



# Ridership prediction and anomaly detection in transportation hubs: an application to New York City

Mingyi He<sup>1,a</sup>, Urwa Muaz<sup>1,b</sup>, Hong Jiang<sup>1,c</sup>, Zengxiang Lei<sup>2,d</sup>, Xiaowei Chen<sup>2,e</sup>, Satish V. Ukkusuri<sup>2,f</sup>, and Stanislav Sobolevsky<sup>1,g</sup>

<sup>1</sup> Center for Urban Science and Progress, New York University, New York, NY 11201, USA

<sup>2</sup> Lyles School of Civil Engineering, Purdue University, West Lafayette, IN 47907, USA

Received 15 October 2021 / Accepted 29 March 2022 / Published online 15 April 2022

© The Author(s), under exclusive licence to EDP Sciences, Springer-Verlag GmbH Germany, part of Springer Nature 2022

**Abstract** Ridership modeling is a growing field critical for Intelligent Transportation. Accurate traffic prediction and early surge detection are vital components in designing public transit dispatching systems. However, modeling Spatio-temporal traffic at a small geographic scale and fine time granularity is challenging due to the sparseness, low signal-to-noise ratio, and the large dimensionality of the mobility network data. We propose a framework for edge-level traffic prediction to tackle these challenges, which addresses the curse of dimensionality through a pipeline of appropriate network aggregation, nonlinear modeling, and final edge-level disaggregation. Subsequently, we show that the low-dimensional aggregated space model residuals are more suited for anomaly detection than raw ridership data. Our framework is evaluated using the for-hire vehicle and taxi ridership dataset from the two airports in New York City, experimenting with different network aggregation techniques and modeling paradigms. The results reinstate the superiority of the proposed pipeline in ridership prediction and anomaly detection compared with single-model methods, and help build up scenario design for transportation simulation and planning.

## 1 Introduction

Intelligent Transportation System is widely used in public transit to solve problems in urban transportation systems like pollution, congestion, and inefficiency caused by increasing mobility [1–3]. To design a public transit system and scheduling, ensuring the system's resilience requires an understanding of the projected demand under regular and irregular conditions. Hence, traffic demand prediction and anomaly detection become vital processes.

We design an accurate and efficient system to make a short-term prediction and detect anomalous days in urban networks in this study by leveraging the open-source data available for New York City, one of the major urban centers in the world. Taxi ridership data form a respectable proportion of urban mobility in New York and thus can be used as a reliably proxy for urban mobility. A precise taxi pick-up and drop-off volume prediction system supports decision-making in dispatching cabs and for-hire vehicles to improve

the taxi service [4, 5]. The widely accepted conception of anomalies comes from Hawkins [6]: "An anomaly is an observation which deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism.". While regular on a recurring basis, transportation flows are also subject to anomalous flows due to various events. National holidays, extreme weather, disasters, game days, and religious events cause significant surges [7] in traffic, leading to issues with traffic management and control. Surges and anomalies are crucial stress points for transport resource scheduling systems, often requiring pre-emptive actions from the authorities. This paper focuses on analyzing taxi ridership system behavior and exception in two airports in New York City: John F. Kennedy International Airport (JFK) and La Guardia Airport (LGA). The idea is to leverage relevant urban data, model the ridership prediction, and then find and analyze various anomalies.

Transport data have complex spatial dependencies and nonlinear temporal dynamics. Since transport data comprise connected origins and destinations and have a highly interdependent topological structure, modeling individual nodes cannot capture the interactions. Furthermore, mobility is highly correlated with external factors. In transportation hubs, pick-up volume depends heavily on arrival flows, and it varies with the weather, time, weekdays, and other elements. We build

<sup>a</sup> e-mail: [mh5172@nyu.edu](mailto:mh5172@nyu.edu)

<sup>b</sup> e-mail: [um367@nyu.edu](mailto:um367@nyu.edu)

<sup>c</sup> e-mail: [hj1274@nyu.edu](mailto:hj1274@nyu.edu)

<sup>d</sup> e-mail: [lei67@purdue.edu](mailto:lei67@purdue.edu)

<sup>e</sup> e-mail: [chen3379@purdue.edu](mailto:chen3379@purdue.edu)

<sup>f</sup> e-mail: [sukkusur@purdue.edu](mailto:sukkusur@purdue.edu)

<sup>g</sup> e-mail: [ss9872@nyu.edu](mailto:ss9872@nyu.edu) (corresponding author)

a dataset to capture critical external features. Then, we experiment with different regression models capable of modeling the influence of these external variables.

Our experimentation shows that nonlinear modeling techniques are more suited to mobility data. We evaluate two complex nonlinear models for this task, Random Forest and long short-term memory networks (LSTM). We use lagged variables with Random Forest to model temporal relationships, and LSTM is autoregressive in nature. We design a prediction model predicts ridership flows which considering the spatial relationships between nodes.

Subsequently, we apply residuals from prediction models to surge detection. The most common approach for anomaly detection is building a probabilistic model of the normal behavior and categorizing anomalies as events that have low likelihood according to the probabilistic model [7, 8]. These models have a shortcoming in that they do not consider the externalities and temporal dependencies of the variable in the study. Unsupervised feature learning approaches like principal component analysis (PCA) and autoencoders which are commonly used before anomaly detection also have the same flaw: they are not capable of modeling externalities like weather and day of the week. We formulate a framework where an initial modeling phase corrects externalities and temporal dependencies in the ridership data. Then, residuals from this stage are used for anomaly detection. Consequently, an anomaly is only detected if the anomalous behavior cannot be explained using the available external data and ridership history. Since we use the taxi zone data and perform zone-level predictions, the high-dimensional residual still poses a problem for anomaly detection. Hence, we experiment with multiple network aggregation techniques like PCA and network community detection to reduce the dimensions for anomaly detection.

Our main findings are summarized as follows:

1. we show that unsupervised learning-based anomaly detection based on regression residuals outperforms anomaly detection from ground truth ridership values;
2. we show that mobility networks have strong nonlinear spatial and temporal dependencies, and thus, nonlinear models are better than legacy linear models for modeling purposes;
3. we provide empirical evidence that dimensionality reduction is necessary before modeling high-dimensional mobility network data;
4. our results show that community detection is a better network aggregation technique when compared to geographical and administrative aggregations in anomaly detection;
5. our results show that autoregressive LSTM outperforms Random Forest, which uses lagged variables for temporal modeling in the out-of-sample test.

## 2 Related work

### 2.1 Taxi ridership volume prediction

Traffic demand prediction is fundamental to building a stable smart-city transportation system [9, 10]. We use taxicab and for-hired vehicle (FHV, e.g., Uber) as they are major components [11, 12] in this network. Taxi demand prediction methodologies lay strong emphasis on the complex spatial [13, 14] and temporal [15, 16] relationships in the dataset and the need for models that can exploit both. One challenge for mobility modeling is that traffic flow is heterogeneous of predictability across geographical regions [17].

Traditional approaches for traffic forecasting are mostly based on techniques like AutoRegressive Integrated Moving Average (ARIMA) model and Kalman filtering [18, 19]. Similarly, researchers model urban mobility using taxi data leveraging ARIMA models [20]. Autoregressive integrated moving average (ARIMA) [18] is the most widely used forecasting model and has been extensively applied to traffic forecasting problems [21, 22]. SARIMA is a variant of ARIMA that also captures seasonal periodicity has also been used for traffic forecasting [23]. These models produce predictions for individual nodes and fail to capture the relationships between nodes, which is a vital aspect of real-world traffic networks. Vector ARMA (VARMA) [24] and space-time ARIMA (STARIMA) [25] are two more sophisticated variants that perform traffic forecasting for multiple nodes in a network. These were initial steps towards edge-level traffic forecasting which considers spatial and temporal relationships. Ding et al. [26] proposed an STARIMA model to predict traffic volume in 5 min on street links, and spatial features include average trip duration on links. Ravi et al. [27] applied a vector autoregressive model to predict traffic volumes on freeways including upstream and downstream. These techniques have three shortcomings:

1. They only work on series that are stationary or stationary after differencing, while the non-stationarity problem in urban transportation is complex and might not necessarily be addressable with differencing.
2. They do not consider spatial and structural dependencies that traffic networks exhibit and forecast each sensor as an individual time series.
3. They suffer from the curse of dimensionality.

Most state-of-the-art models are deep learning-based include deep multi-task multi-graph learning approach [28], Spatio-Temporal Encoder–Decoder Residual Multi-Graph Convolutional network [29], and Spatio-temporal graph convolutional network [30]. Models that incorporate both spatial and temporal features into considera-

tion have been shown to yield significantly improved performance [31]. Zone-based taxi prediction [31–33] is less challenge than OD-based prediction [29, 34, 35]. Furthermore, mobility is highly correlated with external factors [36]. In transportation hubs, pick-up volume depends heavily on arrivals amount, and it varies with the weather, time, weekdays, and other elements. We build a dataset for this set of important external features. Then, we experiment with different regression models capable of modeling external variables.

