International Journal of Engineering, Business and Management (IJEBM)
ISSN: 2456-7817
[Vol-7, Issue-1, Jan-Feb, 2023]
Issue DOI: https://dx.doi.org/10.22161/ijebm.7.1
Article DOI: https://dx.doi.org/10.22161/ijebm.7.1.5

Predicting Operating Train Delays into New York City using Random Forest Regression and XGBoost Regression Models

Thomas Wiese

SUNY Empire State College, New York

Received: 15 Jan 2023; Received in revised form: 11 Feb 2023; Accepted: 20 Feb 2023; Available online: 28 Feb 2023

Abstract— *The Long Island Railroad operates one of the largest commuter rail networks in the U.S.[1]. This study uses data which includes the location and arrival time of trains based on onboard GPS position and other internal sources. This paper analyzes the GPS position of the train to gain insight into potential gaps in on time performance and train operations. This was done by developing a Random Forest Regression model [2] and an XGBoost regression model [3[. Both models prove to be useful to make such predictions and should be used to help railroads to prepare and adjust their operations.*

Keywords— ***Random Forest Regression model, XGBoost Regression Model, Machine Learning, Operations Management, Management, Business Analytics, Analytics, Industrial Internet, Industrial Internet of Things, Trains, Train Delays, Decision Tree***

I. INTRODUCTION

Train delays have become a common occurrence in the United States. Delayed trains can cause significant disruption to travelers, resulting in lost time and money. The causes of train delays are varied and complex, ranging from mechanical issues to weather conditions [4]. Understanding the various factors that contribute to delayed trains is essential for improving service reliability and reducing passenger frustration.

One of the most common causes of train delays is mechanical issues with equipment or track infrastructure. Train operators must perform regular maintenance on their vehicles and tracks to ensure safe operations, but even well-maintained systems can suffer from faults or breakdowns that cause disruptions. Poorly maintained equipment can lead to more frequent malfunctions [5], leading to further delays for passengers waiting on platforms or stuck inside stalled carriages.

Another major factor contributing to train delays is extreme weather conditions such as heavy snowfall or strong winds which can slow down operations significantly if they occur while services are running [6]. During these events, workers must take extra precautions when operating trains due to decreased visibility and slippery surfaces, causing them to travel at slower speeds than usual until it becomes safe enough for normal operations again. Additionally, flooding caused by severe rainstorms may damage tracks and cause delays.

Machine learning has revolutionized the way in which train delays can be predicted. By utilizing complex algorithms and data analysis techniques, machine learning has enabled a more accurate estimation of when a train will arrive at its destination. For example, by combining weather conditions, rail track conditions, historical traffic patterns, and other factors related to train operations, machine learning can accurately predict the arrival time of a train with a higher degree of accuracy than was previously possible [7]. Additionally, machine learning also enables rail operators to recognize patterns in traffic flow and adjust operations accordingly. This allows for better utilization of resources, improved efficiency, and ultimately lower costs for both the operator and its customers.

At the Long Island Railroad, train delays occur between stations due to several factors. Train delays can have a negative

impact on customer experience and cause a negative reputational impact. For this study, GPS train location data and the LIRR train schedule will be compared to determine areas that experience delays at a high frequency. Further, train delays when heading West during the AM peak for trains which arrive at Penn Station are of particular concern. This work developed a machine learning model which predicts how late individual trains will arrive at Penn Station.

II. MATERIALS AND METHODS

The key metric or target that we are interested in is OTP, otherwise known as on time performance in seconds [8]. When a train arrives at a station, the time of arrival is compared to the scheduled time, and an on time performance value is generated under GPS_ARR_OTP column within the data. People are often concerned with when they will arrive at Penn Station while traveling westbound during their morning commute. They are usually worried about being late to work and are trying to predict their arrival time so that they can inform their employer if they will be late.

The relevant features of interest will be the location that the train departs from. We will also use the day of the month, and the day of the week. We're also interested in the run path of a particular train. That is, we'd like to group trains that stop at the same stations, therefore we'll create a grouping filter called RUN_PATH_ID. Since we are particularly interested in the peak travel times, we'll also use whether a train traveled during a peak time 0600-0900 or 1500-1800.

