the delays are assessed. To better understand how to aggregate the delay measures at different node stations, we initiated the work through assessment of the performance of the delay means, medians, sums, and standard deviation over the month of March 2018. The primary objective is to choose a metric that will accentuate the differences in node delays along stations to be adopted later on for clustering. Figure 2 shows the sorted monthly average delays along the different nodes (1 to 7 minutes approximately).

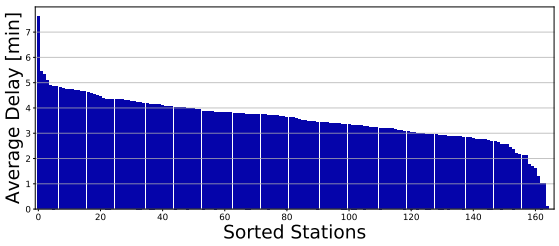

Fig. 2: Monthly average delays along stations

Because average seems to provide a rather uniformly varying distribution of delays in stations, it is rational to use it as a metric for further assessment. In addition to the average delay, the sum of delays over the month also demonstrates a rather uniform distribution of the values along the different nodes. Furthermore, the sum of delays over the month gives an indirect measure of the number of trains crossing each station, and thus, its importance inside the network.

Another useful step is data preprocessing is to visualize the average delays as weights for the different nodes in the network to identify the stations that are highly problematic. As anticipated, the terminal nodes mainly located to the left of the graph (Figure 3) generally have the lowest delay levels. This can be substantiated by the fact that these nodes represent terminal stations away from the center (here located on the upper left part of the graph). The intermediate nodes are proven to have a relatively high delay, and the reason behind this observation will better justified in the later stages of this report.

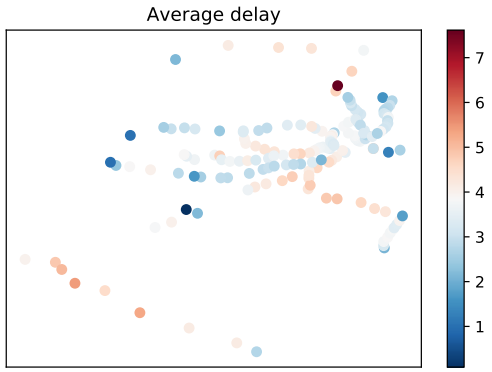

Fig. 3: Average delays as weight of nodes in the network graph

B. Data filtering

As previously mentioned, the uniqueness of the chosen data is that it is time-dependent. This means that delays significantly vary along the time of the day. Consequently, a more accurate representation of the network state must consider the timely variations per stations. The data manipulations that will be shown throughout this project will only consider weekdays (Monday to Friday). In addition, since we are interested in showing how the number of trains coming from different lines and meeting at the central station nodes influence the delays at these nodes, we will only consider interconnected lines. For that purpose, two lines have been removed from the dataset: the Atlantic City Line and the Princeton Shuttle.

Moreover, since the delay level depends on the train direction, filtering the data into inward and outward observations is crucial. After separating the trains into the ones coming from and the ones heading towards the center, it was possible to dynamically plot the hourly average delays per stations over a period of a month for the ingoing and outgoing trips. The dynamic plots are provided in the github repository.

III. Clustering

One major assumption in railway networks is that delays at stations are highly correlated with train frequencies. For the purpose of verifying this speculation, it was interesting to cluster the data based on:

1) The total number of trains passing by a given station over the course of a month.
2) The cumulative delays at each station over the same period of analysis.

This project has explored two different clustering methods: spectral clustering and k-means.

A. Spectral Clustering

The construction of the graphs is based on the similarity of cumulative delays or number of trains crossing each station. The similarity is calculated with the Euclidean distance between data points. A weight is computed for each pair of nodes (Eq. 1), and if the weight is above a certain threshold (ϵ) the two nodes are connected in the graph. Tuning these parameters is essential to create an initial graph with one connected component. However, the nature of the data does not allow to simultaneously come up with a sparse graph with only one connected component.

$$w_{ij} = \exp\{-|x_i - x_j|/2\sigma^2\} \tag{1}$$

Figure 4 shows the clustering of the data based on the cumulative delays and the number of trains crossing each station. In the context of our data, only two clusters seem reasonable. However, Figure 4 shows four clusters. The reason behind this choice is that it makes it possible later on compare the results with the k-means method used below. If the number of clusters is above two, the clustering results yields an irrational distribution of the groups through the network. This is essentially the case for the four clusters of the graph based on cumulative delays. For the graph based on the number of trains, it can be observed that the central stations behave differently than the rest of the network.