While, Graph Neural Network (GNN) [37] is a popular and successful method in traffic prediction [38, 39] as it captures connections between nodes. However, this feature brings no benefits to a star-structure network as it merely propagates information between adjacent nodes and neglects information between unconnected nodes [40]. Hence, it is not an appropriate method in our project as we are working on a star network: all nodes in our graph are only connected to one node (JFK or LGA).

A linear regression model with a large number of engineered features [41] can also outperform the complex time series model in out-of-sample datasets. We use linear regression models as our baseline of aggregated ridership prediction. Subsequently, we experiment with a more robust nonlinear model, Random Forest. We choose Random Forest regression as they have built-in mechanisms to avoid overfitting. Bootstrapping and random subsetting [42] of the feature space are used for building weak classifiers, which are then ensembled for the final prediction. This leads to good generalization and avoids overfitting. Furthermore, using the Random Forest for edge-level prediction allows modeling the spatial and structural dependencies in the nodes. Recurrent neural networks (RNN) are able to model nonlinear temporal dynamics in sequence data and forecasting. However, traditional RNNs suffer from blowing and vanishing gradients problems and thus make it difficult for them to capture long-term temporal dependencies [43]. Besides, they rely on predetermined time lags. Long-Short-Term Memory (LSTM) was proposed to address these challenges [43]. Unlike traditional RNNs, they can model these long-term dependencies using memory units and gated structures. Furthermore, they can automatically determine optimal time lags for the dataset by learning when to open and close relevant gates. Traffic data are known to have long-term temporal dependencies, making LSTM a suitable modeling technique.

Even previous models have impressive performance in ridership prediction, our objective is to train a model which not only returns reliable prediction but also provides insights on anomaly detection.

## 2.2 Anomaly detection in networks

The application of network anomaly detection is a broad research area that is applicable in different domains, including fraudulent activities in transactions [44], social networks data [45, 46], and transportation

networks [7, 47, 48]. One challenge in this field is the lack of widely accepted labeled datasets for anomaly detection [44] which can be used to benchmark different approaches. Thus, in this work, we create our own labeled dataset.

If the input data are high-dimensional, a dimension reduction step is usually applied before anomaly detection. Lakhina et al. [49] applied Principal Component Analysis (PCA) to network anomaly detection for the first time using Principal Component reconstruction residuals as low-dimensional features for anomaly detection. Also, reconstruction error from Autoencoder [50] is also a common method to detect outliers. Lately, Ringberg et al. [51] and Brauckhoff et al. [52] suggested that PCA is not suitable for anomaly detection because of parameter sensitivity and lack of temporal correlation on a transformed dataset. An alternative approach to reduce the graph data dimension is using a graph aggregation technique like community detection. A Community is defined as a group of nodes in a network which interact excessively frequently, and it is a common substructure in networks [53]. Zhenzhang et al. [54] applied community detection in complex evolutionary networks for the first time. GNN and its variants have an advantage in network analysis as they capture relationships between nodes. However, they have no superiority, while there is no connection between neighbors [40].

At the anomaly detection stage, clustering is the most common machine learning technique. Prior work has shown that soft clustering methods are more flexible and feasible as they consider the nature of the data [55]. Gaussian Mixture Model (GMM) has been introduced to network anomaly detection in 2006 by Tran et al. [56], and it is shown to be a better choice than K-means when clusters have different sizes and correlations.

Supervised learning methods are applied to anomaly detection in Spatio-temporal datasets as well [47, 57]. Kong et al. applied a one-class support vector machine to detect urban anomalies based on prediction residuals from LSTM. This paper proves that residuals from Random Forest outperform LSTM in anomaly detection.

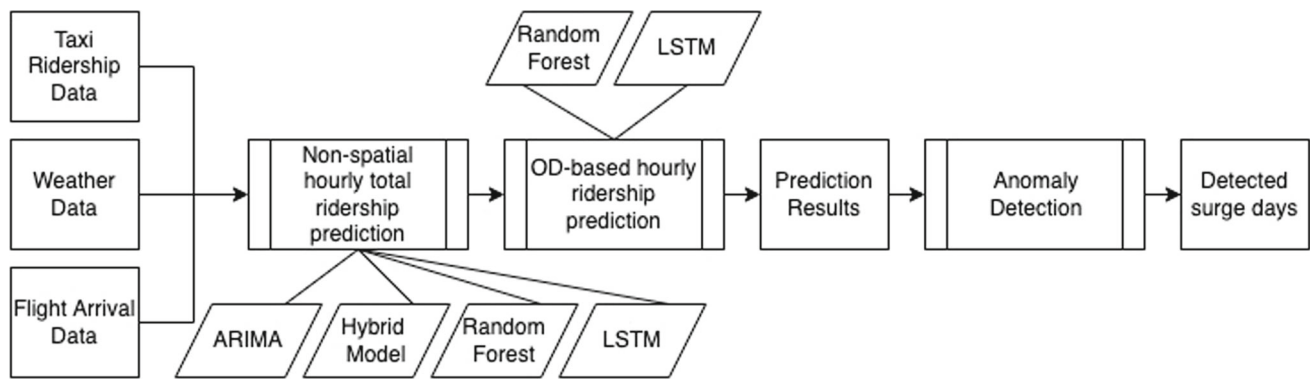
Based on the literature review, we identify four main challenges in temporal network anomaly detection:

1. A lack of a generally applicable method that works on different mobility networks.
2. Low signal-to-noise ratio makes separating normal and abnormal data points harder.
3. A lack of widely accepted labeled datasets for anomaly detection.

## 3 Data and preprocessing

### 3.1 Outgoing traffic volumes from airports

The vehicle's outgoing datasets are collected from two major transportation hubs in New York City, includ-



**Fig. 1** Experiment pipeline

ing JFK and LGA. New York City Taxi and Limousine Commission provides yellow taxi, green taxi, and for-hire vehicles' trip records [58] include fields capturing pick-up and drop-off dates/times, pick-up/drop-off locations. The pick-up/drop-off locations are aggregated into 263 taxi zones, where JFK and LGA act as two taxi zone units. Here, we only take yellow taxis and for-hire vehicles into account, as green taxis are not allowed to pick up passengers in Manhattan. We aggregate the traffic volume by date. The summary of aggregated datasets is in Table 5.

### 3.2 Arrival flights and trains volumes

We assume that the arrival passengers' amount at each transportation hub affects the outgoing traffic volume in a specific time lag. Thus, we collected flight arrival data to predict the traffic volume. Flight datasets came from the Bureau of Transportation Statistics in JFK and LGA in 2018.

Bureau of Transportation Statistics only records domestic flights. According to Port Authority traffic report, about 39% JFK flights (38,260 in 97,853) and 7% LGA flights (2,301 in 29,911) are international. And international flights usually have higher passenger capacity. Therefore, lacking international flights might limit the capacity of our models to explain associated ridership fluctuations.

### 3.3 External datasets: weather data and flight arrival data

Taxi ridership volume from transportation hubs is determined by the combination of incoming passengers' population, the possibility of choosing a taxi, and the available taxi amount. The demand for taxis at transportation hubs and in the city influence each other, according to [59]. During inclement weather, the volume of trips within Manhattan increases. As frequent and short trips bring more income to drivers, airports become less attractive to taxi drivers [60]. Our weather dataset is collected from Aviation System Performance Metrics data, supplied by National Oceanic And Atmospheric Administration.

Besides weather conditions, the temporal variations, such as day-of-week, and time-of-day, significantly impact on the taxi supply and demand market. Dr. Kamga found that on weekend nights, taxis have the highest average revenue, which is caused by higher pick-up rates and relatively shorter distance trips [61].

### 3.4 Urban anomaly events

We set two events datasets separately for each transportation hub to validate our anomaly detection results and events types, including national holidays, extreme weather (snow and storms) days, and major flight delays and cancellations for airports. Details are in Appendix A Table 7.

Extreme weather includes two types: winter storm and thunderstorm wind. Snow will not melt away immediately, the impact will last longer, so the days after winter storms are regarded as winter storm days, as well. Airport events are collected from news from 01/01/2018 to 12/31/2018, and most of them are related to extreme weather days in other cities and states. For example, on Dec 21, a winter storm hit the east coast, and even this day is not recorded as a storm event in New York City, it still led to one-third of flights being delayed in LGA, and more than 15% flights delayed in JFK [62].

## 4 Methodology

Our study has two distinct objectives, developing an accurate traffic prediction model and analyzing the usability of this model for traffic surge detection. Both objectives influence our choice of methodology. The whole process is visualized in Fig. 1. Input datasets include taxi ridership data, weather data, and flight arrival data. Initially, we experiment with an array of hypothesis/modeling paradigms on non-spatial one-dimension timeseries ridership prediction. Then, select the most suitable model for edge-level traffic prediction. Subsequently, we choose the best-performing model and study its utility for surge detection.



We formulate a framework where an initial modeling phase corrects externalities and temporal dependencies in the ridership data. And residuals from this stage are used for anomaly detection. Consequently, an anomaly is only detected if the anomalous behavior cannot be explained using the available external data and ridership history. Throughout this paper, we refer edge-level data as taxi-zone wise, no aggregation data.