The performance metric for the model is Root Means Squared Error or RMSE [9], which is calculated by:

$$\mathrm{RMSE}_{fo} = [\sum_{i=1}^{N} (z_{f_i} - z_{o_i})^2 / N]^{1/2}$$

Σ = summation ("add up")

(zfi – Zoi)Sup>2 = differences, squared

N = sample size.

Or simply:

$$RMSError = \sqrt{1 - r^2} SD_y$$

Where SD is the standard deviation of y.

Much of the data is not relevant to the analysis and contains a significant number of null values. Over the course of the analysis, columns that aren't needed to answer the key question are dropped, and null values are filled where possible [10]. Following that, data types are converted and string values [11] are encoded to facilitate machine learning. The final dataset contained no null values, only had relevant columns, and was optimized for machine learning by enumerating numeric variables.

The features in this data are seen in Figure 1.

```
df_forML[['LOCATION_NUM','RUN_PATH_ID','MONTH','PEAK_NUM','WEEKDAY']]
```

Fig.1: Data features.

LOCATION_NUM is the enumerated value for the station where the record was generated. RUN_PATH_ID is the number assigned to a group of trains that stop at the same stations. PEAK_NUM is a binary value which indicates if the train traversed it's run path during a peak time. Some summary statistics of these data are seen in Table 1.

Table 1. Summary statistics.

	RUN_PATH_ID	MONTH	LOCATION_NUM	SEQUENCE_NUM	GPS_ARR_OTP	PEAK_NUM	WEEKDAY
count	410663.000000	410663.0	410663.000000	410663.000000	410663.000000	410663.000000	410663.000000
mean	194.543869	2.0	254.624286	38.249555	-40.368833	0.719624	0.229935
std	138.145580	0.0	142.640628	24.937156	172.395363	0.449183	0.420792
min	28.000000	2.0	8.000000	0.000000	-20037.000000	0.000000	0.000000
25%	63.000000	2.0	130.000000	17.000000	-13.000000	0.000000	0.000000
50%	135.000000	2.0	251.000000	36.000000	0.000000	1.000000	0.000000
75%	367.000000	2.0	370.000000	55.000000	0.000000	1.000000	0.000000
max	415.000000	2.0	484.000000	106.000000	1057.000000	1.000000	1.000000

A sample of the head of this data are found in Table 2.

Table 2: First five rows of the data set.

	RUN_PATH_ID	MONTH	LOCATION_NUM	FULL_TRAIN_NUM	SEQUENCE_NUM	GPS_ARR_OTP	PEAK_NUM	WEEKDAY	CREW_NUM
0	197.0	2	8	1001	0	0.0	0	0	173
1	197.0	2	9	1001	1	0.0	0	0	173
2	197.0	2	289	1001	2	0.0	0	0	173
3	197.0	2	291	1001	3	0.0	0	0	173
4	197.0	2	292	1001	4	0.0	0	0	173

Random Forest Regression [2] and XGBoost linear regression [3] were used to predict the on-time performance of a particular run path. The features were location, run path, day of the week, and peak or off peak, and the target will be "on time performance," or seconds late. Month of the year was also considered a feature; however, this data is for only one month. Additional months are available for future analysis.

Regression is the best choice for this problem, since the target is continuous. Further, linear regression is ill suited for this task since the data does not appear to contain any linear relationships.

The RFR algorithm was chosen because it was most effective in similar studies. RFR generated the lowest error for other engineers, although they had a smaller data set due to using minutes rather than seconds [12]. The key RFR hyperparameters are n_estimators and max_depth. N_estimators indicate the number of trees in the forest, and max_depth is how many levels deep the decision tree will go.

It's important to note that if a train arrives within 2 minutes late it is still considered on time at Long Island Rail Road. Therefore producing an error that is less than two minutes late is likely impossible. Further, an XGBoost linear regression model was used as a benchmark against a Random Forest Regression decision tree model. The XGBoost model was trained on the same features to predict the same targets as the RFR model and produced an RMSE of 168.72147644880567 seconds.