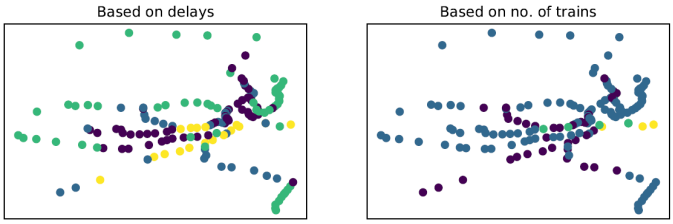

Fig. 4: Spectral clustering based on number of delays and sum of delays

B. K-means

As underlined in previous sections, the time-dependent nature of the data implies the importance of averaging the hourly delay per station, and observe their variations over the course of the day. Consequently, k-means clustering has been used to track down how clusters will vary over day for the inbound and outbound trains. These graphs help the operators identify problematic stations as well as get an insight on how delay propagates in the network [2]. Again, the results of the dynamic plots are provided in the github repository.

The results of the clustering are shown in the Figure 5 below. We note that the method used here is k-means clustering because it has been shown to perform rather well compared to spectral clustering. The reason why four clusters have been chosen is that this number assign closely neighboring nodes to the same clusters. An increase in this value will lead to a more chaotic delay graph, and hence the loss of the distance dependent peculiarity of the resulting clusters. When exclusively considering the clusters based on delays, it can be noted that the majority of the terminal stations belong to the green or yellow clusters, whereas the central ones are represented in blue. In between, the clusters are represented in purple.

A comparison between the two graphs demonstrates of a generally remarkable similarity between the clusters of the two graphs. This validates the initial assumption that train frequencies are de facto connected to cumulative delay levels in the network even if some differences are still well observable.

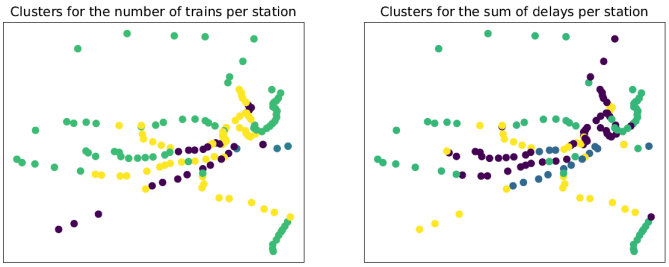

Fig. 5: Clustering based on number of trains and sum of delays

IV. Prediction of Delays

An additional task that we would like to apply on our dataset is to predict the final delay of a train, given its delays at previous stops. Several machine learning methods have been proposed and tested in order to achieve this task.

A. Recurrent Neural Network with LSTM

The intuition to consider a Recurrent Neural Network at a first stage is the temporal dependency aspect of our data. In fact, the accumulation and propagation of delays from the beginning to the end of a line will depend on the punctual delay at each station. However, the overall delay is not the sum of all the delays at each node. Indeed, the delay between two stations can be compensated at the next stop by a change of speed for instance. Since the patterns are progressively learned in time through RNN, we thought that the use of this tool could be adapted to such task.

1) Preprocessing: The particularity of RNN is that data should be ordered in a particular way before being fed to the model. The input of the model should be a tensor of shape [#observations, #timesteps, #features]. Adapted to the case at hand, this results in: [#observations, #previous stations, #features]. We chose two features for our model : the delay at the previous stations and the actual time at which the train arrives at the station.

2) Model: For the sake of simplicity, the model has only been trained on one line: the Northeast Corridor. The model was trained on the delays at the four previous stops, using the data of the whole month. All the stations of the line were used to train the RNN except for the station Metropark, close to the terminus. This station has served has the test of the model.

3) Result: While training the model, we noticed that the loss was not minimized and we could not achieve more than 20% accuracy, which means that the model was predicting the delays randomly. The possible justification for this deficit in accuracy could concern our choice of features, and how we selected our data for training. As a matter of fact, the notion of delay is not solely correlated with the time of arrival at the station and the previous delays on the same line, but it is the consequence of multiple factors related to the contribution of other lines in the stations delay.

On another hand, tuning RNNs is not an easy task because there of the absence of numerous parameters to modify.

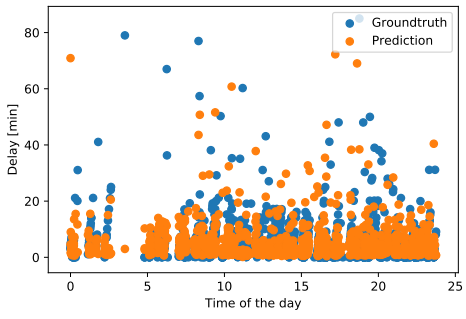

Fig. 6: Prediction of delays at stop ”Metropark”, on line ”Northeast Corridor” for outward trips

Nevertheless, we do not know in which direction the model could be improved, and its architecture should be built in attempt to achieve better results than the ones obtained in Figure 6.