Mobility network data are usually very high-dimensional, and linear machine learning methods do not perform well with high-dimensional data, especially when the dataset size is not many times larger than the number of dimensions. This is a generic issue with machine learning models and is referred to as the curse of dimensionality. The curse of dimensionality can be detrimental to both the ridership prediction and subsequent anomaly detection. Furthermore, edge-level data have a low signal-to-noise ratio, and some level of aggregation is required to address this issue.

To summarize our pipeline, we use a dimension reduction technique to transform the edge-level network data in a reduced feature space. Then, we perform ridership prediction in this aggregated space and use disaggregation techniques to attain edge-level prediction. Anomaly detection uses the low-dimensional residuals of the aggregated space surge isolation.

#### 4.1 Dimension reduction and network aggregation

Once an edge-level prediction model is developed, it is still a challenge to use this high-dimensional residual data for anomaly detection. Most prior works address this issue by adopting a two-stage approach [49, 51, 52], where low-dimensional representation is learned before applying anomaly detection techniques on the latent representation. We address this issue by experimenting with different spatial and structural aggregations and dimension reduction techniques. Then, we perform modeling and residual generation in these aggregated spaces to get low-dimensional residuals for anomaly detection. Since comparing the performance of ridership prediction with modeling at different aggregated spaces is hard, we include a disaggregation step that transforms the aggregated space back to edge level using inverse PCA or multiplying the aggregated demand by edge-level weights. The edge-level weights are the yearly edge-level demand divided by yearly aggregated-level demand.

Different aggregations and dimension reduction techniques that we experimented with are discussed below. At the dimension reduction stage, we aggregated the dataset in three different ways:

1. spatial aggregation that aggregates 263 taxi zones to 5 boroughs;
2. topological aggregation: uses a community detection algorithm to aggregate taxi zones to 6 or 24 communities;
3. linear dimension reduction that applies the principal component analysis method to reduce the raw dataset to 6 or 24 dimensions.

The following subsections introduce more details of dimension reduction and network aggregation techniques.

**Principal component analysis:** Edge-level prediction results in a pipeline including dimension reduction techniques are from an inverse transformation of predicted components from dimension reduction. We choose PCA, because it minimizes the reconstruction error, and the transformation is invertible. Also, the main message of our pipeline is to show the utility of each phase, and PCA is the simplest suitable method to demonstrate the utility of the dimension reduction, while other more advanced methods like kernel PCA or autoencoders could be considered. This model learns a linear transformation that projects the data into another space, where the variance of the data defines vectors of projections. By restricting the dimension to a certain number of components that account for most of the variance of the data set, we can achieve dimension reduction. We perform PCA to get a low-dimensional space which we use for ridership modeling. Inverse PCA is used to get edge-level prediction performance. We try with different numbers of principal components and choose the best-performing one for each hub.

**Spatial aggregation:** New York City is divided into administrative divisions termed as boroughs. There are five boroughs in New York named Bronx, Brooklyn, Manhattan, Queens, and Staten Island. The spatial breakdown of the city into boroughs is available in the Appendix Fig 5.

**Community detection:** provides a spatial topological aggregation for taxi zones, while borough-level aggregation ignores the interaction between taxi zones.

A community is a collection of nodes in a network that interact significantly. In community detection, the algorithm compares the actual edge weight with the average expected value for each edge in the original network. The edges with positive relative strength scores represent particularly strong network connections and are placed inside the community, while edges with negative scores are placed between the communities. This process maximizes the modularity score and determines the optimal partitioning. Therefore, community detection provides a way of aggregation that preserves topological and structural information as opposed to crude spatial aggregation. Our community detection results come from [7] which collected yellow cabs, green cabs, and for-hire vehicles trip within the New York City from Jul 01, 2017, to Dec 31, 2018. 263 taxi zones are aggregated to 6 or 24 communities. The visualization of community detection results is available in Appendix B Figs. 6 and 7. We try different granularities of community detection as our aggregation step before ridership prediction and use the best-performing granularity for each hub. For these experiments, we use weighted disaggregation, which is described below.

**Weighted disaggregation:** This technique is used to convert the aggregated ridership data in community and borough space back to network space. We learn

a linear transform from the historical data that shows the average fractional contribution of each edge towards the community or borough that it resides in. For each unit  $u$  of aggregated space uni (which can be a single community or a borough)

$$R_u = \sum_{e \in u} B_e * R_e, \quad (1)$$

where  $R_u$  is the ridership of that unit  $u$  in aggregated space,  $R_e$  is the rider of edge  $e$ , and  $B_e$  is the fractional contribution that is learned from historical data.

## 4.2 Ridership prediction

We frame the ridership prediction problem as an hourly edge-level prediction for the mobility network based on external variables and ridership history.

As an initial experiment, we compare different models for network-wide aggregated 1-dimension ridership timeseries at the hourly level. In this stage, we compare a linear regression, a hybrid model (combined with linear regression and autoregressive integrated moving average), Random Forest regressor, and LSTM. The success of Random Forest model and LSTM emphasizes the ridership's long-term temporal dependencies.

Subsequently, we frame the problem as hourly ridership prediction at edge level. This modeling is performed on both aggregated and original networks. For this task, we only evaluate two highly nonlinear models, Random Forest, and LSTM. Both models perform network-wide prediction enabling them to model relationships between network nodes. Random Forest uses lagged variables, while LSTM uses autoregressive modeling to model temporal relationships. Furthermore, we explore combinations of different models, aggregation, and disaggregation methods. Models evaluated in this experiment include the following.

### 4.2.1 Linear regression

We formulated the linear regression model for total outgoing traffic prediction from each hub. We use lagged variables to model temporal dependencies. The model formulation is shown below

$$\begin{aligned} \text{RIDERSHIP}_t &= \beta_0 + \beta_1 * \text{ARRIVAL}_{t,m,n} \\ &+ \beta_2 * \text{TimeofDay}_t \\ &+ \beta_3 * \text{DayofWeek}_t, \end{aligned} \quad (2)$$

where variables' explanation is in Appendix A Table 8.

### 4.2.2 Hybrid model

A hybrid model is the combination of the ordinary least square linear regression and autoregressive integrated moving average. Since the normal linear regression model does not incorporate temporal relationships in

ridership prediction, we combined it with AutoRegressive Integrated Moving Average (ARIMA). Based on the result, we combined linear regression with ARIMA and introduced a new variable  $\text{arPred}_t$

$$\text{arPred}_t = \widehat{\text{RIDERSHIP}}'_t - \widehat{\text{RIDERSHIP}}'_{t-1} \quad (3)$$

$$\text{RIDERSHIP}'_t = \text{RIDERSHIP}_t - \text{RIDERSHIP}_{t-1} \quad (4)$$

$$\begin{aligned} \widehat{\text{RIDERSHIP}}'_t &= \mu + \phi_1 * \text{RIDERSHIP}'_t \\ &+ \dots + \phi_8 * \text{RIDERSHIP}'_{t-8} \\ &+ \theta_1 * \epsilon_{t-1} + \dots + \theta_q * \epsilon_{t-4} \end{aligned} \quad (5)$$

where  $\phi$  are the parameters of the autoregressive part,  $\theta$  are the parameters of the moving average, and  $\epsilon$  are error terms.  $\epsilon$  are independent, identically and normally distribution with zero mean.

ARIMA model is applied here to get time series prediction  $\text{arPred}_t$  from the linear combination of lagged  $\text{RIDERSHIP}$  values and the error terms whose values occurred at various times in the past. Then, we add  $\text{arPred}_t$  into the previous OLS model as a new variable. The resulting regression equation is shown below

$$\begin{aligned} \widehat{\text{RIDERSHIP}}'_t &= \beta_0 + \beta_1 * \text{ARRIVAL} \\ &+ \beta_2 * \text{arPred}_t + \beta_3 * \text{TimeofDay}_t \\ &+ \beta_4 * \text{DayofWeek}_t. \end{aligned} \quad (6)$$

### 4.2.3 Random Forest

Ensemble learning has been successfully applied to improve the performance of regression and classification tasks. Two branches of ensemble learning in trees are boosting [63] and bagging [42]. In boosting, successive trees give extra weight to misclassified examples by earlier trees. In bagging, all trees are constructed separately on bootstrapped samples of data. Both use majority voting for ensemble prediction. Random Forest [64] is a further modification of bagged trees, where a subset of available features is picked randomly to pick the best split for each node in the tree. Random Forests are less prone to overfitting which is a big problem for modeling from high-dimensional data. This is because Random Forest is an ensemble of weak predictors and incorporates feature pruning within the algorithm. Furthermore, they are capable of modeling nonlinear spatial and temporal dynamics.

We use Random Forests for nonlinear modeling for both edge level and aggregated space as they are relatively robust to overfitting which is a general hazard for predictive modeling based on high-dimensional data [65]. Furthermore, they have built-in support for multivariate regression, which we use for direct edge-level prediction using a single model. We perform a random grid search and n-fold cross-validation for the search of optimal model parameters. Our experiments uncovered that the data have strong temporal correlations which

we exploited using time-lagged features. Grid search showed that the model performed the best when given ridership history of preceding 12 h as a part of lagged features.