Finally, the method included several steps:

1. Read in and wrangle the data
2. Check that visualizations are appropriate
3. Preprocess the data to optimize for ML, remove strings and replace with ints, or floats
4. Split into training and testing sets
5. Train the model
6. Predict on the testing set
7. Measure the accuracy
8. Tune the hyperparameters

III. RESULTS

3.1 Exploratory Visualization

This section shows some clear groupings associated with LOCATION_NUM, RUN_PATH_ID and GPS_ARR_OTP

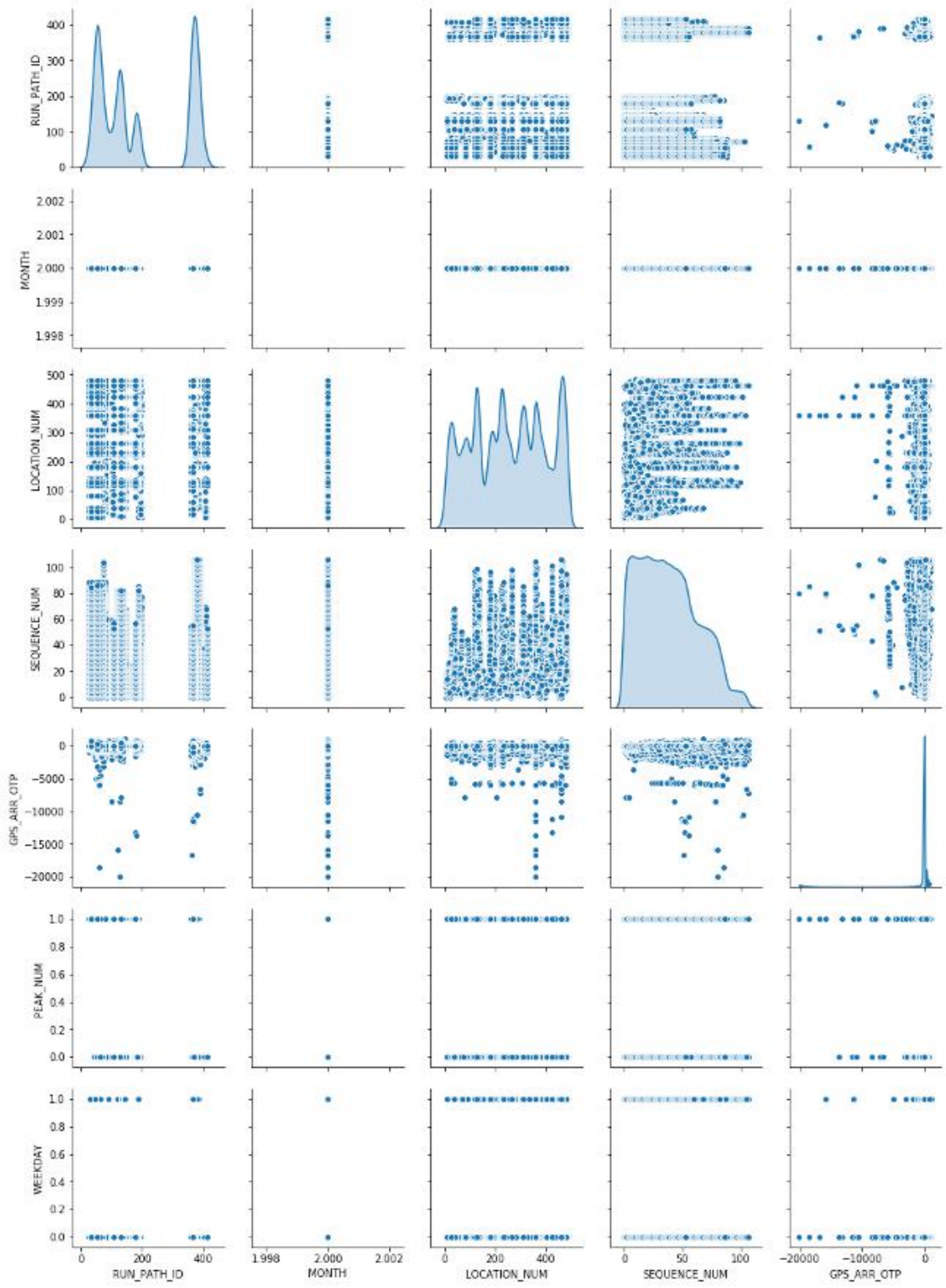

Fig.2. Train GPS data sub plots.

Generally, it's clear that delays occur more frequently in areas where trains more frequently traverse.