Given the complexity and poor initial results of the RNN, the choice was to abandon it to achieve the task previously described. Alternatively, we decided to consider the problem as a classification task using a more practical model: a simple Neural Network.

B. ANN

As discussed previously, the model built did not succeed in having an accurate prediction of the delay at a given hour in a given line. This is the reason the focus was diverted towards simple Artificial Neural Network (ANN) model. We believe that an ANN can achieve an accurate prediction if its parameters are very well tuned. However, the prediction task was time consuming given the continuous nature of delays and the model failed to achieve desired results. Therefore, we decided to bring the prediction problem into a classification one. Our input consists of the considered train_ID, its stop sequence, the station_ID, the line, the day and finally the time in minutes. Before feeding the input to our network, we made sure that each feature lies between the interval [0,1].

1) Binary Classification: We started by labeling the data according to delays: if the delay is less than 1 minute, we assume that the train will arrive at time. Otherwise, we assume that arrivals in stations exceed the expected arrival times. Our ANN model consists of 1 input layer, two hidden layers (12 and 8 neurons) and finally an output layer.

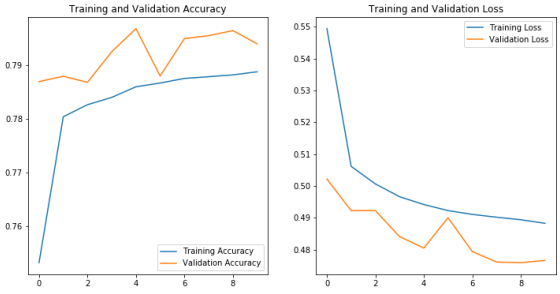

Fig. 7: Binary Classification: Accuracy and loss for both training and validation data

We trained the model on the month of March 2018, and performed the prediction on the month of June 2018. The achieved accuracy is 80%.

2) Multi-label Classification: After successful implementation of the binary classification, we tried to predict if there will be any long delays. To integrate this additional variation in the model, it was necessary at a first step to get proper insight on the distribution of delays. The conclusion reached is that delays in the NJ railway network follows a power law distribution. Therefore, the logarithm of the delays was assumed, and we proceeded with the classification in the following order: No delays, delays less than 7 minutes and finally delays greater than 7 minutes. We chose this partition in order to have the same number of data for each label. In this model, the implemented ANN consists of an input layer, 3 hidden layers (32, 16, and 8 neurons) and an output layer. In addition, we transformed the labels so they lie between [0,1]. Similarly to the previous classification, we trained the network

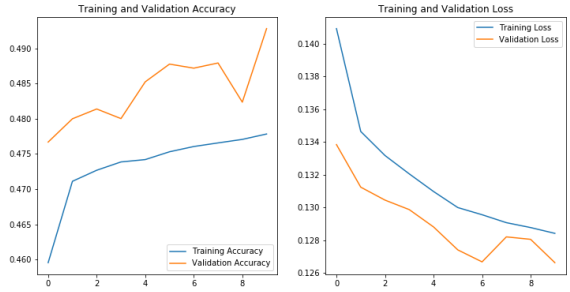

Fig. 8: 3 Labels Classification: Accuracy and loss for both training and validation data

on one month, and evaluated the model on unseen test data. We achieved an accuracy of 49%. Nevertheless, the room for improvement is considerable.

C. Ridge Regression

On a final note, we tried to compare the previous results with a ridge regression model. For that, we performed some feature expansion (sin(x), cos(x), and exp(x)) followed by a polynomial expansion (degree 12). As expected, we had a poor accuracy (14%) due to the complexity of the problem at hand.

V. Conclusion

Delays generally constitute a significant cost for the economy. A proper understanding of their distribution and propagation in the network is hence necessary to put forward powerful strategies to improve the performance of the railway network. Clustering is a potential approach that possibly allows identification of problematic stations where delays originate, and how they propagate over time. Moreover, a link between the number of trains at each station and the level of delays has been established. The findings have shown that k-means performed slightly better than spectral clustering in terms of radial grouping of delays. The final part of this report assessed the possibility to predict delay levels using different machine learning methods. This approach can serve as a passenger aid tool to anticipate the expected delays for a person leaving his origin at a specific hour in the day. Nevertheless, although the final ANN model returned a good accuracy level for the binary classification of delays, further amelioration can be assessed to increase model complexity.

References

[1] NJ Transit + Amtrak (NEC) Rail Performance. https://www.kaggle.com/pranavbadami/nj-transit-amtrak-nec-performance.

[2] Fabrizio Cerreto, Bo Friis Nielsen, Otto Nielsen, and Steven Harrod. Application of data clustering to railway delay pattern recognition. *Journal of Advanced Transportation*, 2018:1–18, 04 2018. doi: 10.1155/2018/6164534.