#### 4.2.4 LSTM

We designed an LSTM model for taxi zone-level ridership prediction. The model is composed of an aggregation layer, LSTM layers, and a disaggregation layer. The aggregation layer is matrix multiplication that transforms the data from edge level to an aggregated community level, and this layer is fixed and initialized to 24 community partition. LSTM layer consists of a stack of LSTM cells that consumes the sequential data in community space and outputs the prediction in low-dimensional latent space. The disaggregation layer is a linear transformation (fully connected layer without nonlinearity) that projects the prediction to taxi zone space. The whole model is trained end to end using adams optimizer. Bayesian optimization is used to tune design parameters like sequence context length and number of LSTM layers and model hyperparameters like learning rate.

### 4.3 Anomaly detection

Residual analysis is a method to study the residual between real data and estimated data in regression problems [66]. Generally, large errors are more likely to be anomalies. We applied log transformation to shape errors as normally distributed and eliminate the impact from the range of real values [67]. We experiment with residuals with and without log transformation and report the best-performing version

$$\text{error}' = \log(\text{Ridership} - \widehat{\text{Ridership}} + \min(\text{Ridership} - \widehat{\text{Ridership}})). \quad (7)$$

Anomaly detection is a special case of imbalanced classification, to avoid overfitting, we applied an unsupervised learning clustering algorithm to cluster patterns in ridership. A Gaussian mixture model is a probability-based clustering model that assumes all data points are from a combination of multiple Gaussian distributions with unknown parameters [68]. Anomalies are defined by the threshold of the rank of log-likelihood returned by the Gaussian Mixture Model. We use an iterative model-fitting approach similar to expectation maximization: the first step is to fit a GMM model on the entire residual dataset and get the likelihood of each data point. Then, drop data points below a likelihood threshold, and refit the model on remaining data points, applying the trained model to the whole residual dataset again. Finally select data points having likelihood below the previous threshold. This iteration is done until data points selected before and after converge to the same dataset. We have experimented with 100 threshold values, ranging from the bottom 1%

to 99% percentile of likelihood. The threshold of log-likelihood rank to divide days into anomalies and normal days is discussed in Sect. 5.3

For Gaussian Mixture Model Component Selection, we run models with 1–5 components on each residual dataset and use Bayesian Information Criterion (BIC) criteria for the selection of optimal components. According to the scikit-learn users' guide [69], model selection for Gaussian Mixture Model could be based on information-theoretic criteria and the Bayesian Information Criterion is a better choice. Then, we used the number of the component which yields the lowest Bayesian Information Criterion score for the further iterated Gaussian Mixture Model. The optimal number of GMM components varies with hubs and prediction methods, and the final choice of the hyper-parameter is available in Sect. 5.2.

### 4.4 Evaluation

Since we are predicting ridership at multiple dimensions, the performance evaluation metrics will be a series  $R^2$  values, one  $R^2$  value for each dimension. To generate a general  $R^2$  which describes the overall predicting model performance in the whole city, averaged  $R^2$  is applied. Instead of a uniformly weighted average, variance-weighted  $R^2$  is used to evaluate the prediction of all the traffic prediction models. Since it is not possible to fairly compare the models at different aggregation levels directly, we use disaggregation to get taxi zone-level predictions for all models, enabling one-to-one out-of-sample comparison. To evaluate the anomaly detection performances, we used the area under the precision-recall curve, as it is preferred over the area under the ROC curve (AUROC) when dealing with a highly imbalanced dataset [70], which is true in our case with anomalies being very sparse

$$\text{Precision} = \text{TruePositives} / (\text{TruePositives} + \text{FalsePositives}) \quad (8)$$

$$\text{Recall} = \text{TruePositives} / (\text{TruePositives} + \text{FalseNegatives}). \quad (9)$$

## 5 Results and discussion

Prediction models are evaluated at first, the model that has the best performance in total demand prediction will be applied to estimated edge-level demand and further anomaly detection combined with different aggregation methods.

### 5.1 Demand prediction

Random Forest and LSTM give the best results for total outgoing ridership prediction for each hub. Random Forest models capture nonlinear temporal dynamics and spatial dependencies in the dataset. Further-

**Table 1** Total outgoing variance weighted  $R^2$ 

Model	JFK	LGA
Linear regression	0.795	0.877
Hybrid model	0.793	0.875
Random forest regressor	0.892	0.939
LSTM	0.826	0.895

more, they are an ensemble of multiple regressors trained using bootstrapping, increasing their generalizability to unseen data. And LSTM is inherently designed for timeseries datasets. Hence, they were able to consistently outperform linear regression (Table 2).

Table 1 compares the performance of different pipelines for edge-level prediction. However, not all the approaches are compatible or reasonable to combine, for example, LSTM–PCA–inverse PCA and Random Forest–community–learnable disaggregation. PCA and inverse PCA cannot be combined with an LSTM, because a neural network could naturally perform dimensionality reduction and reconstruction. Moreover, we have an LSTM pipeline using community-based aggregated input, LSTM to predict, and disaggregates prediction results to edge level using a high-dimension output layer. However, this high-dimension output layer disaggregation method cannot be combined with RF. It is supposed to be connected with the LSTM prediction layers and tuning parameters in all stages sequentially.

Our proposed LSTM model outperforms RF in prediction for the same aggregation level. 24 community LSTM is better than 24 community RF and taxi zone-level LSTM is better than Taxi Zone-level RF. This shows that an autoregressive nonlinear model is better than a nonlinear model that uses lagged variables.

Modeling in aggregated space and disaggregating into edge-level produces better results than direct modeling in taxi zone space. This validates our hypothesis that dimension reduction is vital for modeling on high-dimensional transportation networks and shows that our proposed framework of dimensionality reduction, modeling, and dimension expansion is better than direct edge-level modeling. Additionally, we note that the reason other aggregation schemes do not work as well as PCA and learnable aggregation might be that the adopted disaggregation technique was too simple. Whereas learnable disaggregation and inverse PCA are better disaggregation steps than weighted disaggregation, resulting in better taxi zone level  $R^2$ .

Comparing our results with two related works turns out that our model outperforms previous methods. In *Short-Term Forecasting of Passenger Demand under On-Demand Ride Services: A Spatio-Temporal Deep Learning Approach* [32], Hangzhou (a major city in China) is divided  $7 \times 7$  grids, while the whole city's area is  $6505 \text{ mi}^2$ , so we compared their results to our 24-community aggregated prediction results as New York City is only  $307 \text{ mi}^2$ . Besides, as the total ridership amount in Hangzhou is not mentioned in this paper, we cannot compare RMSE or MAE with it, only  $R^2$

is applicable. The highest  $R^2$  in this project is 0.820, and we have different model performances in different transportation hubs, but all stations have a better performance: 0.865 in JFK, 0.929 in LGA. Another related work is *Taxi Demand Prediction Using Parallel Multi-Task Learning Model* [34], which predicts NYC taxi ridership at taxi zone level, but the discrepancy is they trained model on the first 10 months and forecast the upcoming 2 months, temporal granularity is 2-hour, while we only predict the next hour based on external data and ridership in the past 12 h. In this work, they did not provide  $R^2$ , but as we have the same research area, MAE and RMSE are applicable in this case. In their best model, the average MAE is 14.499, and RMSE is 22.441. However, our highest MAE is 1.2, and the highest RMSE is only 3.2. While it is hard to find a previous work with identical spatial and temporal granularity to conduct the model performance comparison, our models still show superiority in predicting performance.

## 5.2 Anomaly detection

Table 3 shows the normalized area under precision–recall at JFK and LGA. Random selection refers to randomly assigning each observation as an anomaly, and the model failed to capture any signal from the dataset. This serves as a reference point to compare the performance of our different methodologies. And scores from other methods are normalized by this value.

Residuals from Random Forest based on 24-communities aggregated datasets achieve the best performance in anomaly detection at both JFK and LGA. And based on BIC results, the number of components in GMM is 3 for anomaly detection in both JFK and LGA taxi ridership. This establishes the superiority of topological aggregation over crude spatial aggregation, and linear dimension reduction. This result makes sense as the geospatial locations and connections define interactions between taxi zones. For example, even LGA is located in Queens, community detection results show a stronger bond with and midtown Manhattan other than Queens. Besides, high-dimension residuals outperform low-dimension residuals. As fluctuations caused by anomalies could be diluted in highly aggregated datasets. Furthermore, residuals from prediction models with high  $R^2$  values do not promise better performance in anomaly detection. For example, at 24-communities' modeling space, LSTM beats Random-Forest in prediction but does not perform as well in anomaly detection. The potential reason is that LSTM captures more pre-anomaly signals and adapts quickly to temporal trend changes, thus being poor in anomaly detection. Community detection aggregated networks based on topology instead of an arbitrary administrative region and generated a dataset more likely to capture anomalies in networks.