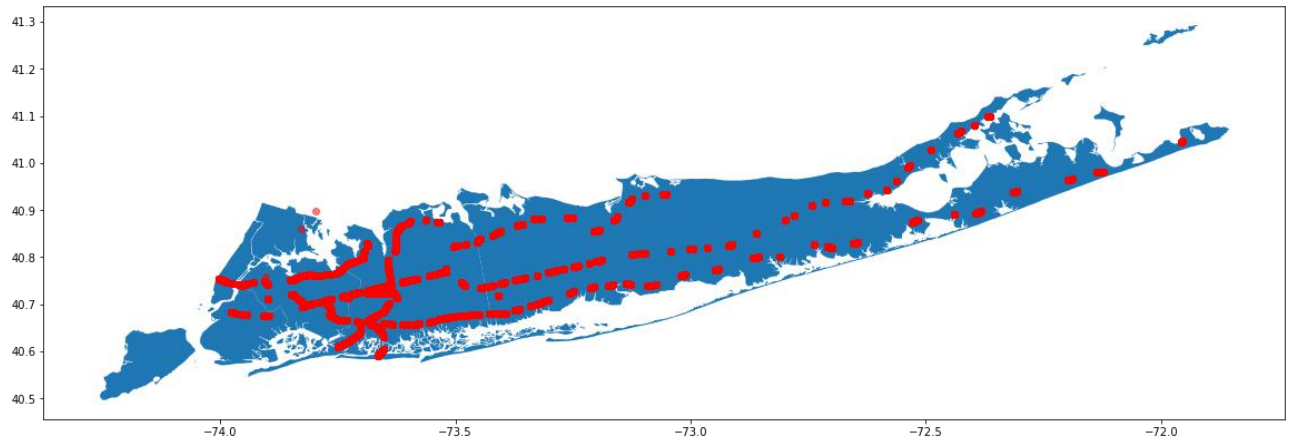

Fig.3. Geospatial analysis of train delay occurrences.

Using the same dataset, we are also able to plot on time performance for individual trains over a period. This is useful in determining that some trains have median average OTP that is more than two minutes late. From this you can also assert that the train is never on time to Huntington, Greenlawn, and Northport, and therefore either the speed, or the schedule, should be changed (Figure 4):

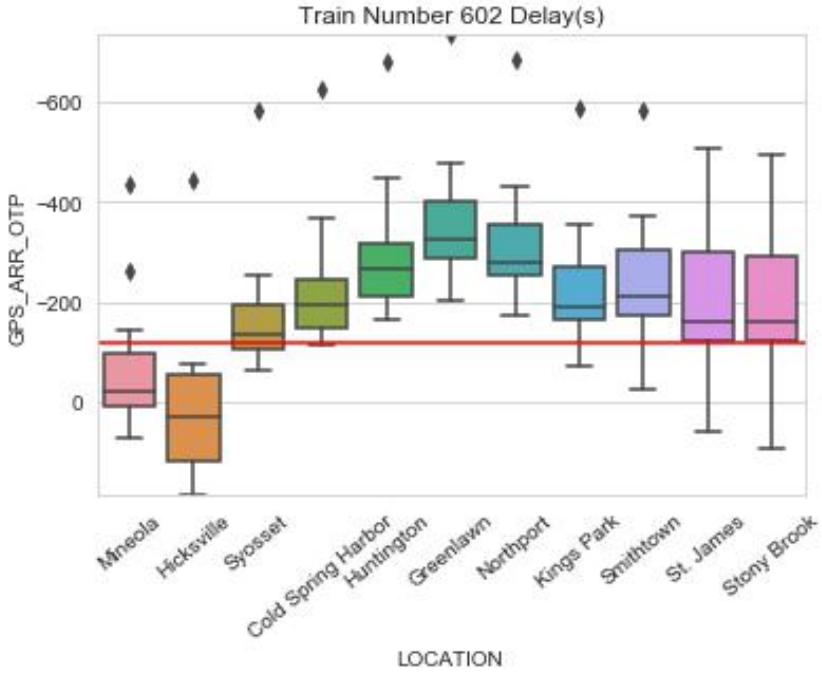

Fig.4. Series boxplot of train delays by station on the Port Jefferson line.