**Table 2** Variance weighted  $R^2$  from different pipelines

Model	Modeling space	Disaggregation	JFK $R^2$			LGA $R^2$		
			TaxiZone	Aggregated	Taxi Zone anormal days	TaxiZone	Aggregated	Taxi Zone anormal days
RF	Taxi Zone	–	0.545	–	0.520	0.766	–	0.742
RF	Borough	Weighted	0.491	0.865	0.463	0.623	0.929	0.621
RF	PCA-6	Inverse PCA	0.568	0.807	0.533	0.775	0.873	0.753
RF	PCA-24	Inverse PCA	0.553	0.624	0.51	0.766	0.702	0.724
RF	Comm-6	Weighted	0.512	0.859	0.481	0.68	0.925	0.667
RF	Comm-24	Weighted	0.538	0.816	0.514	0.713	0.907	0.698
LSTM	Comm-24	Weighted	0.537	0.855	0.546	0.727	0.912	0.708
LSTM Pipeline	Comm-24	Learnable	0.586	–	0.661	0.776	–	0.750

**Table 3** Area under precision–recall curve at three transportation hubs

Model	Modeling space	JFK	LGA
Random selection		1.0	1.0
RF	City	1.331	1.448
RF	Taxi Zone	0.985	2.752
RF	Borough	3.060	1.086
RF	PCA-6	4.270	1.358
RF	PCA-24	1.711	1.575
RF	Comm-6	4.149	1.521
RF	Comm-24	4.356	3.005
LSTM	Comm-24	3.596	1.340

### 5.3 Spatial footprint of anomalies

We aim at detecting interpretable and distinctive spatial patterns of anomalies. Anomalies have different types and different scopes of influence. An event could lead to overestimation or underestimation of the prediction, and the effect could be global or local. The following methods are based on outputs from Comm24-RandomForest pipeline as it has the best performance in anomaly detection.

The first step is to locate a reasonable threshold from the anomaly detection framework to label anomalies and regular days. The framework has 100 iterations in total, the threshold of likelihood in iterations from 0.01 to 1.00 to divide anomalous days and normal days. To determine a proper threshold, we compare the mean absolute error of taxi-zone-level prediction values between anomalies and ordinary days (Appendix B Figs. 8 and 9). The threshold at JFK is 13%, which means that days below the bottom 13% likelihood will be labeled as anomalies, and the threshold at LGA is 5%. In total, JFK has 47 anomalous days, and LGA has 18.

The next step is to investigate anomalies’ spatial footprints. We applied the Gaussian Mixture Model to cluster detected anomalous days based on community-level residuals from Comm24-Random Forest. Same as anomaly detection, the number of components in a

GMM is selected from the one returns the lowest BIC. At JFK, we have 4 different types of anomalies, 10 anomalies in type 0, 13 in type 1, and both type 2 and type 3 have 11 anomalies. LGA anomalies are divided; each type has 9 anomalies.

To measure the impact on prediction from anomalies in different types, we calculated normalized residual and the consistency

$$R' = \frac{T - T' + 1}{T + 1}, \tag{10}$$

where  $R'$  is normalized residuals,  $T$  is actual ridership, and  $T'$  is predicted ridership

$$C = \frac{D'}{D}, \tag{11}$$

where  $C$  is the consistency. For one community, if the average of residuals from one cluster of anomalies is greater than 0, then  $D'$  is the number of days that normalized residuals are positive, and if the average of residuals are below 0, then  $D'$  is the number of days that normalized residuals are negative.

We observe four types of anomalies for taxi+FHV ridership at JFK and two at LGA. Figures 2 and 3 are direction and consistency of impact from different clusters of anomalies for JFK ridership. At the first glance, we conclude that, on average, type 1 anomalies lead to underestimation in the entire NYC, while type 0, type 2, and type 3 anomalies have different impact directions at different communities. Also, anomalies in type 3 have a constant impact in most communities, while in other types, effects are only consistent in some communities. For example, more than 80% (8 days out of 10) of type 0 anomalies lead to overestimation at East Staten Island, Long Beach, Crown Heights, Astoria, Long Island City, and Bronx. In contrast, anomalies of the same type have different impact directions at other locations. It is also clear that anomalies (13 days) in type 1 lead to underestimation in the entire New York City in average, and the underestimating impact is constant in at least 11 out of 13 days at West Staten Island, Long Beach, South Brooklyn, East Queens, and

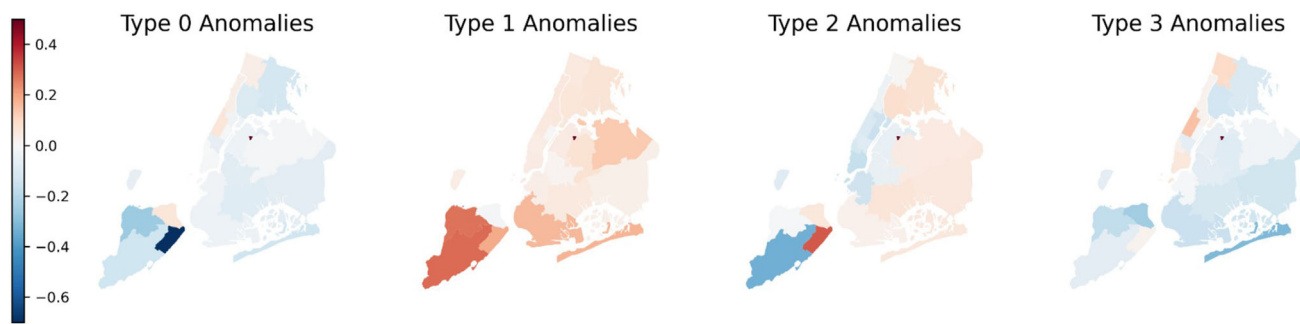


Fig. 2 Normalized residual distribution at JFK

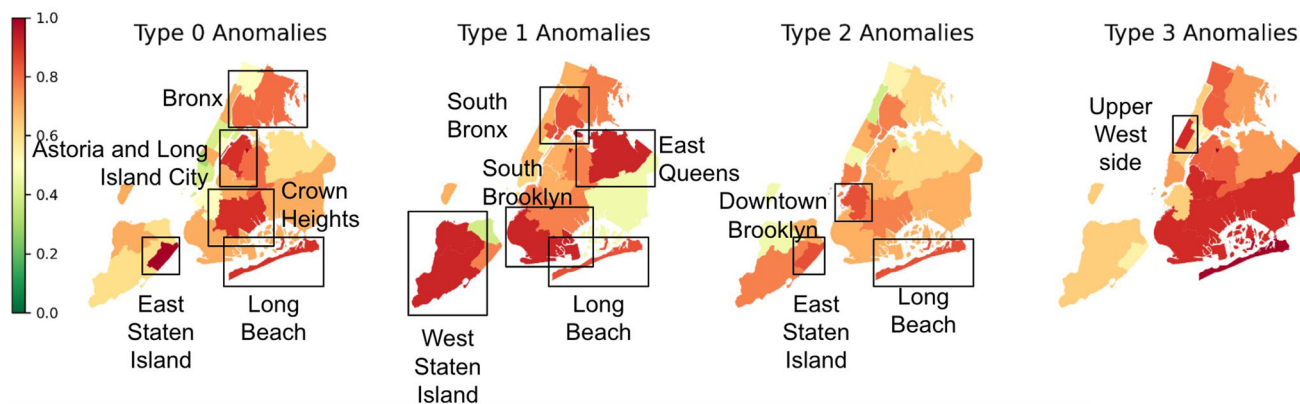


Fig. 3 Consistency distribution at JFK

Table 4 Detected events in JFK ridership

	Type 0	Type 1	Type 2	Type 3
Federal holidays and the day before or after it	3	3	2	1
Extreme weather	1	1	2	0
Airport event	0	1	0	0
Other	6	8	9	10
Total	10	13	13	11

South Bronx. However, the impacts of type 1 anomalies are not consistent in all communities. Type 2 anomalies are consistent at the least amount of communities, only at East Staten Island, Long Beach, and downtown Brooklyn.

We also explore the temporal patterns in clustered anomalies: consecutive days are more likely to be clustered into the same group. Tables 3 and 4 show the contribution of different types of anomalies in different clusters. Besides our labeled events like federal holidays, detected abnormalities include observed holidays like the day before/after Christmas day. It can be observed that certain types of events are contributing more to some clusters. The future step of this task is to examine the exact events of these clusters to generate a group of representative scenarios for travel demand anomalies.

## 6 Conclusion

We evaluate prediction and anomaly detection methods for the taxi ridership with the destination in the two transportation hubs in New York City—JFK and LGA airports. Our study emphasized that transportation data have strong nonlinear temporal and spatial dependencies, and hence, the nonlinear Random Forest prediction model outperforms the baseline linear model. Furthermore, we find that leveraging LSTM deep learning techniques could improve spatio-temporal traffic modeling.

Our work addresses the challenge of modeling high-dimensional network data. The proposed pipeline approach for predictive ridership modeling utilizes appropriate spatial aggregation, nonlinear modeling, and subsequent disaggregation back to the original spa-

tial scale. This way, it outperforms the direct ridership modeling on the original spatial resolution leading to poor anomaly detection due to the lack of consideration of the underlying network topology. And the comparison of different aggregation techniques showed that network community-based aggregation performs the best for the prediction as well as further anomaly detection, highlighting the importance of accounting for the network topology.

We consider the assumptions used in preparing the list of the labeled anomalous events as a potential limitation of this study. Since the complete ground truth data for the anomalies are not present, our analysis assumes that the collected events should represent the major anomalies in the mobility network. Besides, as our list of anomalous events is not exhaustive, detecting anomalies other than those listed does not necessarily mean a shortcoming of the approach. Therefore, our ability to evaluate the anomaly detection performance is limited by the comprehensiveness of the labeled anomalous events.

We have observed that our proposed pipeline framework of dimensionality reduction, modeling, and disaggregation outperforms the direct modeling and anomaly detection for the original transportation network. Furthermore, we note that when we jointly optimize the modeling and disaggregation step using gradient descent in our pipeline LSTM model, it results in superior performance. This might warrant further study considering a model where all three stages, including the aggregation, are jointly trained for optimizing prediction accuracy. Further evaluations of the modeling and anomaly detection methodology could also involve the graph neural network embedding of the mobility network [<https://arxiv.org/abs/2105.03388>].

**Acknowledgements** This work is partly funded by the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy, under Award Number DE-EE0008524. The authors are solely responsible for the findings in this paper.

**Data Availability Statement** This manuscript has associated data in a data repository. [Authors' comment: The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request. Original datasets' sources are publicly available, please refer to Sect. 3.]

## Appendix A Tables

See Tables 5, 6, 7, and 8.

**Table 5** Summary of aggregated outgoing traffic data sets

Vehicle type	Explanation	Outgoing traffic volume	
		JFK	LGA
FHV	For-hire vehicle (Uber, Lyft, etc.)	3,499,760	3,099,427
Yellow	Yellow taxi cab	2,577,485	2,894,811
Total	The sum of FHV and Yellow	6,077,245	5,994,238

**Table 6** Arrival record dataset at transport hubs

Name	Explanation	Unit	Arrival count		Record count	
			JFK	LGA	JFK	LGA
Arrival	BTS detailed statistics arrivals	Per flight	129,954	171,104	129,954	171,104

**Table 7** Events datasets’ description

	Federal holiday	Extreme weather	Airport event
Days in JFK	10	6	4
Days in LGA	10	6	1
Data source	Official holidays [71]	National Centers for Environmental Information [72]	news

**Table 8** Hybrid model variables explanation

Variable	Explanation	Range
$RIDERSHIP_t$	Number of selected type of ridership at window index = t	JFK, LGA
$ARRIVAL_{t,m,n}$	$ARRIVAL_{t,m,n} = arrival_{t-m} + arrival_{t-m+1} + \dots + arrival_{t-n}$ arrival <sub>t</sub> is arrival flight/passenger amount at window index = t	Arrival
$DayofWeek_t$	Dummy variables indicating whether t is on MON, ..., SUN $SUNTimeofDay_t = IfMon_t + \dots + IfSun_t$	0,1
$TimeofDay_t$	Dummy variables indicating whether t is on [0am–1am], ..., [23pm–24pm].	0, 1
$\beta$	$\beta_1 = (\beta_{1,t-m}, \dots, \beta_{1,t-m})^T$ $\beta_2 = (\beta_{2,1}, \dots, \beta_{2,7})^T$ $\beta_3 = (\beta_{3,0}, \dots, \beta_{3,23})^T$	–
t	Window index, where window size is 1 hour.	0, ..., (365*24*60 / window-1)



## Appendix B Figures

See Figs. 4, 5, 6, 7, 8, 9, 10, and 11.



Fig. 4 Boroughs New York



Fig. 5 Taxi zones New York

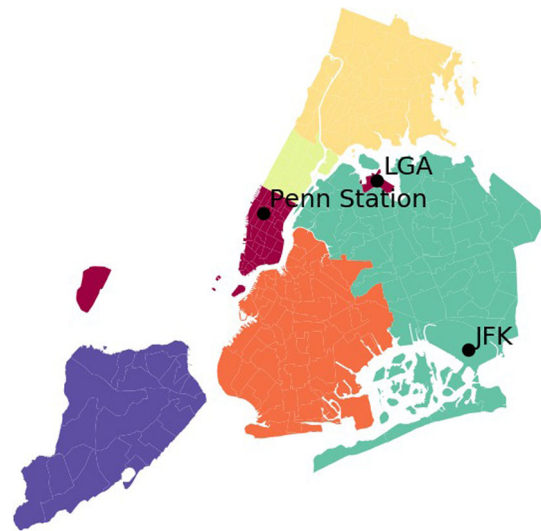


Fig. 6 Taxi zones to 6 communities

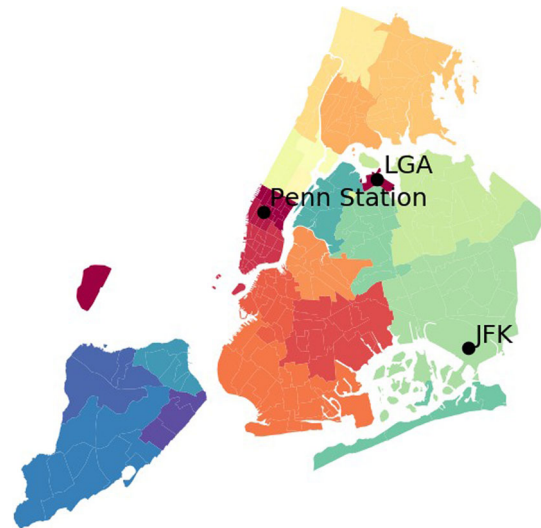
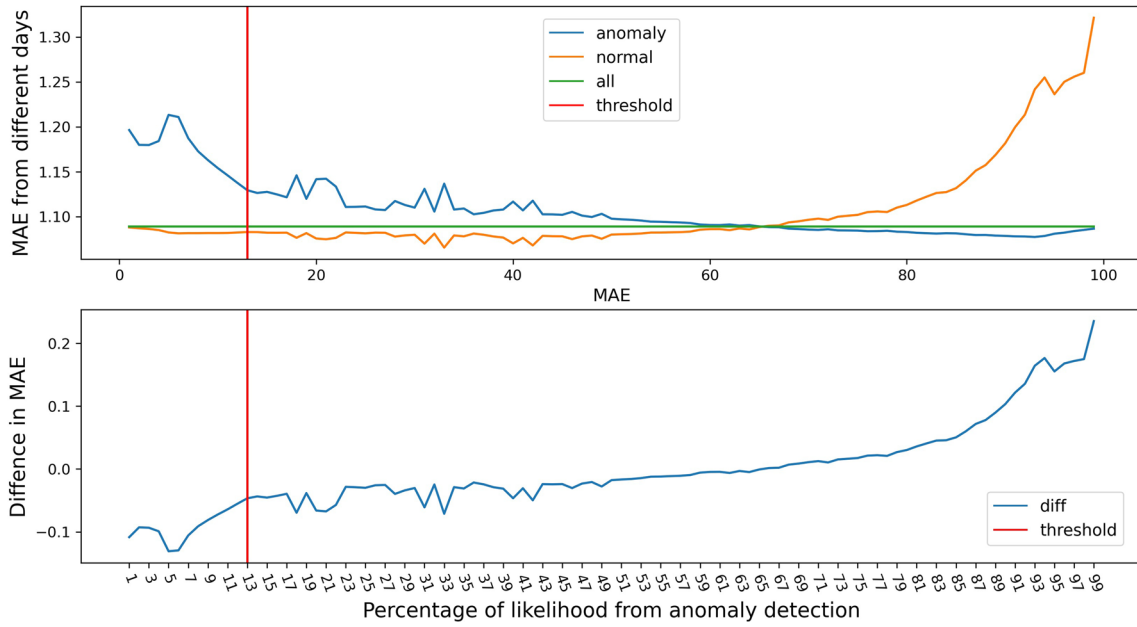
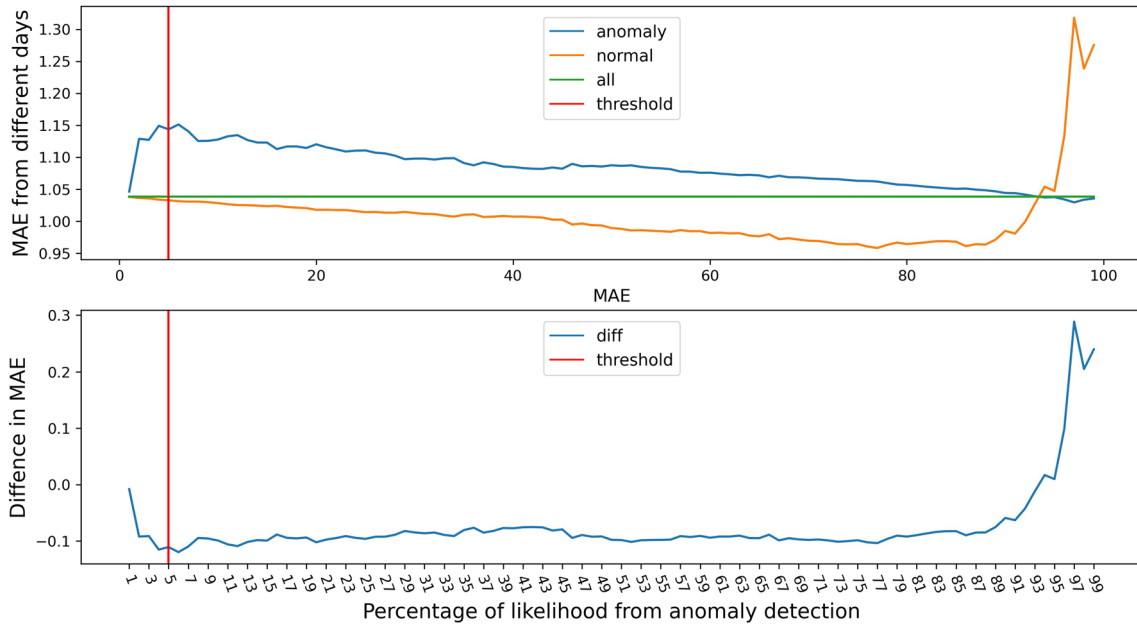