A decision tree (Figure 5) generates the lowest error against the benchmark. RFR, and decision trees in general, learn by using the features to narrow down the range of possible predictions to as little as possible through a series of boolean questions [13] . Nodes represent a decision point. For example, WEEKDAY < 5, means the record is during the week. If this is true, then the decision proceeds down the true leaf of the tree.

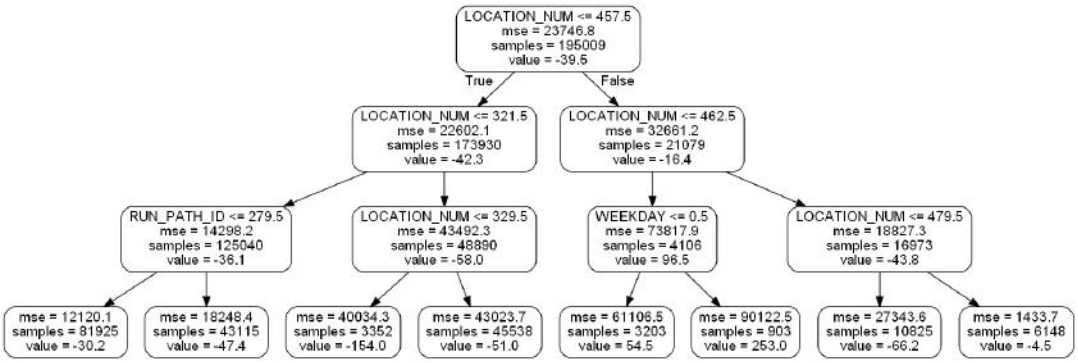

Fig.5. Subsect of random forest regression decision tree.

Linear regression draws a straight line through the scattered values on a plot, calculates an R^2 value less than one, which represents the spread of the data points. An R^2 of one indicates a perfectly straight line. The model then uses the value for x to predict Y based on the position of the line outwards to infinity.

3.2 Data Preprocessing

Figure 5 dropped several columns that weren't useful for the analysis including tail engine number, lead engine number, car count, version, canceled. Many of these rare cases like "CANCELED" are made irrelevant due to the size of the dataset.

```
df_trnevents.drop(['ACT_TAIL_ENGINE','CANCELLED_AT_LOCA-
TION','SCHED_CAR_COUNT','ACT_CAR_COUNT','ACT_LEAD_ENGINE.1','ACT_TAIL_EN-
GINE.1','CREW_NUM'], axis=1, inplace=True)
df_trnevents.drop(['TRAIN_NUM','VERSION','INSERTED','CANCELLED'], axis=1, inplace=True)
df_trnevents['PASS_BRANCH_NAME'].fillna('Unknown',inplace = True)
df_trnevents.drop(['ARR_TRACK',
'SCH_TRACK','GPS_DEP_SOURCE_TYPE','DEP_TRACK','GPS_DEP_OTP','GPS_DEP_DTM'], axis=1, inplace =
True)
df_trnevents['GPS_ARR_OTP'].fillna( '0' ,inplace=True)
df_trnevents['GPS_ARR_OTP'] = pd.to_numeric(df_trnevents['GPS_ARR_OTP'], errors='coerce')
df_trnevents.drop('GPS_ARR_SOURCE_TYPE', axis=1, inplace = True)
df_trnevents['GPS_ARR_DTM'].fillna(df_trnevents['RUN_DATE'])
#some data cleaning
```

Fig.6. Drops of null values.

One thing to consider for future iterations of the model might be that the RFR generated an overfit. Perhaps additional batching in "day of the month" where weekdays are batched into one ground, and weekends are batched into another, will reduce the overfit.

Further, arrival on time performance had some null values. Since most records are zero, these cells were filled with 0.

3.3 Classic Regression Implementation

The sklearn train_test_split function was used to separate the data into training and testing sets. 80% of the available data was used for training and 20% of the data was used for testing. There is a time series element to this data that was considered but was not well understood. Therefore, this model was implemented as a classic regression implementation [14]. Eventually, a web app will need to be developed to allow a customer to get the information that they need as it relates to how late their train is likely to be.

The original hyperparameters used were n_estimators = 10 and max_depth = 3. This was to see if the model would work quickly, with the understanding that tuning would be required in order to improve prediction accuracy.

A technical complication I ran into was an apparent overfit, future iterations of this model will attempt to address this, but it may prove an endless enterprise [15].