Fig. 7 Taxi zones to 24 communities

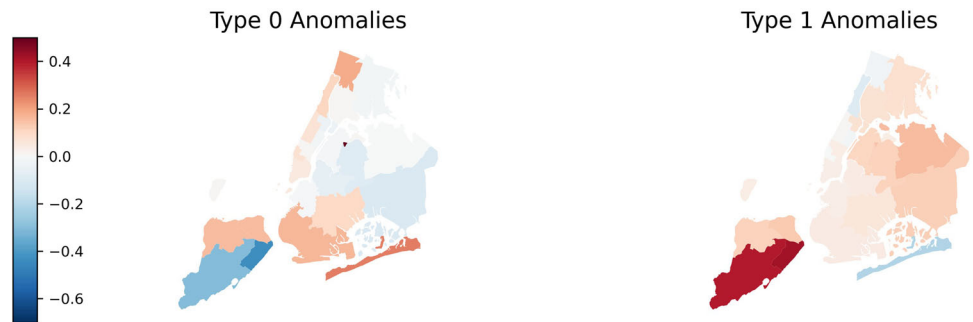


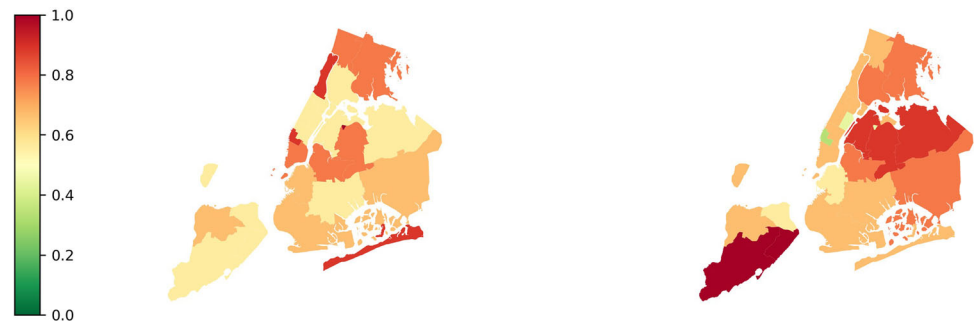
**Fig. 8** JFK threshold of anomalies in mean absolute error



**Fig. 9** LGA threshold of anomalies in mean absolute error

**Fig. 10** Normalized residual distribution at LGA



**Fig. 11** Consistency distribution at JFK

## References

1. L. Figueiredo, I. Jesus, J.A.T. Machado, J.R. Ferreira, J.L. Martins De Carvalho, Towards the development of intelligent transportation systems. In: ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585), pp. 1206–1211. IEEE (2001)
2. M.L. Anderson, Subways, strikes, and slowdowns: the impacts of public transit on traffic congestion. *Am. Econ. Rev.* **104**(9), 2763–96 (2014)
3. E.L. Glaeser, M.E. Kahn, The greenness of cities: carbon dioxide emissions and urban development. *J. Urban Econ.* **67**(3), 404–418 (2010)
4. L. Zha, Y. Yin, X. Zhengtian, Geometric matching and spatial pricing in ride-sourcing markets. *Transp. Res. Part C Emerg. Technol.* **92**, 58–75 (2018)
5. S. Qida, D.Z.W. Wang, Morning commute problem with supply management considering parking and ride-sourcing. *Transp. Res. Part C Emerg. Technol.* **105**, 626–647 (2019)
6. D.M. Hawkins, *Identification of Outliers*, vol. 11 (Springer, Berlin, 1980)
7. M. He, S. Pathak, U. Muaz, J. Zhou, S. Saini, S. Malinchik, S. Sobolevsky, Pattern and anomaly detection in urban temporal networks. *arXiv preprint arXiv:1912.01960* (2019)
8. V. Hodge, J. Austin, A survey of outlier detection methodologies. *Artif. Intell. Rev.* **22**(2), 85–126 (2004)
9. H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, Z. Li, Deep multi-view spatial-temporal network for taxi demand prediction. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
10. X. Qian, S.V. Ukkusuri, C. Yang, F. Yan, Short-term demand forecasting for on-demand mobility service. *IEEE Trans. Intell. Transp. Syst.* **23**(2), 1019–1029 (2022). <https://doi.org/10.1109/TITS.2020.3019509>
11. X. Qian, S.V. Ukkusuri, Spatial variation of the urban taxi ridership using GPS data. *Appl. Geogr.* **59**, 31–42 (2015)
12. X. Jun, R. Rahmatizadeh, L. Bölöni, D. Turgut, Real-time prediction of taxi demand using recurrent neural networks. *IEEE Trans. Intell. Transp. Syst.* **19**(8), 2572–2581 (2017)
13. D. Correa, K. Xie, K. Ozbay, Exploring the taxi and uber demand in New York City: an empirical analysis and spatial modeling. Technical report (2017)
14. M.P. Souza, A.A.M. Oliveira, M.A. Pereira, F.A.L. Reis, P.E.M. Almeida, E.J. Silva, D.S. Crepalde, Optimization of taxi cabs assignment using a geographical location-based system in distinct offer and demand scenarios. *Rev. Bras. De Cartogr* **68**, 1143–1155 (2015)
15. H.R. Sayarshad, J.J. Chow, Survey and empirical evaluation of nonhomogeneous arrival process models with taxi data. *J. Adv. Transp.* **50**(7), 1275–1294 (2016)
16. F. Rodrigues, I. Markou, F.C. Pereira, Combining time-series and textual data for taxi demand prediction in event areas: a deep learning approach. *Inf. Fusion* **49**, 120–129 (2019)
17. K. Zhao, D. Khryashchev, J. Freire, C. Silva, H. Vo, Predicting taxi demand at high spatial resolution: approaching the limit of predictability. In: 2016 IEEE International Conference on Big Data (Big Data), pp. 833–842. IEEE (2016)
18. G.E.P. Box, D.A. Pierce, Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *J. Am. Stat. Assoc.* **65**(332), 1509–1526 (1970)
19. D. Simon, Kalman filtering. *Embed. Syst. Program.* **14**(6), 72–79 (2001)
20. L. Moreira-Matias, J. Gama, M. Ferreira, L. Damas, A predictive model for the passenger demand on a taxi network. In: 2012 15th International IEEE Conference on Intelligent Transportation Systems, pp. 1014–1019. IEEE (2012)
21. M. Levin, Y.-D. Tsao. On forecasting freeway occupancies and volumes (abridgment). *Transp. Res. Record* (773) (1980)
22. B.M. Williams, P.K. Durvasula, D.E. Brown, Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models. *Transp. Res. Record* **1644**(1), 132–141 (1998)
23. A. Klein, Forecasting the Antwerp maritime traffic flows using transformations and intervention models. *J. Forecast.* **15**(5), 395–412 (1996)
24. H. Lütkepohl, Linear transformations of vector ARMA processes. *J. Econom.* **26**(3), 283–293 (1984)
25. P.E. Pfeifer, S.J. Deutsch, Seasonal space-time ARIMA modeling. *Geogr. Anal.* **13**(2), 117–133 (1981)
26. Q.Y. Ding, X.F. Wang, X.Y. Zhang, Z.Q. Sun, Forecasting traffic volume with space-time ARIMA model. *Adv. Mater. Res.* **156**, 979–983 (2011)
27. S.R. Chandra, H. Al-Deek, Predictions of freeway traffic speeds and volumes using vector autoregressive models. *J. Intell. Transp. Syst.* **13**(2), 53–72 (2009)
28. J. Ke, S. Feng, Z. Zhu, H. Yang, J. Ye, Joint predictions of multi-modal ride-hailing demands: a deep multi-task multi-graph learning-based approach. *Transp. Res. Part C Emerg. Technol.* **127**, 103063 (2021)
29. J. Ke, X. Qin, H. Yang, Z. Zheng, Z. Zhu, J. Ye, Predicting origin-destination ride-sourcing demand with a spatio-temporal encoder-decoder residual multi-graph

- convolutional network. *Transp. Res. Part C Emerg. Technol.* **122**, 102858 (2021)
30. J. Tang, J. Liang, F. Liu, J. Hao, Y. Wang, Multi-community passenger demand prediction at region level based on spatio-temporal graph convolutional network. *Transp. Res. Part C Emerg. Technol.* **124**, 102951 (2021)
  31. H. Yao, X. Tang, H. Wei, G. Zheng, Y. Yu, Z. Li, Modeling spatial-temporal dynamics for traffic prediction. arXiv preprint [arXiv:1803.01254](https://arxiv.org/abs/1803.01254) (2018)
  32. J. Ke, H. Zheng, H. Yang, X.M. Chen, Short-term forecasting of passenger demand under on-demand ride services: a spatio-temporal deep learning approach. *Transp. Res. Part C Emerg. Technol.* **85**, 591–608 (2017)
  33. Yu. Junbo Zhang, J.S. Zheng, D. Qi, Flow prediction in spatio-temporal networks based on multitask deep learning. *IEEE Trans. Knowl. Data Eng.* **32**(3), 468–478 (2019)
  34. C. Zhang, F. Zhu, X. Wang, L. Sun, H. Tang, Y. Lv, Taxi demand prediction using parallel multi-task learning model. *IEEE Trans. Intell. Transpo. Syst.* **23**(2), 794–803 (2022). <https://doi.org/10.1109/TITS.2020.3015542>
  35. L. Liu, Z. Qiu, G. Li, Q. Wang, W. Ouyang, L. Lin, Contextualized spatial-temporal network for taxi origin-destination demand prediction. *IEEE Trans. Intell. Transp. Syst.* **20**(10), 3875–3887 (2019)
  36. A. Klein, C. Craun, R.S. Lee, Airport delay prediction using weather-impacted traffic index (WITI) model. In: 29th Digital Avionics Systems Conference, pp. 2–B. IEEE (2010)
  37. M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering. *Adv. Neural Inf. Process. Syst.* **29** (2016)
  38. B. Yu, H. Yin, Z. Zhu, Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. arXiv preprint [arXiv:1709.04875](https://arxiv.org/abs/1709.04875) (2017)
  39. W. Chen, Yu. Ling Chen, W.C. Xie, Y. Gao, X. Feng, Multi-range attentive bicomponent graph convolutional network for traffic forecasting. *Proc. AAAI Conf. Artif. Intell.* **34**, 3529–3536 (2020)
  40. Z. Pan, F. Cai, W. Chen, H. Chen, M. de Rijke, Star graph neural networks for session-based recommendation. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management, pp. 1195–1204 (2020)
  41. Y. Tong, Y. Chen, Z. Zhou, L. Chen, J. Wang, Q. Yang, J. Ye, W. Lv, The simpler the better: a unified approach to predicting original taxi demands based on large-scale online platforms. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1653–1662. ACM (2017)
  42. L. Breiman, Bagging predictors. *Mach. Learn.* **24**(2), 123–140 (1996)
  43. S. Hochreiter, J. Schmidhuber, Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
  44. M. Ahmed, A.N. Mahmood, J. Hu, A survey of network anomaly detection techniques. *J. Netw. Comput. Appl.* **60**, 19–31 (2016)
  45. J. Bohannon, Tweeting the London riots. *Science* **336**(6083), 831 (2012). <https://doi.org/10.1126/science.336.6083.831>
  46. H. Abdelhaq, C. Sengstock, M. Gertz, Eventtweet: online localized event detection from twitter. *Proc. VLDB Endow.* **6**(12), 1326–1329 (2013)
  47. X. Kong, H. Gao, O. Alfarraj, Q. Ni, C. Zheng, G. Shen, HUAD: hierarchical urban anomaly detection based on spatio-temporal data. *IEEE Access* **8**, 26573–26582 (2020)
  48. K. Pasini, Forecast and anomaly detection on time series with dynamic context. Application to the mining of transit ridership data. PhD thesis, Université gustave eiffel (2021)
  49. A. Lakhina, M. Crovella, C. Diot, Diagnosing network-wide traffic anomalies. *ACM SIGCOMM Comput. Commun. Rev.* **34**(4), 219–230 (2004)
  50. C.C. Aggarwal, An introduction to outlier analysis, in *Outlier Analysis*. (Springer, Berlin, 2017), pp. 1–34
  51. H. Ringberg, A. Soule, J. Rexford, C. Diot, Sensitivity of PCA for traffic anomaly detection. In: Proceedings of the 2007 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, pp. 109–120 (2007)
  52. D. Brauckhoff, K. Salamatian, M. May, Applying PCA for traffic anomaly detection: problems and solutions. In: IEEE INFOCOM 2009, pp. 2866–2870. IEEE (2009)
  53. M. Girvan, M.E.J. Newman, Community structure in social and biological networks. *Proc. Natl. Acad. Sci.* **99**(12), 7821–7826 (2002)
  54. Z. Chen, W. Hendrix, N.F. Samatova, Community-based anomaly detection in evolutionary networks. *J. Intell. Inf. Syst.* **39**(1), 59–85 (2012)
  55. D. Liu, C.-H. Lung, I. Lambadaris, N. Seddigh, Network traffic anomaly detection using clustering techniques and performance comparison. In: 2013 26th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 1–4. IEEE (2013)
  56. D. Tran, W. Ma, D. Sharma, Network anomaly detection using fuzzy gaussian mixture models. *Int. J. Future Gener. Commun. Netw.* **1**(1), 37–42 (2006)
  57. W. Kuang, S. An, H. Jiang, Detecting traffic anomalies in urban areas using taxi GPS data. *Math. Probl. Eng.* **2015** (2015)
  58. New York City Taxi & Limousine Commission. Tlc trip record data. <https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page>. Accessed: 2019-11-30
  59. D. Zhang, T. He, S. Lin, S. Munir, J.A. Stankovic, Taxi-passenger-demand modeling based on big data from a roving sensor network. *IEEE Trans. Big Data* **3**(3), 362–374 (2016)
  60. F. Miao, S. Han, S. Lin, J.A. Stankovic, D. Zhang, S. Munir, H. Huang, T. He, G.J. Pappas, Taxi dispatch with real-time sensing data in metropolitan areas: a receding horizon control approach. *IEEE Trans. Autom. Sci. Eng.* **13**(2), 463–478 (2016)
  61. C. Kamga, M.A. Yazici, A. Singhal, Analysis of taxi demand and supply in New York City: implications of recent taxi regulations. *Transp. Plan. Technol.* **38**(6), 601–625 (2015)
  62. B. Mutzabaugh, Flight delays mount at nyc airports as pre-christmas storm slams region, *Herald* (2018)
  63. R.E. Shapire, Y. Singer. Boostexter: a system for multi-label text categorization. *Mach. Learn.* **39**(2/3), 135–168 (1998)
  64. A. Liaw, M. Wiener et al., Classification and regression by randomforest. *R News* **2**(3), 18–22 (2002)
  65. G. Biau, E. Scornet, A random forest guided tour. *TEST* **25**(2), 197–227 (2016)



66. Y. She, A.B. Owen, Outlier detection using nonconvex penalized regression. *J Am Stat Assoc* **106**(494), 626–639 (2011)
67. A.A. Green, M.D. Craig, Analysis of aircraft spectrometer data with logarithmic residuals. In: *JPL Proceedings of the Airborne Imaging Spectrometer Data Anal. Workshop* (1985)
68. L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, G. Varoquaux. API design for machine learning software: experiences from the SCIKIT-learn project. In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122 (2013)
69. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, SCIKIT-learn: machine learning in python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
70. J. Davis, M. Goadrich. The relationship between precision-recall and roc curves. In: *Proceedings of the 23rd International Conference on Machine Learning*, pp. 233–240 (2006)
71. Official Holidays. TLC trip record data. <https://www.officeholidays.com/countries/usa/new-york/2018>. Accessed: 2019-11-30
72. National Centers for Environmental Information. Storm events database. [https://www.ncdc.noaa.gov/stormevents/listevents.jsp?eventType=ALL&beginDate\\_mm=01&beginDate\\_dd=01&beginDate\\_yyyy=2018&endDate\\_mm=12&endDate\\_dd=31&endDate\\_yyyy=2018&county=NEW%2BYORK%3A61&hailfilter=0.00&tornfilter=0&windfilter=000&sort=DT&submitbutton=Search&statefips=36%2CNEW+YORK](https://www.ncdc.noaa.gov/stormevents/listevents.jsp?eventType=ALL&beginDate_mm=01&beginDate_dd=01&beginDate_yyyy=2018&endDate_mm=12&endDate_dd=31&endDate_yyyy=2018&county=NEW%2BYORK%3A61&hailfilter=0.00&tornfilter=0&windfilter=000&sort=DT&submitbutton=Search&statefips=36%2CNEW+YORK). Accessed: 2019-11-30