IV. DISCUSSION

Hyperparameter tuning [16] was used to determine the best parameters for the model. However, due to memory limitations the execution time for this job was long. The model

would be better served in terms of error reduction by loading in additional data, which is currently available to 9/2019. However, for the purpose of this case, I will only be using a single month. The hyper-parameters generated from the random grid are found in Figure 7.

```
{'n_estimators': 800,
'min_samples_split': 10,
'min_samples_leaf': 4,
'max_features': 'sqrt',
'max_depth': 50,
'bootstrap': True}
```

Fig.7. Hyper-parameters generated from the random grid search.

These parameters (Figure 7) resulted in an RMSE of 149.519479103 seconds while using the RFR model.

The final model is a Random Forest Regression decision tree from sklearn. The model produced a RMSE of 153.34323300184303 seconds before hyper-parameter tuning. Therefore it appears that tuning increased the accuracy of the model by approximately four seconds.

The features used had different levels of importance as it relates to train arrival time. Their scores can be found in descending order in Figure 8. This should be used to prioritize action items. In this research, location was the variable most related to delayed arrival time.

```
Variable: LOCATION_NUM      Importance: 0.54
Variable: RUN_PATH_ID       Importance: 0.39
Variable: PEAK_NUM          Importance: 0.04
Variable: WEEKDAY           Importance: 0.04
Variable: MONTH             Importance: 0.0
```

Fig.8. Variables in relation to importance.

Month was unimportant because a single month was used to train the model, but future iterations which include all available data will consider month. I would like to improve the prediction to an RMSE of 120 seconds. Additional data, and dealing with overfit, will be fundamental to achieving this goal. I identified the overfit by plotting the predicted data for the testing set, and then ran the model on the testing set and plotted one over the other as seen in Figure 9.

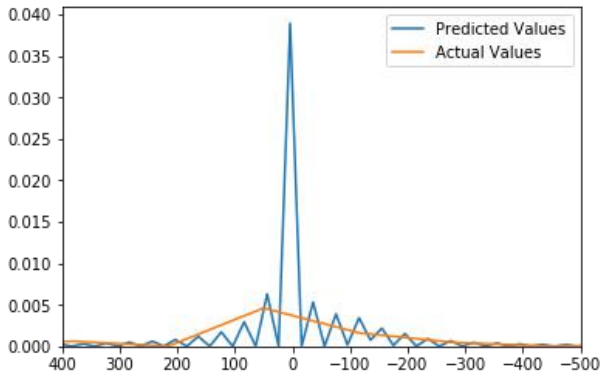

Fig.9. Plot of predicted versus actual train delays.

Perhaps the overfit can be addressed by batching the day of the week into two categories. The categories can be weekday or weekend because the schedules are typically different during those periods.

The benchmark XGBoost Linear Regression model performed at an RMSE of 168.8 seconds. By comparison, the hyper-parameter tuned RFR model performed at a RMSE of 149.5 seconds. Therefore, the prediction was improved by nearly 20 seconds compared to the baseline, which is significant. This model provides a reasonable prediction of train delays heading to Penn Station. It is only 30 seconds higher than the acceptable error for an actual train's arrival time compared to scheduled arrival time.

Interestingly, there is some precedent to claim that trains can accumulate delay over time as they proceed along their run path. Such a system may lend itself to more complex models, like neural networks [17], where the state of the previous node affects the state of the current node.

Figure 10 illustrates the delays of a single train over the month as it proceeds along its normal run path in sequence. It's clear that a linear model could be applied to this specific case. It's also clear that outliers exist in this data. These can be due to equipment failures or police actions that cause trains to sit for extended periods of time. However, this data is very large, and accounts for outliers.

V. CONCLUSION

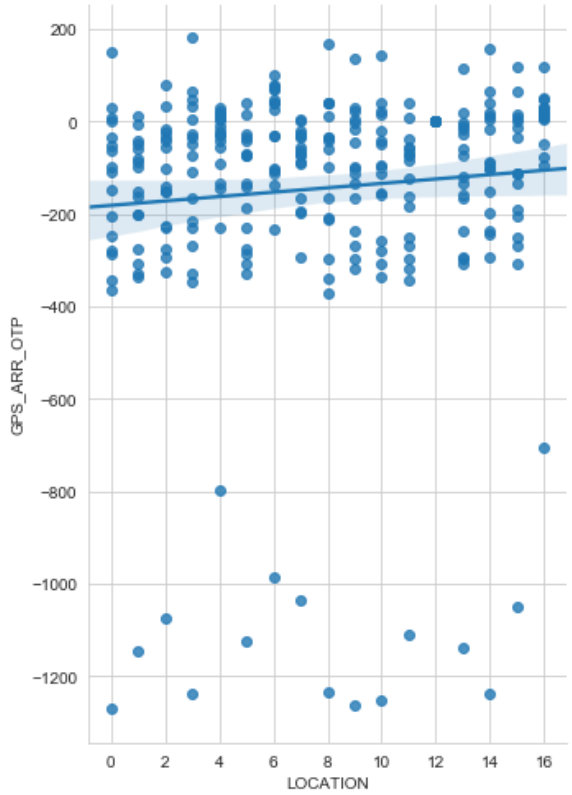

Fig.10. Delays of a single train over the month as it proceeds along its normal run path in sequence.

Eventually, the data can be put into production, either at mylirr.org, or as a standalone webapp for customers to have access to predictions about when they will arrive at their destination.

REFERENCES

[1] Hinsdale EB. History of the Long Island Railroad Company, 1834-1898. Evening Post Job Printing House; 1898.

[2] Segal MR. Machine learning benchmarks and random forest regression.

[3] Kankanamge KD, Witharanage YR, Withanage CS, Hansini M, Lakmal D, Thayasivam U. Taxi trip travel time prediction with isolated XGBoost regression. In2019 Moratuwa Engineering Research Conference (MERCon) 2019 Jul 3 (pp. 54-59). IEEE.

[4] Wang R, Work DB. Data driven approaches for passenger train delay estimation. In2015 IEEE 18th International Conference on Intelligent Transportation Systems 2015 Sep 15 (pp. 535-540). IEEE.

[5] Koshiishi I. Maintenance issues for railway facilities and future prospects in maintenance. JR East Technical Review. 2014;29.

[6] Sajan GV, Kumar P. Forecasting and Analysis of Train Delays and Impact of Weather Data using Machine Learning. In2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT) 2021 Jul 6 (pp. 1-8). IEEE.

[7] Nilsson R, Henning K. Predictions of train delays using machine learning.

[8] Henderson G, Darapanemi V. Managerial uses of causal models of subway on-time performance. Transportation Research Record. 1994(1451).

[9] Chai T, Draxler RR. Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. Geoscientific model development. 2014 Jun 30;7(3):1247-50.

[10] Zaniolo C. Database relations with null values. InProceedings of the 1st ACM SIGACT-SIGMOD symposium on Principles of database systems 1982 Mar 29 (pp. 27-33).

[11] Abdulla PA, Atig MF, Chen YF, Diep BP, Dolby J, Janků P, Lin HH, Holík L, Wu WC. Efficient handling of string-number conversion. InProceedings of the 41st ACM SIGPLAN Conference on Programming Language Design and Implementation 2020 Jun 11 (pp. 943-957).

[12] Coulston JW, Blinn CE, Thomas VA, Wynne RH. Approximating prediction uncertainty for random forest regression models. Photogrammetric Engineering & Remote Sensing. 2016 Mar 1;82(3):189-97.

[13] De Ville B. Decision trees. Wiley Interdisciplinary Reviews: Computational Statistics. 2013 Nov;5(6):448-55.

[14] Lehmann A, Overton JM, Leathwick JR. GRASP: generalized regression analysis and spat ial prediction. Ecological modelling. 2002 Nov 30;157(2-3):189-207.

[15] Chicco D. Ten quick tips for machine learning in computational biology. BioData mining. 2017 Dec;10(1):1-7.

[16] Bardenet R, Brendel M, Kégl B, Sebag M. Collaborative hyperparameter tuning. InInternational conference on machine learning 2013 May 13 (pp. 199-207). PMLR.

[17] Jin W, Li ZJ, Wei LS, Zhen H. The improvements of BP neural network learning algorithm. InWCC 2000-ICSP 2000. 2000 5th international conference on signal processing proceedings. 16th world computer congress 2000 2000 Aug 21 (Vol. 3, pp. 1647-1649). IEEE.